aws

Entwicklerhandbuch

Amazon CloudFront



Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon CloudFront: Entwicklerhandbuch

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Die Marken und Handelsmarken von Amazon dürfen nicht in einer Weise in Verbindung mit nicht von Amazon stammenden Produkten oder Services verwendet werden, die geeignet ist, Kunden irrezuführen oder Amazon in irgendeiner Weise herabzusetzen oder zu diskreditieren. Alle anderen Handelsmarken, die nicht Eigentum von Amazon sind, gehören den jeweiligen Besitzern, die möglicherweise zu Amazon gehören oder nicht, mit Amazon verbunden sind oder von Amazon gesponsert werden.

Table of Contents

Was ist Amazon CloudFront? 1
Wie richten Sie CloudFront die Bereitstellung von Inhalten ein 2
Auswahl zwischen Standarddistribution und Multi-Tenant-Distribution 5
Preisgestaltung 5
Möglichkeiten zur Verwendung CloudFront 6
Beschleunigen der Bereitstellung von Inhalten auf einer statischen Website 6
Bereitstellen von On-Demand- oder Live-Streaming-Video 6
Verschlüsseln von bestimmten Feldern während der Systemverarbeitung 7
An der Grenze anpassen 7
Bereitstellen von privaten Inhalten mit Hilfe von Lambda@Edge-Anpassungen 8
Wie liefert Inhalte CloudFront 9
Wie CloudFront werden Inhalte an Ihre Nutzer geliefert 9
Wie CloudFront funktioniert das mit regionalen Edge-Caches 10
CloudFront Edge-Server 12
Verwenden Sie die Liste der CloudFront verwalteten Präfixe 13
Mit AWS SDKs arbeiten 14
CloudFront technische Ressourcen 15
Erste Schritte 16
Richten Sie Ihre ein AWS-Konto 16
Melden Sie sich für eine an AWS-Konto 17
Wählen Sie, wie Sie darauf zugreifen möchten CloudFront 17
Erste Schritte mit einer Standarddistribution 18
Voraussetzungen 18
Erstellen eines Buckets 19
Hochladen der Inhalte 19
Erstellen einer Distribution 20
Zugriff auf den Inhalt 21
Bereinigen 22
Verbessern Ihrer grundlegenden Distribution 22
Fangen Sie an (AWS CLI) 22
Voraussetzungen 23
Erstellen Sie einen Amazon S3 S3-Bucket 23
Hochladen des Inhalts in den Bucket 24
Erstellen einer Ursprungszugriffssteuerung (OAC) 24

Erstellen einer Standarddistribution ... 25
Aktualisieren Ihrer S3-Bucket-Richtlinie ... 26
Bestätigen der Bereitstellung der Distribution ... 27
Greifen Sie auf Ihre Inhalte zu über CloudFront ... 27
Bereinigen ... 28
Erste Schritte mit einer sicheren statischen Website ... 29
Übersicht über die Lösung ... 30
Bereitstellen der Lösung ... 31
E-commerce Verteilungen ... 38
Übersicht über die Architektur ... 38
Verhalten für E-Commerce zwischenspeichern ... 39
TTL-Strategie für Marketplace-Inhalte ... 41
Origin-Konfiguration ... 43
Amazon S3-Ursprung für statische Assets ... 43
ALB-Ursprung für dynamische Inhalte ... 43
Application-level Caching mit ElastiCache (Valkey) ... 45
Häufig gestellte Fragen ... 46
Wie gehe ich beim Caching mit personalisierten Inhalten um? ... 46
Wie wähle ich zwischen Invalidierung und kurzen TTLs? ... 46
Wie bereite ich mich auf schnelle Verkäufe und Traffic-Spitzen vor? ... 46
Sollte ich Origin Shield für meinen Marktplatz verwenden? ... 46
E-commerce Sicherheitscheckliste ... 48
Signierte URLs und Cookies für Premium-Inhalte ... 48
Origin Access Control (OAC) für Amazon S3 ... 49
Verschlüsselung während der Übertragung ... 50
VPC-Sicherheitsgruppen für den Cache-Cluster ... 51
Zusätzliche Sicherheitskontrollen ... 52
CloudFront Pauschalpreispläne ... 54
Vorteile von CloudFront Flatrate-Tarifen ... 55
Funktionen nach Preisstufen ... 56
Funktionen des Preisplans ... 56
Monatliche Nutzungsgebühren ... 102
Was wird auf Ihre Nutzungsgebühr angerechnet ... 103
Was passiert, wenn die Nutzung das Limit überschreitet ... 103
Konfigurierbare Nutzungsberechtigungen im Premium-Tarif ... 104
Die Teilnahmeberechtigung basiert auf der bisherigen Nutzung ... 105

Kosten, die durch Ihren Tarif abgedeckt werden ... 106
Route 53, DNS-Verwaltung und Ihr Tarif ... 106
Insgesamt reduzieren AWS Kosten mit Preisplänen ... 108
Verwalten Sie Ihre Pauschalpreispläne ... 108
Abonnieren Sie einen Preisplan für eine neue Distribution ... 109
Abonnieren Sie für eine bestehende Distribution einen Preisplan ... 109
Aktualisieren Sie einen Preisplan ... 110
Ändern Sie die Nutzungsstufe Ihres Premium-Tarifs ... 110
Einen Preisplan herabstufen ... 111
Stornieren Sie einen Preisplan ... 112
Stornieren Sie eine ausstehende Planänderung ... 112
Löschen einer Verteilung mit einem Preisplan ... 113
Berechtigungen ... 113
Flat-rate Kontingente für Preispläne ... 113
Nicht unterstützte Funktionen ... 114
Nicht unterstützte Funktionen ... 114
Nicht unterstützte Zuordnungen ... 116
Account-level Einschränkungen ... 117
Resource-level Einschränkungen ... 117
Zusätzliche Funktionen, die sich auf Ihren Preisplan auswirken können ... 117
Preispläne im Vergleich zu Pay-as-you-go-Preisen ... 119
Konfigurieren von Distributionen ... 120
Verständnis der Funktionsweise von Multi-Tenant-Distributionen ... 122
Funktionsweise ... 123
Bedingungen ... 125
Nicht unterstützte Funktionen ... 127
Anpassungen von Distributionsmandanten ... 128
Anfordern von Zertifikaten (Distributionsmandant) ... 132
Erstellen einer benutzerdefinierten Verbindungsgruppe (optional) ... 141
Migrieren zu einer Multi-Tenant-Distribution ... 143
Eine Distribution erstellen ... 144
Erstellen Sie eine CloudFront Distribution in der Konsole ... 146
Werte, die angezeigt werden ... 152
Weitere Links ... 153
Fügen Sie Ihrer CloudFront Standarddistribution eine Domain hinzu ... 154
Vorkonfigurierte Distributionseinstellungen ... 156

Amazon-S3-Ursprung ... 157
API-Gateway-Ursprung ... 158
Benutzerdefinierter Ursprung und EC2-Instance ... 159
Elastic Load Balancing als Ursprung ... 161
MediaPackage v1-Ursprung ... 163
MediaPackage v2-Ursprung ... 164
MediaTailor Herkunft ... 165
Alle Distributionseinstellungen ... 166
Ursprungseinstellungen ... 167
Einstellungen für das Cache-Verhalten ... 179
Distribution Settings (Einstellungen für die Verteilung) ... 195
Benutzerdefinierte Fehlerseiten und Zwischenspeicherung von Fehlern ... 205
Geografische Einschränkungen ... 206
Testen einer Distribution ... 207
Erstellen von Links zu Ihren Objekten ... 207
Eine Verteilung aktualisieren ... 208
Aktualisieren einer Distribution in der Konsole ... 208
Kennzeichnen Sie eine CloudFront Ressource ... 211
Tag-Einschränkungen ... 212
Markieren von Ressourcen für die Fakturierung ... 213
Hinzufügen, Bearbeiten und Löschen von Tags ... 213
Programmgesteuertes Markieren ... 214
Löschen einer -Verteilung ... 215
Verwenden von verschiedenen Ursprüngen ... 217
Verwenden eines Amazon-S3-Buckets ... 218
Verwenden Sie einen MediaStore Container oder einen MediaPackage Kanal ... 231
Verwenden eines Application Load Balancer ... 232
Verwenden eines Network Load Balancer ... 232
Verwenden einer Lambda-Funktions-URL ... 232
Verwenden Sie eine AWS Lambda Web-Funktion ... 234
Verwenden von Amazon EC2 (oder eines anderen benutzerdefinierten Ursprungs) ... 235
Verwenden Sie CloudFront Ursprungsgruppen ... 237
Verwenden von Amazon API Gateway ... 237
IPv6 aktivieren ... 238
IPv6-Viewer-Anforderungen ... 238
IPv6-Ursprungsanforderungen ... 240

Verwenden der kontinuierlichen Bereitstellung zum sicheren Testen von Änderungen 240
CloudFront kontinuierlicher Bereitstellungs-Workflow 243
Arbeiten mit einer Staging-Distribution und einer Richtlinie für die kontinuierliche Bereitstellung 244
Überwachen einer Staging-Distribution 255
Informationen zur kontinuierlichen Bereitstellung 256
Kontingente und andere zu berücksichtigende Aspekte bei der kontinuierlichen Bereitstellung 258
Verwenden benutzerdefinierter URLs 260
Voraussetzungen für die Verwendung von alternativen Domänennamen 260
Einschränkungen bei der Verwendung alternativer Domänennamen 262
Hinzufügen eines alternativen Domainnamens 265
Verschieben eines alternativen Domainnamens 269
Entfernen eines alternativen Domainnamens 283
Verwenden von Platzhaltern in alternativen Domainnamen 284
Verwenden WebSockets 285
Wie funktioniert das WebSocket Protokoll 285
WebSocket Anforderungen 286
Empfohlene Header WebSocket 286
Anfordern statischer Anycast-IPs zur Verwendung für Zulassungslisten 287
Voraussetzungen 288
Anfordern einer statischen Anycast-IP-Liste 288
Erstellen einer statischen Anycast-IP-Liste 288
Zuordnen einer statischen Anycast-IP-Liste zu einer vorhandenen Distribution 289
Zuordnen einer statischen Anycast-IP-Liste zu einer neuen Distribution 290
Ordnen Sie eine statische Anycast-IP-Liste einer Verbindungsgruppe zu 291
Aktualisieren Sie eine statische Anycast-IP-Liste 291
Verwenden Sie IPAM CloudFront mit Ihrer eigenen IP-Adresse 292
Verwenden von gRPC 297
So funktioniert gRPC in CloudFront 298
Arbeiten mit gemeinsam genutzten Ressourcen in CloudFront 300
Voraussetzungen für die gemeinsame Nutzung von Ressourcen 301
Gemeinsame Nutzung eines VPC-Ursprungs 301
Verwenden eines gemeinsam genutzten VPC-Ursprungs 304
Identifizierung eines gemeinsamen VPC-Ursprungs 305
Die gemeinsame Nutzung eines VPC-Ursprungs aufheben 305

Zuständigkeiten und Berechtigungen für gemeinsam genutzte VPC-Ursprünge 306
Berechtigungen für Besitzer 306
Berechtigungen für Konsumenten 307
AWSRAMDefaultPermissionCloudFront 307
Fakturierung und Messung 307
Kontingente für gemeinsam genutzte Ressourcen 307
Caching und Verfügbarkeit 308
Verbessern Ihrer Cache-Trefferrate 309
Geben Sie an, wie lange Ihre CloudFront Objekte zwischengespeichert werden 309
Verwenden von Origin Shield 309
Zwischenspeichern auf der Grundlage von Abfragezeichenfolgeparametern 310
Zwischenspeichern auf der Grundlage von Cookie-Werten 311
Zwischenspeichern auf der Grundlage von Anfrage-Headern 312
Entfernen des `Accept-Encoding`-Headers, wenn keine Kompression erforderlich ist 313
Bereitstellen von Medieninhalten über HTTP 313
Verwenden von Origin Shield 313
Anwendungsfälle für Origin Shield 315
Wählen Sie das Symbol AWS Region für Origin Shield 321
Origin Shield aktivieren 323
Schätzung der Kosten für Origin Shield 326
Hochverfügbarkeit bei Origin Shield 327
Wie Origin Shield mit anderen CloudFront Funktionen interagiert 327
Verbessern der Verfügbarkeit mit Origin Failover 328
Erstellen einer Ursprungsgruppe 330
Steuern von Timeouts und Verbindungsversuchen für Ursprünge 332
Verwenden von Origin Failover mit Lambda@Edge-Funktionen 333
Verwenden von benutzerdefinierten Fehlerseiten mit Ursprung-Failover 334
Verwalten des Cache-Ablaufdatums 335
Verwenden von Headern zum Steuern der Cache-Dauer für einzelne Objekte 336
Bereitstellung veralteter (abgelaufener) Inhalte 337
Geben Sie an, wie lange Objekte zwischengespeichert werden CloudFront 340
Hinzufügen von Headern zu Ihren Objekten mithilfe der Amazon-S3-Konsole 347
Caching- und Abfragezeichenfolgenparameter 347
Konsolen- und API-Einstellungen für die Weiterleitung und Zwischenspeicherung von Abfragezeichenfolgen 350
Optimieren der Zwischenspeicherung 350

Parameter für Abfragezeichenfolgen und CloudFront Standardprotokolle (Zugriffsprotokolle) ... 352
Zwischenspeichern von Inhalten auf der Grundlage von Cookies ... 352
Zwischenspeichern von Inhalten auf der Grundlage von Anforderungsheadern ... 356
Header und Verteilungen – Übersicht ... 357
Auswählen der Header, auf denen die Zwischenspeicherung basieren soll ... 358
Konfigurieren Sie so CloudFront , dass die CORS-Einstellungen berücksichtigt werden ... 360
Konfigurieren der Zwischenspeicherung basierend auf dem Gerätetyp ... 360
Konfigurieren der Zwischenspeicherung basierend auf der Sprache des Viewers ... 361
Konfigurieren der Zwischenspeicherung basierend auf dem Standort des Viewers ... 361
Konfigurieren der Zwischenspeicherung basierend auf dem Protokoll der Anforderung ... 361
Konfigurieren der Zwischenspeicherung für komprimierte Dateien ... 362
Auswirkungen der Zwischenspeicherung auf der Grundlage von Headern auf die Leistung ... 362
Auswirkungen der Schreibweise von Headern und Header-Werten auf die Zwischenspeicherung ... 362
Header, die zum CloudFront Viewer zurückkehren ... 363
Steuern des Cache-Schlüssels mit einer Richtlinie ... 364
Grundlegendes zu Cache-Richtlinien ... 365
Richtlinieninformationen ... 365
Einstellungen für Time to Live (TTL, Gültigkeitsdauer der Verbindung) ... 365
Cache-Schlüssel-Einstellungen ... 367
Erstellen von Cache-Richtlinien ... 373
Verwenden verwalteter Cache-Richtlinien ... 378
Amplify ... 378
CachingDisabled ... 380
CachingOptimized ... 380
CachingOptimizedForUncompressedObjects ... 381
Elementar- MediaPackage ... 382
UseOriginCacheControlHeaders ... 383
UseOriginCacheControlHeaders-QueryStrings ... 384
Grundlegendes zum Cache-Schlüssel ... 385
Standard-Cache-Schlüssel ... 386
Anpassen des Cache-Schlüssels ... 387
Steuern von Ursprungsanforderungen anhand einer Richtlinie ... 390
Grundlegendes zu Ursprungsanforderungsrichtlinien ... 391
Richtlinieninformationen ... 391

Ursprungsanforderungseinstellungen ... 392
Erstellen von Ursprungsanforderungsrichtlinien ... 394
Verwenden verwalteter Ursprungsanforderungsrichtlinien ... 399
AllViewer ... 400
AllViewerAndCloudFrontHeaders-2022-06 ... 400
AllViewerExceptHostHeader ... 401
CORS-CustomOrigin ... 402
CORS-S3Origin ... 403
Elemental-MediaTailor-PersonalizedManifests ... 403
HostHeaderOnly ... 404
UserAgentRefererHeaders ... 404
CloudFront Anforderungsheader hinzufügen ... 405
Gerätetyp-Header ... 406
Viewer-Standort-Header ... 406
Header zur Bestimmung der Header-Struktur des Viewers ... 408
TLS-bezogene Header ... 408
Andere Header CloudFront ... 410
Erläuterung des Zusammenwirkens von Ursprungsanforderungsrichtlinien und Cache-Richtlinien ... 410
Hinzufügen oder Entfernen von Antwortheadern mit einer Richtlinie ... 415
Grundlegendes zu Antwortheader-Richtlinien ... 416
Richtliniendetails (Metadaten) ... 417
CORS-Header ... 417
Sicherheits-Header ... 421
Benutzerdefinierte Header ... 423
Entfernen von Headern ... 424
Server-Timing Kopfzeile ... 426
Erstellen von Antwortheader-Richtlinien ... 431
Verwenden verwalteter Antwortheader-Richtlinien ... 438
Cors-und- SecurityHeadersPolicy ... 439
CORS-With-Preflight ... 440
CORS- - with-preflight-and SecurityHeadersPolicy ... 441
SecurityHeadersPolicy ... 442
SimpleCORS ... 443
Verhalten von Anfragen und Antworten ... 445
Wie CloudFront verarbeitet HTTP- und HTTPS-Anfragen ... 445

Verhalten von Anfragen und Antworten für Amazon-S3-Ursprünge 446
Wie CloudFront verarbeitet und leitet Anfragen an Ihren Amazon S3-Ursprung weiter 446
Wie CloudFront verarbeitet Antworten von Ihrem Amazon S3-Ursprung 454
Verhalten von Anfragen und Antworten für benutzerdefinierte Ursprungsserver 456
Wie CloudFront verarbeitet und leitet Anfragen an Ihren benutzerdefinierten Ursprung weiter 457
Wie CloudFront verarbeitet Antworten von Ihrem benutzerdefinierten Ursprung 477
Verhalten von Anfragen und Antworten für Ursprungsgruppen 482
Hinzufügen benutzerdefinierter Header zu Ursprungsanforderungen 483
Anwendungsfälle 484
Konfiguriere CloudFront , um benutzerdefinierte Header zu Quellanfragen hinzuzufügen 485
Benutzerdefinierte Header, die nicht zu CloudFront Quellanfragen hinzugefügt werden können 485
Konfigurieren Sie CloudFront , um den Authorization-Header weiterzuleiten 486
Wie CloudFront verarbeitet Bereichs-GETs 487
Verwenden von Bereichsanforderungen zum Zwischenspeichern großer Objekte 489
Wie CloudFront verarbeitet HTTP 3xx-Statuscodes von Ihrem Ursprung 490
Wie CloudFront verarbeitet die HTTP 4xx- und 5xx-Statuscodes von Ihrem Ursprung 490
Wie CloudFront verarbeitet Fehler, wenn Sie benutzerdefinierte Fehlerseiten konfiguriert haben 491
Wie CloudFront verarbeitet Fehler, wenn Sie keine benutzerdefinierten Fehlerseiten konfiguriert haben 494
HTTP-Statuscodes 4xx und 5xx, die zwischengespeichert werden CloudFront 496
Erstellen von benutzerdefinierten Fehlerantworten 498
Konfigurieren des Fehlerantwortverhaltens 499
Erstellen einer benutzerdefinierten Fehlerseite für bestimmte HTTP-Statuscodes 501
Speichern von Objekten und benutzerdefinierten Fehlerseiten an anderen Speicherorten ... 503
Ändern Sie die Antwortcodes, die zurückgegeben wurden von CloudFront 503
Steuern Sie, wie lange Fehler CloudFront zwischengespeichert werden 504
Hinzufügen, Entfernen oder Ersetzen von Inhalten 507
Hinzufügen von Inhalten und Zugriff darauf 507
Verwenden der Versionsverwaltung für Dateien, um vorhandenen Inhalt zu aktualisieren oder zu entfernen 508
Aktualisieren vorhandener Dateien mithilfe von versionierten Dateinamen 508
Inhalte entfernen, damit sie CloudFront nicht verteilt werden 509
Anpassen von Datei-URLs 509

Verwenden Ihres eigenen Domainnamens (example.com) 510
Verwenden eines abschließenden Schrägstrichs (/) in URLs 510
Erstellen signierter URLs für eingeschränkte Inhalte 511
Angeben eines Standardstammobjekts 511
So geben Sie ein Standardstammobjekt an 512
So funktioniert das Standardstammobjekt 514
Wie CloudFront funktioniert das, wenn Sie kein Root-Objekt definieren 515
Aufheben der Gültigkeit von Dateien zum Entfernen von Inhalten 516
Auswählen zwischen der Aufhebung der Gültigkeit von Dateien und der Verwendung von versionierten Dateinamen 516
Ermitteln, welche Pfade ungültig gemacht werden sollen 517
Was Sie zur Aufhebung der Gültigkeit von Pfaden wissen müssen 518
Aufhebung der Gültigkeit durch Cache-Tags 522
Was Sie zur Aufhebung der Gültigkeit von Tags wissen müssen 527
Aufheben der Gültigkeit von Dateien 529
Kosten der Aufhebung der Gültigkeit von Dateien 533
Bereitstellen von komprimierten Dateien 534
Konfigurieren Sie CloudFront , um Objekte zu komprimieren 534
Wie funktioniert die CloudFront Komprimierung 535
Bedingungen für die Komprimierung 536
Dateitypen, die komprimiert werden CloudFront 538
ETag-Header-Konvertierung 540
Verwenden Sie AWS WAF Schutzmaßnahmen 542
Aktivieren AWS WAF für Distributionen 543
Aktivieren AWS WAF für eine neue Distribution 543
Verwenden einer vorhandenen Web-ACL 544
Aktivieren von Bot Control 545
Konfigurieren des Schutzes nach Bot-Kategorie 546
Verwalten Sie die AWS WAF Sicherheitsvorkehrungen für CloudFront 547
Voraussetzungen 548
Aktivieren AWS WAF Protokolle 548
Festlegen der Ratenbegrenzung 549
Deaktivieren AWS WAF Sicherheitsvorkehrungen 550
Konfigurieren des sicheren Zugriffs und Beschränken des Zugriffs auf Inhalte 552
Verwenden Sie HTTPS mit CloudFront 553
Erfordert HTTPS zwischen Zuschauern und CloudFront 554

Erzwingen von HTTPS für einen benutzerdefinierten Ursprung ... 557
Erzwingen von HTTPS für einen Amazon-S3-Ursprung ... 560
Unterstützte Protokolle und Chiffren zwischen Viewern und CloudFront ... 562
Unterstützte Protokolle und Chiffren zwischen CloudFront und dem Ursprung ... 571
Verwenden von alternativen Domainnamen und HTTPS ... 573
Wählen Sie aus, wie CloudFront HTTPS-Anfragen verarbeitet werden ... 574
Anforderungen für die Verwendung von SSL/TLS Zertifikaten mit CloudFront ... 578
Kontingente für die Verwendung von SSL/TLS Zertifikaten mit CloudFront (HTTPS nur zwischen Zuschauern und CloudFront nur) ... 584
Konfigurieren alternativer Domainnamen und HTTPS ... 586
Ermitteln Sie die Größe des öffentlichen Schlüssels in einem SSL/TLS RSA-Zertifikat ... 590
Erhöhen Sie die Kontingente für Zertifikate SSL/TLS ... 591
SSL/TLS Zertifikate rotieren ... 592
Kehren Sie von einem benutzerdefinierten SSL/TLS Zertifikat zum Standardzertifikat zurück CloudFront ... 593
Wechseln Sie von einem benutzerdefinierten SSL/TLS Zertifikat mit dedizierten IP-Adressen zu SNI ... 595
Gegenseitige TLS-Authentifizierung mit CloudFront (Viewer mTLS) ... 596
Funktionsweise ... 596
Anwendungsfälle ... 597
Vertrauensspeicher und Zertifikatsverwaltung ... 597
Aktivieren Sie Mutual TLS für Distributionen CloudFront ... 605
Ordnen Sie eine CloudFront Verbindungsfunktion zu ... 609
Konfiguration zusätzlicher Einstellungen ... 615
Zeigt die mTLS-Header für Cache-Richtlinien an und leitet sie an den Ursprung weiter ... 617
Widerruf von Zertifikaten ... 621
Hilfsmethoden für gegenseitiges TLS ... 630
Zusätzliche Validierungsmodi ... 631
Beobachtbarkeit mithilfe von Verbindungsprotokollen ... 635
Origin Mutual TLS mit CloudFront ... 641
Viewer mTLS gegen Origin mTLS ... 641
Funktionsweise ... 641
Anwendungsfälle ... 642
Wichtig: Anforderungen an den Origin-Server ... 642
Erste Schritte ... 643
Zertifikatsverwaltung mit AWS Zertifikatsmanager ... 643

Aktivieren Sie Mutual Origin TLS für Distributionen CloudFront ... 647
Verwenden von CloudFront Funktionen mit gegenseitigem Ursprungs-TLS ... 651
Beschränken von Inhalten mit signierten URLs und signierten Cookies ... 654
So können Sie private Inhalte bereitstellen ... 654
Beschränken des Zugriffs auf Dateien ... 655
Angeben vertrauenswürdiger Signaturgeber ... 658
Entscheidung zwischen signierten URLs und signierter Cookies ... 669
Verwenden signierter URLs ... 670
Verwenden signierter Cookies ... 696
Linux-Befehle und OpenSSL für die Base64-Codierung und die Verschlüsselung ... 729
Code-Beispiele für signierte URLs ... 730
Beschränken Sie den Zugriff auf ein AWS Ursprungsserver ... 761
Beschränken Sie den Zugriff auf einen AWS Elemental MediaPackage v2-Ursprung ... 762
Beschränken Sie den Zugriff auf ein AWS Elemental MediaStore Ursprungsserver ... 770
Beschränken Sie den Zugriff auf ein AWS Lambda Herkunft der Funktions-URL ... 778
Beschränken Sie den Zugriff auf ein AWS Lambda Ursprung der Webfunktion ... 789
Beschränken des Zugriffs auf einen Amazon-S3-Ursprung ... 796
Zugriffsbeschränkung mit VPC-Ursprüngen ... 813
Beschränken Sie den Zugriff auf einen Amazon S3 Multi-Region Access Point-Ursprung ... 821
Beschränken des Zugriffs auf Application Load Balancer ... 828
Konfigurieren Sie CloudFront , um Anfragen einen benutzerdefinierten HTTP-Header hinzuzufügen ... 830
Konfigurieren eines Application Load Balancers zum Weiterleiten von Anforderungen, die einen bestimmten Header enthalten ... 832
(Optional) Verbesserung der Sicherheit dieser Lösung ... 833
(Optional) Beschränken Sie den Zugriff auf Origin mithilfe der AWS-verwaltete Präfixliste für CloudFront ... 835
Geografische Einschränkung ... 835
Verwenden Sie CloudFront geografische Einschränkungen ... 836
Verwenden des Geolokalisierungsdienstes eines Drittanbieters ... 838
Vertrauliche Daten durch Verschlüsselung auf Feldebene schützen ... 840
Überblick über die Verschlüsselung auf Feldebene ... 842
Einrichten der Verschlüsselung auf Feldebene ... 843
Entschlüsseln von Datenfeldern an Ihrem Ursprung ... 849
Video-on-Demand und Live-Video-Streaming ... 853
Informationen zum Streamen von Videos ... 853

Bereitstellen von Video-on-Demand ... 854
Konfigurieren von Video-on-Demand für Microsoft Smooth Streaming ... 855
Bereitstellen von Video-Streaming ... 857
Stellen Sie das Video bereit, indem Sie AWS Elemental MediaStore als Ursprung ... 858
Servieren Sie Live-Videos, formatiert mit AWS Elemental MediaPackage ... 859
Bieten Sie Video-on-Demand-Inhalte mit AWS Elemental MediaPackage ... 866
Media Quality-Aware Resiliency ... 872
MQAR-Protokollfelder ... 874
Verwenden von Funktionen zur Anpassung am Edge ... 875
Unterschiede zwischen CloudFront Functions und Lambda @Edge ... 876
Mit CloudFront Funktionen personalisieren ... 879
Tutorial: Erstelle eine einfache CloudFront Funktion ... 880
Tutorial: Erstellen Sie eine CloudFront Funktion, die Schlüsselwerte verwendet ... 883
Schreiben von Funktionscode ... 886
Erstellen von Funktionen ... 993
Testfunktionen ... 997
Aktualisieren von Funktionen ... 1002
Veröffentlichen von Funktionen ... 1005
Verknüpfen von Funktionen mit Distributionen ... 1006
CloudFront KeyValueStore ... 1010
Passen Sie es mit CloudFront Verbindungsfunktionen an ... 1033
Überblick und Arbeitsablauf ... 1034
Konfiguration und Grenzwerte ... 1036
Erstellen Sie CloudFront Verbindungsfunktionen für die gegenseitige TLS-Validierung (Viewer) ... 1037
Schreiben Sie den CloudFront Verbindungsfunktionscode für die gegenseitige TLS-Validierung (Viewer) ... 1041
Testen Sie die CloudFront Verbindungsfunktionen vor der Bereitstellung ... 1051
Ordnen Sie Verbindungsfunktionen Verteilungen zu ... 1053
Implementieren Sie den Widerruf von Zertifikaten für gegenseitiges TLS (Viewer) mit Functions und CloudFront KeyValueStore ... 1054
Anpassen mit Lambda@Edge ... 1062
So arbeitet Lambda@Edge mit Anforderungen und Antworten ... 1063
Anwendungsmöglichkeiten von Lambda@Edge ... 1063
Erste Schritte mit Lambda@Edge ... 1064
Einrichten von IAM-Berechtigungen und -Rollen ... 1073

Schreiben und Erstellen einer Lambda@Edge-Funktion 1080
Hinzufügen von Auslösern für eine Lambda@Edge-Funktion 1086
Testen und Debuggen 1094
Löschen von Funktionen und Replikaten 1102
Ereignisstruktur 1103
Arbeiten mit Anforderungen und Antworten 1120
Beispielfunktionen 1127
Einschränkungen für Edge-Funktionen 1166
Einschränkungen für alle Edge-Funktionen 1167
Einschränkungen bei Funktionen CloudFront 1173
Einschränkungen für Lambda@Edge 1176
Berichte, Metriken und Protokolle 1181
AWS Abrechnungs- und Nutzungsberichte für CloudFront 1181
Sehen Sie sich das an AWS Abrechnungsbericht für CloudFront 1182
Sehen Sie sich das an AWS Nutzungsbericht für CloudFront 1183
Interpretieren Sie Ihre AWS Rechnungs- und Nutzungsberichte für CloudFront 1185
CloudFront Konsolenberichte anzeigen 1193
Berichte zu CloudFront Cache-Statistiken anzeigen 1193
Berichte über CloudFront beliebte Objekte anzeigen 1200
Berichte mit den CloudFront häufigsten Referrern anzeigen 1206
CloudFront Nutzungsberichte anzeigen 1210
Berichte von CloudFront Zuschauern anzeigen 1219
Überwachen Sie CloudFront Metriken mit Amazon CloudWatch 1231
Funktionsmetriken anzeigen CloudFront und erweitern 1232
Erstellen von -Alarmen 1240
Herunterladen von Metrikdaten 1241
CloudFront Metriken 1245
CloudFront und Edge-Funktionsprotokollierung 1251
Protokollieren von Anfragen 1251
Protokollieren von Edge-Funktionen 1253
Protokollieren von Service-Aktivität 1253
Zugriffsprotokolle (Standardprotokolle) 1253
Verwenden Sie Zugriffsprotokolle in Echtzeit 1304
Protokolle für Edge-Funktionen 1328
AWS CloudTrail protokolliert 1332
Verfolgen Sie Konfigurationsänderungen mit AWS Config 1346

Richten Sie ein mit AWS Config CloudFront 1346
CloudFront Konfigurationshistorie anzeigen 1348
Evaluieren Sie CloudFront Konfigurationen mit AWS Config Regeln 1349
Sicherheit 1350
Datenschutz 1351
Verschlüsselung während der Übertragung 1352
Verschlüsselung im Ruhezustand 1353
Einschränken des Zugriffs auf Inhalte 1354
Identitäts- und Zugriffsverwaltung 1355
Zielgruppe 1355
Authentifizierung mit Identitäten 1356
Verwalten des Zugriffs mit Richtlinien 1357
Wie Amazon mit CloudFront IAM arbeitet 1359
Identity-based Beispiele für Richtlinien 1365
AWS verwaltete Richtlinien 1377
Serviceverknüpfte Rollen verwenden 1387
Probleme bei CloudFront Identität und Zugriff beheben 1392
Protokollierung und Überwachung 1394
Compliance-Validierung 1395
CloudFront Bewährte Verfahren zur Einhaltung 1396
Ausfallsicherheit 1398
Origin Failover in CloudFront 1398
Sicherheit der Infrastruktur 1398
Fehlerbehebung 1400
Fehlerbehebung bei Problemen mit der Distribution 1400
CloudFront gibt eine zurück Zugriff verweigert error 1400
CloudFront gibt eine zurück InvalidViewerCertificate Fehler beim Versuch, einen alternativen Domainnamen hinzuzufügen 1403
CloudFront gibt einen falsch konfigurierten DNS-Eintragsfehler zurück, wenn ich versuche, einen neuen CNAME hinzuzufügen 1405
Ich kann die Dateien in meiner Distribution nicht anzeigen. 1405
Fehlermeldung: Certificate: < certificate-id > wird verwendet von CloudFront 1407
Beheben von Fehlerstatuscodes 1408
HTTP 400-Statuscode (Bad Request) 1408
HTTP-Statuscode 401 (Nicht autorisiert) 1410
HTTP-Statuscode 403 (Ungültige Methode) 1411

HTTP-Statuscode 403 (Genehmigung verweigert) ... 1411
HTTP-Statuscode 404 (Nicht gefunden) ... 1414
HTTP-Statuscode 412 (Vorbedingung fehlgeschlagen) ... 1415
HTTP-Statuscode 500 (Interner Serverfehler) ... 1415
HTTP 502-Statuscode (Bad Gateway) ... 1416
HTTP 503-Statuscode (Service nicht verfügbar) ... 1421
HTTP 504-Statuscode (Gateway Timeout) ... 1425
Testen CloudFront ... 1430
Kontingente ... 1432
Allgemeine Kontingente ... 1433
Allgemeine Kontingente für Verteilungen ... 1434
Allgemeine Kontingente für Richtlinien ... 1436
Kontingente für mTLs und Trust Stores ... 1438
Kontingente für CloudFront Funktionen ... 1439
Kontingente für Verbindungsfunktionen ... 1439
Kontingente für Schlüsselwertspeicher ... 1440
Kontingente für Lambda@Edge ... 1441
Kontingente für SSL-Zertifikate ... 1443
Kontingente für Aufhebungen ... 1444
Kontingente für Schlüsselgruppen ... 1444
Kontingente für WebSocket Verbindungen ... 1445
Kontingente für Verschlüsselung auf Feldebene ... 1445
Kontingente für Cookies (Legacy-Cache-Einstellungen) ... 1446
Kontingente für Abfragezeichenfolgen (Legacy-Cache-Einstellungen) ... 1447
Kontingente für Header ... 1447
Kontingente zu Multi-Tenant-Distributionen ... 1448
Ähnliche Informationen ... 1450
Codebeispiele ... 1451
Grundlagen ... 1452
Aktionen ... 1453
Szenarien ... 1523
Erstellen einer Multi-Tenant-Distribution und eines Distributionsmandanten ... 1523
Löschen von Signaturressourcen ... 1535
Beginnen Sie mit einem grundlegenden Netzwerk zur Verteilung von Inhalten ... 1537
Erste Schritte mit Webanwendungs-Firewalls ... 1548
Signieren von URLs und Cookies ... 1560

CloudFront Funktionen, Beispiele 1564
Hinzufügen von HTTP-Sicherheitsheadern 1565
Hinzufügen eines CORS-Headers 1566
Hinzufügen eines Cache-Control-Headers 1567
Hinzufügen eines echten Client-IP-Headers 1568
Hinzufügen eines Origin-Headers 1568
Hinzufügen von index.html zu Anfrage-URLs 1569
Normalisieren von Abfragezeichenfolgeparametern 1570
Weiterleiten an eine neue URL 1571
Neuschreiben einer Anforderungs-URI 1573
Auswählen des dem Viewer näher gelegenen Ursprungs 1575
Verwenden von Schlüssel-Wert-Paaren 1577
Validieren eines einfachen Tokens 1578
Dokumentverlauf 1583
..... mdcxvi

Was ist Amazon CloudFront?

Amazon CloudFront ist ein Webservice, der die Verteilung Ihrer statischen und dynamischen Webinhalte wie .html, .css, .js und Bilddateien an Ihre Nutzer beschleunigt. CloudFront stellt Ihre Inhalte über ein weltweites Netzwerk von Rechenzentren bereit, die als Edge-Standorte bezeichnet werden. Wenn ein Benutzer Inhalte anfordert, die Sie bereitstellen CloudFront, wird die Anfrage an den Edge-Standort weitergeleitet, der die niedrigste Latenz (Zeitverzögerung) bietet, sodass die Inhalte mit der bestmöglichen Leistung bereitgestellt werden.

- Wenn sich der Inhalt bereits am Edge-Standort mit der niedrigsten Latenz befindet, wird er CloudFront sofort bereitgestellt.
- Wenn sich der Inhalt nicht an diesem Edge-Standort befindet, CloudFront ruft er ihn von einem von Ihnen definierten Ursprung ab, z. B. einem Amazon S3-Bucket, einem MediaPackage Channel oder einem HTTP-Server (z. B. einem Webserver), den Sie als Quelle für die endgültige Version Ihres Inhalts identifiziert haben.

Nehmen wir an, dass Sie ein Bild über einen traditionellen Webserver und nicht mit CloudFront bereitstellen. Sie können beispielsweise ein Bild, sunsetphoto.png, über die URL bereitstelle `https://example.com/sunsetphoto.png`.

Ihre Benutzer können problemlos zu dieser URL navigieren und das Bild anzeigen. Aber sie wissen wahrscheinlich nicht, dass ihre Anfrage von einem Netzwerk zu einem anderen weitergeleitet wird, bis das Bild schließlich gefunden wurde – durch die komplexe Sammlung miteinander verbundener Netzwerke, aus dem das Internet besteht.

CloudFront beschleunigt die Verteilung Ihrer Inhalte, indem jede Benutzeranfrage über das AWS Backbone-Netzwerk an den Edge-Standort weitergeleitet wird, der Ihre Inhalte am besten bereitstellen kann. In der Regel handelt es sich dabei um einen CloudFront Edge-Server, der dem Zuschauer die schnellste Übertragung bietet. Durch die Verwendung des AWS Netzwerks wird die Anzahl der Netzwerke, die die Anfragen Ihrer Benutzer durchlaufen müssen, drastisch reduziert, wodurch die Leistung verbessert wird. Benutzer erfahren eine geringere Latenz – die Zeit bis zum Laden des ersten Bytes der Datei – und erzielen höhere Datenübertragungsraten.

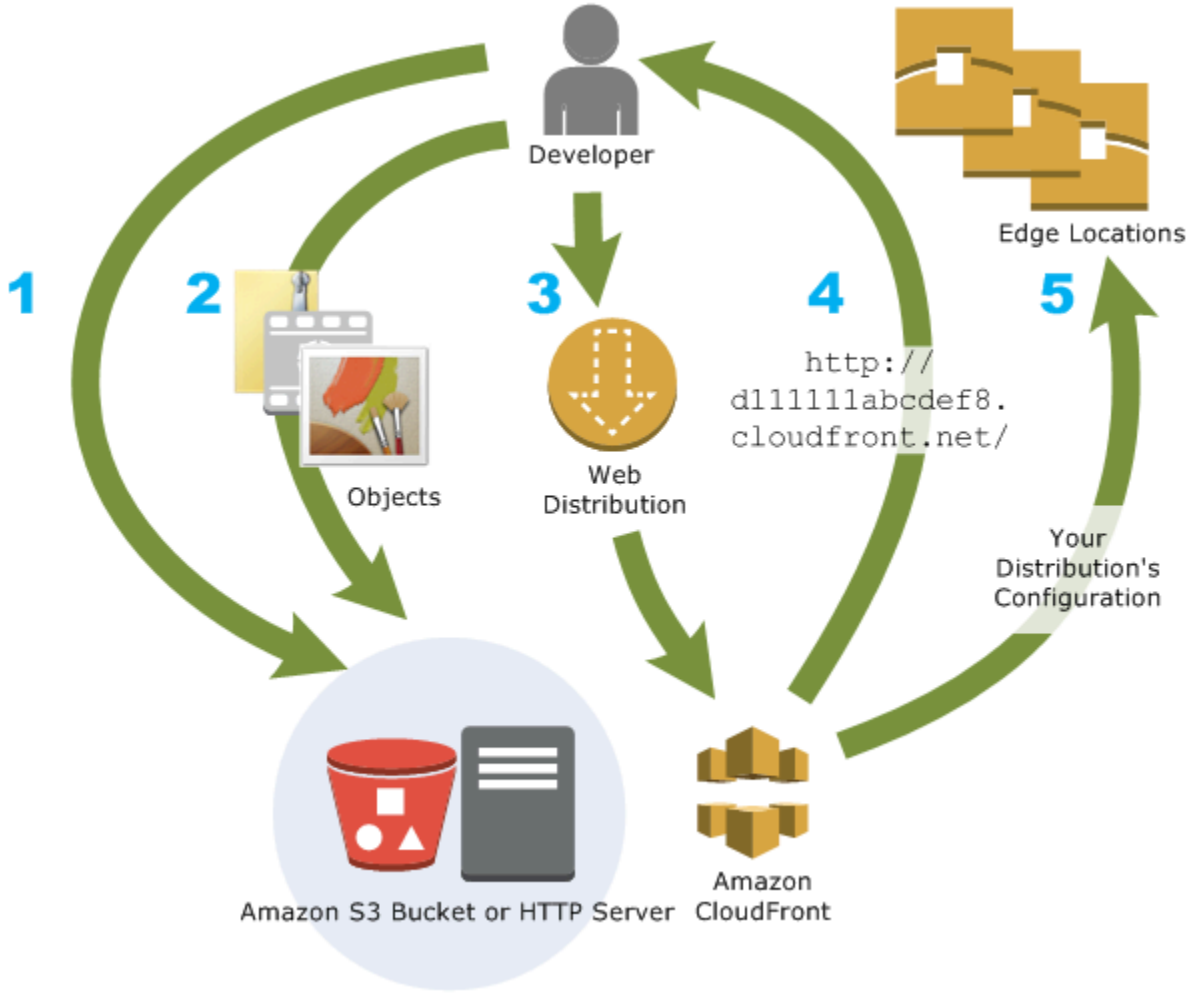
Darüber hinaus erhöht sich die Zuverlässigkeit und die Verfügbarkeit, da Kopien Ihrer Dateien (auch als Objekte bezeichnet) jetzt an mehreren Edge-Standorten auf der ganzen Welt verfügbar (oder dort im Cache gespeichert) sind.

Themen

- Wie richten Sie CloudFront die Bereitstellung von Inhalten ein
- Auswahl zwischen Standarddistribution und Multi-Tenant-Distribution
- Preisgestaltung
- Möglichkeiten zur Verwendung CloudFront
- Wie liefert Inhalte CloudFront
- Standorte und IP-Adressbereiche von CloudFront Edge-Servern
- Verwendung CloudFront mit einem AWS SDK
- CloudFront technische Ressourcen

Wie richten Sie CloudFront die Bereitstellung von Inhalten ein

Sie erstellen eine CloudFront Distribution, in der Sie CloudFront angeben, von wo die Inhalte geliefert werden sollen, und um zu erfahren, wie Sie die Bereitstellung von Inhalten verfolgen und verwalten können. Anschließend werden Computer — Edge-Server — CloudFront verwendet, die sich in der Nähe Ihrer Zuschauer befinden, um die Inhalte schnell bereitzustellen, wenn jemand sie sehen oder verwenden möchte.



Wie konfigurierst du CloudFront die Bereitstellung deiner Inhalte

1. Sie geben Ursprungsserver an, z. B. einen Amazon S3-Bucket oder Ihren eigenen HTTP-Server, von denen CloudFront Ihre Dateien abgerufen werden, die dann von CloudFront Edge-Standorten auf der ganzen Welt verteilt werden.

   Auf einem Ursprungsserver sind die originalen, definitiven Versionen Ihrer Dateien gespeichert. Wenn Sie Inhalte über HTTP bereitstellen, ist Ihr Ursprungs-Server entweder ein Amazon S3-Bucket oder ein HTTP-Server, z. B. ein Web-Server. Ihr HTTP-Server kann auf einer Amazon Elastic Compute Cloud-Instance (Amazon EC2) oder auf einem von Ihnen verwalteten Server ausgeführt werden. Diese Server werden auch als benutzerdefinierte Ursprünge bezeichnet.
2. Sie laden Ihre Dateien auf Ihre Ursprungsserver hoch. Ihre Dateien, auch als Objekte bezeichnet, enthalten normalerweise Webseiten, Bilder und Mediendateien, können jedoch alles sein, was über HTTP bereitgestellt werden kann.

Wenn Sie einen Amazon S3-Bucket als Ursprungsserver verwenden, können Sie die Objekte in Ihrem Bucket öffentlich lesbar machen, sodass jeder, der die CloudFront URLs für Ihre Objekte kennt, darauf zugreifen kann. Sie haben auch die Möglichkeit, private Objekte einzurichten und den Zugriff zu kontrollieren. Siehe Bereitstellen privater Inhalte mit signierten URLs und signierten Cookies.

3. Sie erstellen eine CloudFront Verteilung, die angibt, von CloudFront welchen Ursprungsservern Ihre Dateien abgerufen werden sollen, wenn Benutzer die Dateien über Ihre Website oder Anwendung anfordern. Gleichzeitig geben Sie Details an, z. B. ob Sie alle Anfragen protokollieren CloudFront möchten und ob die Distribution aktiviert werden soll, sobald sie erstellt wurde.
4. CloudFront weist Ihrer neuen Distribution einen Domainnamen zu, den Sie in der CloudFront Konsole sehen können oder der in der Antwort auf eine programmatische Anfrage, z. B. eine API-Anfrage, zurückgegeben wird. Wenn Sie möchten, können Sie einen alternativen Domänennamen hinzufügen, sodass er stattdessen verwendet wird.
5. CloudFront sendet die Konfiguration Ihrer Distribution (aber nicht Ihre Inhalte) an alle Edge-Standorte oder Points of Presence (POPs) — Ansammlungen von Servern in geografisch verteilten Rechenzentren, in denen Kopien Ihrer Dateien zwischengespeichert werden. CloudFront

Bei der Entwicklung Ihrer Website oder Anwendung verwenden Sie den Domainnamen, der Ihre URLs bereitstellt. CloudFront Wenn `d111111abcdef8.cloudfront.net` beispielsweise der Domainname für Ihre Distribution CloudFront zurückgegeben wird, lautet die URL für logo.jpg in Ihrem Amazon S3-Bucket (oder im Stammverzeichnis auf einem HTTP-Server)`https://d111111abcdef8.cloudfront.net/logo.jpg`.

Sie können auch einrichten CloudFront , dass Ihr eigener Domainname für Ihre Distribution verwendet wird. In diesem Fall könnte die URL folgendermaßen lauten: `https://www.example.com/logo.jpg`.

Optional können Sie Ihren Ursprungsserver so konfigurieren, dass er den Dateien Header hinzufügt, um anzugeben, wie lange die Dateien an CloudFront Edge-Speicherorten im Cache verbleiben sollen. Standardmäßig wird jede Datei 24 Stunden lang an einem Edge-Standort gespeichert; danach läuft sie ab. Die minimale Ablaufzeit beträgt 0 Sekunden; eine maximale Ablaufzeit existiert nicht. Weitere Informationen finden Sie unter Verwalten der Dauer, die Inhalte im Cache bleiben (Ablauf).

Auswahl zwischen Standarddistribution und Multi-Tenant-Distribution

CloudFront bietet Verteilungsoptionen für einzelne Websites oder Apps sowie für Szenarien mit mehreren Mandanten.

Standarddistribution

Konzipiert für einzigartige Konfigurationen pro Website oder Anwendung. Wählen Sie diese Option in den folgenden Anwendungsfällen:

- Sie benötigen eine eigenständige Distribution CloudFront
- Jede Website oder Anwendung benötigt ihre eigenen benutzerdefinierten Einstellungen.

Die meisten beginnen mit einer Standarddistribution.

Multi-tenant Mieter für Vertrieb und Vertrieb (CloudFront SaaS Manager)

Speziell für SaaS-Anbieter und Multi-Tenant-Szenarien konzipiert. Wählen Sie diese Option in den folgenden Anwendungsfällen:

- Sie entwickeln eine SaaS-Plattform, um mehrere Kundenwebsites oder -anwendungen zu bedienen.
- Sie müssen mehrere ähnliche Distributionen effizient verwalten.
- Sie möchten eine zentrale Steuerung gemeinsam genutzter Konfigurationen.

Weitere Informationen finden Sie unter Verständnis der Funktionsweise von Multi-Tenant-Distributionen.

Preisgestaltung

CloudFront berechnet Gebühren für Datenübertragungen von seinen Edge-Standorten aus zusammen mit HTTP- oder HTTPS-Anfragen. Die Preise variieren je nach Nutzungsart, geografischer Region und Funktionsauswahl.

Die Datenübertragung von Ihrem Ursprung zu CloudFront ist immer kostenlos, wenn Sie AWS Quellen wie Amazon Simple Storage Service (Amazon S3), Elastic Load Balancing oder Amazon API Gateway verwenden. Wenn Sie Origins verwenden AWS , wird Ihnen nur die ausgehende Datenübertragung von CloudFront zum Viewer in Rechnung gestellt.

Weitere Informationen finden Sie unter CloudFront Preise und in den häufig gestellten Fragen zum Abrechnungs- und Sparpaket.

Möglichkeiten zur Verwendung CloudFront

Die Verwendung CloudFront kann Ihnen helfen, eine Vielzahl von Zielen zu erreichen. In diesem Abschnitt werden nur einige aufgelistet, zusammen mit Links zu weiteren Informationen, um Ihnen eine Vorstellung davon zu geben, welche Möglichkeiten Sie haben.

Themen

- Beschleunigen der Bereitstellung von Inhalten auf einer statischen Website
- Bereitstellen von On-Demand- oder Live-Streaming-Video
- Verschlüsseln von bestimmten Feldern während der Systemverarbeitung
- An der Grenze anpassen
- Bereitstellen von privaten Inhalten mit Hilfe von Lambda@Edge-Anpassungen

Beschleunigen der Bereitstellung von Inhalten auf einer statischen Website

CloudFront kann die Bereitstellung Ihrer statischen Inhalte (z. B. Bilder, Stylesheets usw.) für Zuschauer auf der ganzen Welt beschleunigen. JavaScript Durch die Verwendung CloudFront können Sie das AWS Backbone-Netzwerk und die CloudFront Edge-Server nutzen, um Ihren Zuschauern ein schnelles, sicheres und zuverlässiges Erlebnis zu bieten, wenn sie Ihre Website besuchen.

Ein einfacher Ansatz für das Speichern und Bereitstellen von statischen Inhalten ist die Verwendung eines Amazon S3-Buckets. Die gemeinsame Verwendung von S3 CloudFront bietet eine Reihe von Vorteilen, darunter die Option, Origin Access Control zu verwenden, um den Zugriff auf Ihre Amazon S3-Inhalte einfach einzuschränken.

Weitere Informationen zur Verwendung von Amazon S3 zusammen mit Amazon S3 CloudFront, einschließlich einer CloudFormation Vorlage, die Ihnen den Einstieg erleichtert, finden Sie unterErste Schritte mit einer sicheren statischen Website.

Bereitstellen von On-Demand- oder Live-Streaming-Video

CloudFront bietet verschiedene Optionen für das Streamen Ihrer Medien an globale Zuschauer — sowohl aufgezeichnete Dateien als auch Live-Events.

- Für Video-on-Demand-Streaming (VOD) können Sie es verwenden, CloudFront um in gängigen Formaten wie MPEG DASH, Apple HLS, Microsoft Smooth Streaming und CMAF auf jedes Gerät zu streamen.
- Für das Broadcasting eines Live-Streams können Sie Medienfragmente am Edge-Standort zwischenspeichern, sodass mehrere Anforderungen an die Manifestdatei, die die Fragmente in der richtigen Reihenfolge bereitstellt, kombiniert werden können, um den Workload auf Ihrem Ursprungs-Server zu verringern.

Weitere Informationen zur Bereitstellung von Streaming-Inhalten mit finden Sie unter. CloudFront Video auf Abruf und Live-Streaming-Video mit CloudFront

Verschlüsseln von bestimmten Feldern während der Systemverarbeitung

Wenn Sie HTTPS mit konfigurieren CloudFront, verfügen Sie bereits über sichere Ende-zu-Ende-Verbindungen zu den Ursprungsservern. Wenn Sie eine Verschlüsselung auf Feldebene hinzufügen, können Sie neben der HTTPS-Sicherheit bestimmte Daten während der gesamten Systemverarbeitung schützen, sodass nur bestimmte Anwendungen an ihrem Ursprung die Daten sehen können.

Um die Verschlüsselung auf Feldebene einzurichten, fügen Sie einen öffentlichen Schlüssel hinzu CloudFront und geben dann die Gruppe von Feldern an, die mit dem Schlüssel verschlüsselt werden sollen. Weitere Informationen finden Sie unter Vertrauliche Daten durch Verschlüsselung auf Feldebene schützen.

An der Grenze anpassen

Durch das Ausführen von serverlosem Code am Edge-Standort ergeben sich eine Reihe von Möglichkeiten zum Anpassen der Inhalte und Erfahrungen für Betrachter, bei reduzierter Latenz. Sie können zum Beispiel eine benutzerdefinierte Fehlermeldung zurückgeben, wenn Ihr Ursprungs-Server wegen Wartungsarbeiten nicht verfügbar ist, damit Betrachter keine allgemeine HTTP-Fehlermeldung erhalten. Sie können auch eine Funktion verwenden, um Benutzer zu autorisieren und den Zugriff auf Ihre Inhalte zu kontrollieren, bevor Sie eine Anfrage an Ihren CloudFront Absender weiterleiten.

Die Verwendung von Lambda @Edge mit CloudFront ermöglicht eine Vielzahl von Möglichkeiten, den Inhalt, der bereitgestellt wird, anzupassen. CloudFront Weitere Informationen zu Lambda@Edge und dazu, wie Sie Funktionen mit CloudFront erstellen und bereitstellen können, finden Sie unter

Anpassen am Edge mit Lambda@Edge. Eine Reihe von Codebeispielen, die Sie für Ihre eigenen Lösungen anpassen können, finden Sie unter Beispielfunktionen für Lambda@Edge.

Bereitstellen von privaten Inhalten mit Hilfe von Lambda@Edge-Anpassungen

Mithilfe von Lambda @Edge können Sie Ihre CloudFront Distribution so konfigurieren, dass neben signierten URLs oder signierten Cookies auch private Inhalte von Ihrem eigenen, benutzerdefinierten Ursprung bereitgestellt werden.

Gehen Sie wie folgt vor CloudFront, um private Inhalte bereitzustellen:

- Verlangen Sie, dass Ihre Benutzer (Viewer) über signierte URLs oder signierte Cookies auf Inhalte zugreifen.
- Beschränken Sie den Zugriff auf Ihren Ursprungsserver, sodass er nur von Servern aus verfügbar ist, die CloudFront mit dem Absender verbunden sind. Dazu können Sie einen der folgenden Schritte ausführen:
  - Für einen Amazon-S3-Ursprung können Sie eine Ursprungszugriffssteuerung (OAC) verwenden.
  - Für einen benutzerdefinierten Ursprung können Sie folgendermaßen vorgehen:
    - Wenn der benutzerdefinierte Ursprung durch eine Amazon VPC-Sicherheitsgruppe oder eine Amazon VPC-Sicherheitsgruppe geschützt ist AWS Firewall Manager, können Sie die CloudFront verwaltete Präfixliste verwenden, um eingehenden Datenverkehr nur CloudFront von IP-Adressen zu Ihrem Ursprung zuzulassen.
    - Verwenden Sie einen benutzerdefinierten HTTP-Header, um den Zugriff nur auf Anfragen von zu beschränken. CloudFront Weitere Informationen erhalten Sie unter the section called “Beschränken des Zugriffs auf Dateien auf benutzerdefinierten Ursprungsservern” und the section called “Hinzufügen benutzerdefinierter Header zu Ursprungsanforderungen”. Ein Beispiel, das einen benutzerdefinierten Header verwendet, um den Zugriff auf einen Application-Load-Balancer-Ursprung einzuschränken, finden Sie unter the section called “Beschränken des Zugriffs auf Application Load Balancer”.
    - Wenn der benutzerdefinierte Ursprung eine benutzerdefinierte Zugriffskontrolllogik erfordert, können Sie Lambda @Edge verwenden, um diese Logik zu implementieren, wie in diesem Blogbeitrag beschrieben: Bereitstellung privater Inhalte mit Amazon CloudFront und Lambda @Edge.

Wie liefert Inhalte CloudFront

Nach einer gewissen Ersteinrichtung CloudFront arbeitet es mit Ihrer Website oder Anwendung zusammen und beschleunigt die Bereitstellung Ihrer Inhalte. In diesem Abschnitt wird erklärt, CloudFront wie deine Inhalte bereitgestellt werden, wenn Zuschauer sie anfordern.

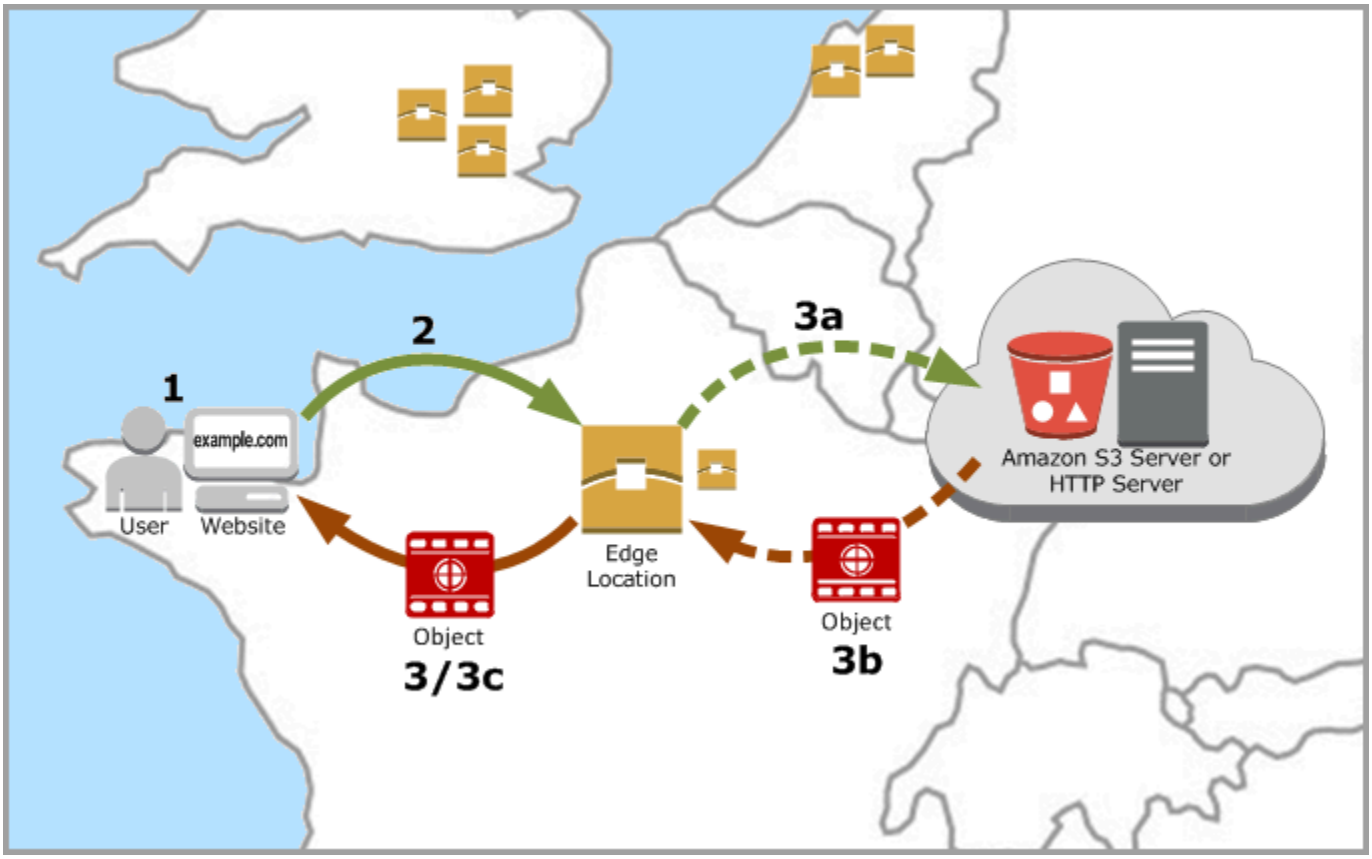
Themen

- Wie CloudFront werden Inhalte an Ihre Nutzer geliefert
- Wie CloudFront funktioniert das mit regionalen Edge-Caches

Wie CloudFront werden Inhalte an Ihre Nutzer geliefert

Nachdem Sie CloudFront die Bereitstellung Ihrer Inhalte konfiguriert haben, passiert Folgendes, wenn Benutzer Ihre Objekte anfordern:

1. Ein Benutzer greift auf Ihre Website oder Anwendung zu und sendet eine Anforderung für ein Objekt, z. B. eine Bilddatei und eine HTML-Datei.
2. DNS leitet die Anfrage an den CloudFront POP (Edge-Standort) weiter, der die Anfrage am besten bearbeiten kann. In Bezug auf die Latenz ist dies in der Regel der CloudFront POP, der am nächsten liegt.
3. CloudFront überprüft seinen Cache auf das angeforderte Objekt. Wenn sich das Objekt im Cache befindet, wird es an den Benutzer CloudFront zurückgegeben. Wenn sich das Objekt nicht im Cache befindet, gehen CloudFront Sie wie folgt vor:
   a. CloudFront vergleicht die Anfrage mit den Spezifikationen in Ihrer Distribution und leitet die Anfrage für das entsprechende Objekt an Ihren Ursprungsserver weiter — zum Beispiel an Ihren Amazon S3-Bucket oder Ihren HTTP-Server.
   b. Der Ursprungsserver sendet das Objekt an den Edge-Standort zurück.
   c. Sobald das erste Byte vom Ursprung eintrifft, CloudFront beginnt die Weiterleitung des Objekts an den Benutzer. CloudFront fügt das Objekt auch dem Cache hinzu, damit es das nächste Mal von jemandem angefordert wird.



Wie CloudFront funktioniert das mit regionalen Edge-Caches

CloudFront Präsenzpunkte (auch bekannt als POPs oder Edge-Standorte) sorgen dafür, dass beliebte Inhalte Ihren Zuschauern schnell zur Verfügung gestellt werden können. CloudFront verfügt außerdem über regionale Edge-Caches, die Ihren Zuschauern mehr von Ihren Inhalten näher bringen, auch wenn die Inhalte nicht beliebt genug sind, um bei einem POP zu bleiben, um die Leistung dieser Inhalte zu verbessern.

Regionale Edge-Caches sind für alle Arten von Inhalten nützlich, insbesondere wenn die Inhalte im Laufe der Zeit immer seltener abgerufen werden. Dazu gehören von Benutzern erstellte Inhalte wie Videos, Fotos oder Bildmaterial, E-Commerce-Elemente wie Produktfotos und -videos sowie Nachrichten und ereignisbezogene Inhalte, die plötzlich sehr stark angefragt werden können.

So funktionieren regionale Caches

Regionale Edge-Caches sind CloudFront Standorte, die weltweit in der Nähe Ihrer Zuschauer eingesetzt werden. Sie befinden sich zwischen Ihrem Ursprungsserver und den weltweit verteilten POPs. Diese Edge-Standorte stellen Inhalte Ihren Viewern direkt bereit. Wenn die Häufigkeit der Anfragen für Ihre Objekte abnimmt, werden diese ggf. an einzelnen POPs entfernt, um Platz für

beliebtere Inhalte zu schaffen. Regionale Edge-Caches verfügen über mehr Cache-Speicherplatz als ein einzelner POP, sodass Ihre Objekte am nächstgelegenen Standort eines regionalen Edge-Cache länger gespeichert werden. Auf diese Weise können Sie Ihren Zuschauern mehr Inhalte näher bringen, sodass Sie nicht mehr CloudFront zu Ihrem Ursprungsserver zurückkehren müssen, und die Gesamtleistung für die Zuschauer wird verbessert.

Wenn ein Betrachter eine Anfrage auf Ihrer Website oder über Ihre Anwendung sendet, leitet DNS die Anfrage an den POP weiter, der die Anforderung des Benutzers am besten bedienen kann. In Bezug auf die Latenz ist dieser Standort in der Regel der nächstgelegene CloudFront Edge-Standort. CloudFrontÜberprüft im POP den Cache auf das angeforderte Objekt. Wenn sich das Objekt im Cache befindet, wird es an den Benutzer CloudFront zurückgegeben. Wenn das Objekt nicht im Cache vorhanden ist, wechseln die POPs zum nächstgelegenen regionalen Edge-Cache, um es abzurufen. Weitere Informationen darüber, wann der POP den regionalen Edge-Cache überspringt und direkt zum Ursprung wechselt, finden Sie in der folgenden Anmerkung.

Überprüft am Standort des regionalen Edge-Cache CloudFront erneut den Cache auf das angeforderte Objekt. Wenn sich das Objekt im Cache befindet, CloudFront leitet es an den POP weiter, der es angefordert hat. Sobald das erste Byte vom regionalen Edge-Cache-Standort eintrifft, CloudFront beginnt die Weiterleitung des Objekts an den Benutzer. CloudFront fügt das Objekt außerdem dem Cache im POP hinzu, damit es das nächste Mal von jemandem angefordert wird.

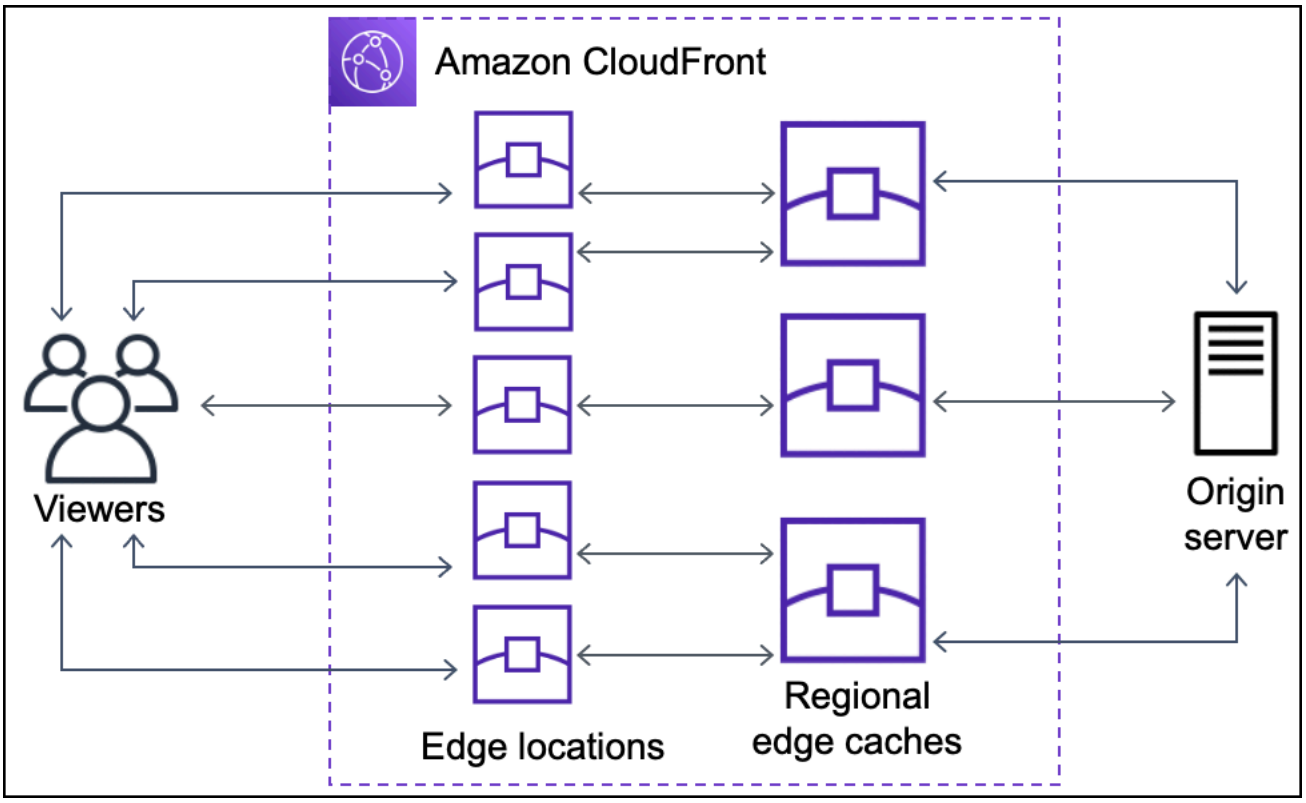
CloudFront Vergleicht bei Objekten, die weder am POP- noch am regionalen Edge-Cache-Standort zwischengespeichert sind, die Anforderung mit den Spezifikationen in Ihren Distributionen und leitet die Anfrage an den Originalserver weiter. Nachdem Ihr Ursprungsserver das Objekt an den regionalen Edge-Cache-Standort zurückgesendet hat, wird es an den POP weitergeleitet und dann an den CloudFront Benutzer weitergeleitet. In diesem Fall wird das Objekt zusätzlich zum POP CloudFront auch dem Cache am regionalen Edge-Cache-Standort hinzugefügt, damit es das nächste Mal von einem Betrachter angefordert wird. Dadurch wird sichergestellt, dass sich alle POPs in einer Region einen lokalen Cache teilen, wodurch mehrere Anfragen an die Ursprungsserver vermieden werden. CloudFront hält auch persistente Verbindungen zu den Ursprungsservern aufrecht, sodass Objekte so schnell wie möglich von den Ursprüngen abgerufen werden.

> **Note**
>
> - Regionale Edge-Caches haben denselben Funktionsumfang wie POPs. Eine Cache-Invalidierungsanforderung entfernt ein Objekt z. B. sowohl aus POP-Caches als auch aus regionalen Edge-Caches, bevor es abgelaufen ist. Wenn einen Betrachter das Objekt das

nächste Mal anfordert, bezieht CloudFront die neueste Version des Objekts wieder von dem Ursprungsserver.

- Die Proxy-HTTP-Methoden (`PUT`, `POST`, `PATCH`, `OPTIONS` und `DELETE`) werden von den POPs direkt an den Ursprungsserver gesendet und nicht über die regionalen Edge-Caches geleitet.
- Dynamische Anforderungen, die zum Anforderungszeitpunkt bestimmt werden, fließen nicht durch regionale Edge-Caches, sondern gehen direkt zum Ursprung.
- Wenn der Ursprung ein Amazon S3-Bucket ist und sich der optimale regionale Edge-Cache der Anfrage im selben AWS-Region wie der S3-Bucket befindet, überspringt der POP den regionalen Edge-Cache und geht direkt zum S3-Bucket.

Das folgende Diagramm veranschaulicht, wie Anfragen und Antworten durch CloudFront Edge-Standorte und regionale Edge-Caches fließen.



Standorte und IP-Adressbereiche von CloudFront Edge-Servern

Eine Liste der Standorte der CloudFront Edge-Server finden Sie auf der Amazon CloudFront Global Edge Network-Seite.

Amazon Web Services (AWS) veröffentlicht seine aktuellen IP-Adressbereiche im JSON-Format. Laden Sie die Datei ip-ranges.json herunter, um die aktuellen Bereiche anzuzeigen. Weitere Informationen finden Sie unter AWS IP-Adressbereiche im Allgemeine Amazon Web Services-Referenz.

Um die IP-Adressbereiche zu finden, die CloudFront Edge-Servern zugeordnet sind, suchen Sie in ip-ranges.json nach der folgenden Zeichenfolge:

```
"region": "GLOBAL",
"service": "CLOUDFRONT"
```

Alternativ können Sie nur die IP-Bereiche unter einsehen. CloudFront https://d7uri8nf7uskq.cloudfront.net/tools/list-cloudfront-ips

Verwenden Sie die Liste der CloudFront verwalteten Präfixe

Die Liste der CloudFront verwalteten Präfixe enthält die IP-Adressbereiche aller CloudFront global verteilten Server mit Ursprungsadresse. Wenn Ihr Ursprung auf einer Amazon VPC-Sicherheitsgruppe gehostet AWS und von dieser geschützt wird, können Sie die CloudFront verwaltete Präfixliste verwenden. Die Präfixliste lässt eingehenden Datenverkehr zu Ihrem Ursprung nur von Servern zu, die CloudFront mit Ihrem Ursprung verbunden sind. Dadurch wird verhindert, dass CloudFront Datenverkehr, der nicht zu Ihrem Absender gehört, Ihren Ursprung erreicht. CloudFront verwaltet die Liste der verwalteten Präfixe, sodass sie immer auf dem neuesten Stand ist, was die IP-Adressen aller CloudFront globalen Ursprungsserver angeht. Mit der Liste der CloudFront verwalteten Präfixe müssen Sie keine Liste von IP-Adressbereichen selbst lesen oder verwalten.

Stellen Sie sich zum Beispiel vor, dass Ihr Ursprung eine Amazon-EC2-Instance in der Region Europa (London) (`eu-west-2`) ist. Wenn sich die Instance in einer VPC befindet, können Sie eine Sicherheitsgruppenregel erstellen, die eingehenden HTTPS-Zugriff über die Liste der CloudFront verwalteten Präfixe zulässt. Dadurch können alle CloudFront globalen Server, die mit dem Ursprung verbunden sind, die Instance erreichen. Wenn Sie alle anderen Regeln für eingehenden Verkehr aus der Sicherheitsgruppe entfernen, verhindern Sie, dass jeglicher CloudFront Datenverkehr die Instance erreicht.

Die CloudFront verwalteten Präfixlisten lauten wie folgt:

- `com.amazonaws.global.cloudfront.origin-facing` (IPv4)
- `com.amazonaws.global.ipv6.cloudfront.origin-facing` (IPv6)

Weitere Informationen finden Sie unter Verwenden einer AWS verwalteten Präfixliste im Amazon VPC-Benutzerhandbuch.

> **Important**
>
> Die Liste der CloudFront verwalteten Präfixe ist insofern einzigartig, als sie für Amazon VPC-Kontingente gilt. Weitere Informationen finden Sie im Abschnitt zur Gewichtung der AWS-verwalteten Präfixliste im Amazon-VPC-Benutzerhandbuch.

Verwendung CloudFront mit einem AWS SDK

AWS Software Development Kits (SDKs) sind für viele gängige Programmiersprachen verfügbar. Jedes SDK bietet eine API, Codebeispiele und Dokumentation, die es Entwicklern erleichtern, Anwendungen in ihrer bevorzugten Sprache zu erstellen.

SDK-Dokumentation	Codebeispiele
AWS SDK für C++	AWS SDK für C++ Code-Beispiele
AWS CLI	AWS CLI Code-Beispiele
AWS SDK für Go	AWS SDK für Go Code-Beispiele
AWS SDK für Java	AWS SDK für Java Code-Beispiele
AWS SDK für JavaScript	AWS SDK für JavaScript Code-Beispiele
AWS SDK für Kotlin	AWS SDK für Kotlin Code-Beispiele
AWS SDK für .NET	AWS SDK für .NET Code-Beispiele
AWS SDK für PHP	AWS SDK für PHP Code-Beispiele
AWS -Tools für PowerShell	AWS -Tools für PowerShell Code-Beispiele
AWS SDK für Python v4	AWS SDK für Python v4-Codebeispiele
AWS SDK für Python (Boto3)	AWS SDK für Python (Boto3) Codebeispiele

SDK-Dokumentation	Codebeispiele
AWS SDK für Ruby	AWS SDK für Ruby Code-Beispiele
AWS SDK für Rust	AWS SDK für Rust Code-Beispiele
AWS SDK für SAP ABAP	AWS SDK für SAP ABAP Code-Beispiele
AWS SDK für Swift	AWS SDK für Swift Code-Beispiele

> **Beispiel für die Verfügbarkeit**
>
> Sie können nicht finden, was Sie brauchen? Fordern Sie ein Codebeispiel an, indem Sie unten den Link Feedback geben auswählen.

CloudFront technische Ressourcen

Verwenden Sie die folgenden Ressourcen, um Antworten auf technische Fragen zu folgenden Themen zu erhalten CloudFront:

- AWS re:POST — Eine Community-Website mit Fragen und Antworten, auf der Entwickler technische Fragen zu folgenden Themen erörtern können. CloudFront
- Support Center — Diese Website enthält Informationen zu Ihren letzten Support-Anfragen sowie zu den Ergebnissen und Zustandsprüfungen. AWS Trusted Advisor Sie stellt auch Links zu Diskussionsforen, technischen FAQs und dem Servicezustands-Dashboard sowie Informationen zu Support -Tarifen bereit.
- AWS Premium-Support — Erfahren Sie mehr über den AWS Premium-Support, einen persönlichen, schnell reagierenden Supportkanal, der Sie bei der Entwicklung und Ausführung von Anwendungen unterstützt. AWS

Fangen Sie an mit CloudFront

Die Themen in diesem Abschnitt zeigen Ihnen, wie Sie mit der Bereitstellung Ihrer Inhalte über Amazon beginnen können CloudFront.

In Richten Sie Ihre ein AWS-Konto diesem Thema werden die Voraussetzungen für die folgenden Tutorials beschrieben, z. B. das Erstellen eines Benutzers AWS-Konto und das Erstellen eines Benutzers mit Administratorzugriff.

Das Tutorial zur grundlegenden Distribution zeigt Ihnen, wie Sie die Ursprungszugriffssteuerung (OAC) einrichten, um authentifizierte Anforderungen an einen Amazon-S3-Ursprung zu senden.

Im Tutorial zur sicheren statischen Website erfahren Sie, wie Sie mithilfe von OAC mit einem Amazon-S3-Ursprung eine sichere statische Website für Ihren Domainnamen erstellen. Das Tutorial verwendet eine Amazon CloudFront (CloudFront) -Vorlage für Konfiguration und Bereitstellung.

Das AWS CLI Tutorial zeigt Ihnen, wie Sie eine Standarddistribution mithilfe von AWS Command Line Interface (AWS CLI) erstellen, aktualisieren und löschen. Weitere Informationen finden Sie unter Erste Schritte mit einer Standarddistribution (AWS CLI).

Themen

- Richten Sie Ihre ein AWS-Konto
- Beginnen Sie mit einer CloudFront Standarddistribution
- Erste Schritte mit einer Standarddistribution (AWS CLI)
- Erste Schritte mit einer sicheren statischen Website

Richten Sie Ihre ein AWS-Konto

In diesem Thema werden vorbereitende Schritte beschrieben, z. B. das Erstellen eines AWS-Konto, um Sie auf die Nutzung von Amazon vorzubereiten CloudFront.

Themen

- Melden Sie sich für eine an AWS-Konto
- Wählen Sie, wie Sie darauf zugreifen möchten CloudFront

Melden Sie sich für eine an AWS-Konto

Um loszulegen AWS, benötigen Sie eine AWS-Konto. Es gibt zwei Möglichkeiten, sich anzumelden AWS. Wenn Sie die vollständige Kontrolle über die Kontokonfiguration und den Speicherort Ihrer Inhalte haben möchten oder wenn Sie regulierte Workloads wie HIPAA oder FedRAMP haben, verwenden Sie Sign up for (advanced). AWS Wenn Sie ein neuer Benutzer sind und einen Anmeldenamen verwenden möchten, den Sie bereits besitzen, um sich anzumelden AWS und eine AWS Umgebung mit vorkonfigurierten Standardeinstellungen in select zu erstellen, verwenden Sie Sign up for. AWS-RegionenAWS

Wählen Sie, wie Sie darauf zugreifen möchten CloudFront

Sie können auf folgende Weise auf Amazon CloudFront zugreifen:

- AWS-Managementkonsole— Die Verfahren in diesem Handbuch erläutern, wie Sie mit AWS-Managementkonsole dem Aufgaben ausführen können.
- AWS SDKs — Wenn Sie eine Programmiersprache verwenden, die ein SDK für AWS bereitstellt, können Sie ein SDK für den Zugriff CloudFront verwenden. SDKs vereinfachen die Authentifizierung, lassen sich problemlos in Ihre Entwicklungsumgebung integrieren und bieten Zugriff CloudFront auf Befehle. Weitere Informationen finden Sie unter Verwendung CloudFront mit einem AWS SDK.
- CloudFront API — Wenn Sie eine Programmiersprache verwenden, für die kein SDK verfügbar ist, finden Sie in der Amazon CloudFront API-Referenz Informationen zu API-Aktionen und zum Stellen von API-Anfragen.
- AWS CLI— Das AWS Command Line Interface (AWS CLI) ist ein einheitliches Tool für die Verwaltung AWS-Services. Weitere Informationen zur Installation und Konfiguration der AWS CLI finden Sie unter Installieren oder Aktualisieren auf die neueste Version der AWS CLI im Benutzerhandbuch für AWS Command Line Interface .
- Tools für Windows PowerShell — Wenn Sie Erfahrung mit Windows haben PowerShell, ziehen Sie es vielleicht vor, es zu verwenden AWS Tools for Windows PowerShell. Weitere Informationen finden Sie unter Installieren der AWS Tools for Windows PowerShell im AWS -Tools für PowerShell -Benutzerhandbuch.

Beginnen Sie mit einer CloudFront Standarddistribution

Die Verfahren in diesem Abschnitt zeigen Ihnen, wie Sie eine Standardverteilung einrichten, die Folgendes bietet: CloudFront

- Erstellt einen S3-Bucket zur Verwendung als Distributionsursprung.
- Speichert die Originalversionen Ihrer Objekte in einem Amazon Simple Storage Service (Amazon S3)-Bucket.
- Verwendet die Ursprungszugriffssteuerung (OAC), um authentifizierte Anforderungen an Ihren Amazon-S3-Ursprung zu senden. OAC sendet Anfragen durch CloudFront , um zu verhindern, dass Zuschauer direkt auf Ihren S3-Bucket zugreifen. Weitere Informationen zu OAC finden Sie unter Beschränken des Zugriffs auf einen Amazon-S3-Ursprung.
- Verwendet den CloudFront Domainnamen in URLs für Ihre Objekte (z. B.`https://d111111abcdef8.cloudfront.net/index.html`).
- Bewahrt Ihre Objekte für die Standarddauer von 24 Stunden an CloudFront Randpositionen (die Mindestdauer beträgt 0 Sekunden).

Das meiste davon wird automatisch für Sie konfiguriert, wenn Sie eine CloudFront Verteilung erstellen.

Themen

- Voraussetzungen
- Erstellen Sie einen Amazon S3 S3-Bucket
- Hochladen des Inhalts in den Bucket
- Erstellen Sie eine CloudFront Distribution, die einen Amazon S3 S3-Ursprung mit OAC verwendet
- Greifen Sie auf Ihre Inhalte zu über CloudFront
- Bereinigen
- Verbessern Ihrer grundlegenden Distribution

Voraussetzungen

Bevor Sie beginnen, sollten Sie sicherstellen, dass Sie die in beschriebenen Schritte ausgeführt habe Richten Sie Ihre ein AWS-Konto.

Erstellen Sie einen Amazon S3 S3-Bucket

Ein Amazon S3 S3-Bucket ist ein Container für Dateien (Objekte) oder Ordner. CloudFront kann fast jeden Dateityp für Sie verteilen, wenn ein S3-Bucket die Quelle ist. CloudFront kann beispielsweise Text, Bilder und Videos verteilen. Die Menge der Daten, die Sie in Amazon S3 speichern können, ist nicht begrenzt.

In diesem Tutorial erstellen Sie einen S3-Bucket mit den bereitgestellten `hello world`-Beispieldateien, die Sie zur Erstellung einer einfachen Webseite verwenden.

So erstellen Sie einen Bucket

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wir empfehlen, dass Sie für diese erste Schritte unser Hello-World-Beispiel verwenden. Laden Sie die Webseite Hello World herunter: hello-world-html.zip. Entpacken Sie sie und speichern Sie den `css`-Ordner und die `index`-Datei an einem geeigneten Ort, z. B. auf dem Desktop, auf dem Sie Ihren Browser ausführen.
3. Wählen Sie Create Bucket (Bucket erstellen) aus.
4. Geben Sie einen eindeutigen Bucket-Namen ein, der den Benennungsregeln für allgemeine Buckets im Benutzerhandbuch für Amazon Simple Storage Service entspricht.
5. Wir empfehlen, als Region eine AWS-Region zu wählen, die geografisch in Ihrer Nähe liegt. (Dies reduziert die Latenz und die Kosten.)

   - Sie können aber auch eine andere Region auswählen. Eine solche Auswahl könnte zum Beispiel erforderlich sein, um gesetzliche Anforderungen zu berücksichtigen.
6. Belassen Sie alle anderen Einstellungen auf ihren Standardeinstellungen und wählen Sie dann Bucket erstellen.

Hochladen des Inhalts in den Bucket

Nachdem Sie Ihren Amazon-S3-Bucket erstellt haben, laden Sie den Inhalt der entpackten `hello world`-Datei in den Bucket hoch. (Sie haben diese Datei in Erstellen Sie einen Amazon S3 S3-Bucket heruntergeladen und entpackt.)

So laden Sie den Inhalt in Amazon S3 hoch

1. Wählen Sie im Abschnitt Allzweck-Buckets den Namen Ihres neuen Buckets aus.

2. Klicken Sie auf Upload.
3. Ziehen Sie auf der Seite Upload den `css`-Ordner und die `index`-Datei in den Ablagebereich.
4. Belassen Sie alle anderen Einstellungen auf ihren Standardeinstellungen und wählen Sie dann Hochladen.

Erstellen Sie eine CloudFront Distribution, die einen Amazon S3 S3-Ursprung mit OAC verwendet

In diesem Tutorial erstellen Sie eine CloudFront Distribution, die einen Amazon S3 S3-Ursprung mit Origin Access Control (OAC) verwendet. OAC hilft Ihnen dabei, authentifizierte Anforderungen sicher an Ihren Amazon-S3-Ursprung zu senden. Weitere Informationen zu OAC finden Sie unter Beschränken des Zugriffs auf einen Amazon-S3-Ursprung.

Um eine CloudFront Distribution mit einem Amazon S3 S3-Ursprung zu erstellen, der OAC verwendet

1. Öffnen Sie die CloudFront Konsole unter. https://console.aws.amazon.com/cloudfront/v4/home
2. Wählen Sie Create distribution (Distribution erstellen) aus.
3. Geben Sie einen Distributionsnamen für die Standarddistribution ein. Der Name wird als Wert für den `Name`-Schlüssel als Tag angezeigt. Sie können diesen Wert später ändern. Sie können bis zu 50 Tags für Ihre Standarddistribution hinzufügen. Weitere Informationen finden Sie unter Kennzeichnen Sie eine Ressource CloudFront.
4. Wählen Sie Einzelne Website oder App und dann Weiter aus.
5. Wählen Sie Weiter aus.
6. Wählen Sie als Ursprungstyp-Seite Amazon S3 aus.
7. Wählen Sie unter S3-Ursprung die Option S3 durchsuchen und dann den S3-Bucket aus, den Sie für dieses Tutorial erstellt haben.
8. Wählen Sie unter Einstellungen die Option Empfohlene Origin-Einstellungen verwenden aus. CloudFront verwendet die empfohlenen Standardeinstellungen für Cache und Origin für Ihren Amazon S3 S3-Ursprung, einschließlich der Einrichtung von Origin Access Control (OAC). Weitere Informationen zu den empfohlenen Einstellungen finden Sie unter Referenz für vorkonfigurierte Distributionseinstellungen.
9. Wählen Sie Weiter aus.
10. Wählen Sie auf der Seite Sicherheitsvorkehrungen aktivieren aus, ob AWS WAF Sicherheitsvorkehrungen aktiviert werden sollen.

11. Wählen Sie Weiter aus.
12. Wählen Sie Verteilung erstellen aus. CloudFront aktualisiert die S3-Bucket-Richtlinie für Sie.
13. Sehen Sie sich den Abschnitt Details für die neue Distribution an. Wenn die Bereitstellung Ihrer Distribution abgeschlossen ist, ändert sich das Feld Letzte Änderung von Wird bereitgestellt in ein Datum und eine Uhrzeit.
14. Notieren Sie sich den Domainnamen, der Ihrer Distribution CloudFront zugewiesen ist. Er sieht in etwa wie folgt aus: `d111111abcdef8.cloudfront.net`.

Bevor Sie die Distribution und den S3-Bucket aus diesem Tutorial in einer Produktionsumgebung verwenden, stellen Sie sicher, dass sie Ihren spezifischen Anforderungen entsprechen. Informationen zum Konfigurieren des Zugriffs in einer Produktionsumgebung finden Sie unter Konfigurieren des sicheren Zugriffs und Beschränken des Zugriffs auf Inhalte.

Greifen Sie auf Ihre Inhalte zu über CloudFront

Um über auf Ihre Inhalte zuzugreifen CloudFront, kombinieren Sie den Domainnamen für Ihre CloudFront Distribution mit der Hauptseite für Ihre Inhalte. (Sie haben Ihren Distributionsdomainnamen in Erstellen Sie eine CloudFront Distribution, die einen Amazon S3 S3-Ursprung mit OAC verwendet aufgezeichnet.)

- Ihr Vertriebsdomänenname könnte so aussehen: `d111111abcdef8.cloudfront.net`.
- Der Pfad zur Hauptseite einer Website ist in der Regel `/index.html`.

Daher CloudFront könnte die URL, über die Sie auf Ihre Inhalte zugreifen, wie folgt aussehen:

`https://d111111abcdef8.cloudfront.net/index.html`.

Wenn Sie die vorherigen Schritte befolgt und die Hello World-Webseite verwendet haben, sollte eine Webseite mit der Aufschrift Hello World! angezeigt werden.

Wenn Sie weitere Inhalte in diesen S3-Bucket hochladen, können Sie auf die Inhalte zugreifen, CloudFront indem Sie den Namen der CloudFront Distributionsdomain mit dem Pfad zum Objekt im S3-Bucket kombinieren. Wenn Sie eine neue Datei mit dem Namen `new-page.html` zum Stammverzeichnis Ihres S3-Buckets hochladen, sieht die URL beispielsweise wie folgt aus:

`https://d111111abcdef8.cloudfront.net/new-page.html`.

Bereinigen

Wenn Sie die Distribution und den S3-Bucket nur zur Übung erstellt haben, löschen Sie sie, damit keine weiteren Gebühren für sie anfallen. Löschen Sie zunächst die Distribution. Weitere Informationen finden Sie unter den folgenden Links:

- Löschen einer -Verteilung
- Einen Bucket löschen

Verbessern Ihrer grundlegenden Distribution

Dieses Erste-Schritte-Tutorial bietet einen minimalen Rahmen für die Erstellung einer Distribution. Wir empfehlen Ihnen, sich mit den folgenden Verbesserungen vertraut zu machen:

- Sie können die Funktion für CloudFront private Inhalte verwenden, um den Zugriff auf die Inhalte in den Amazon S3 S3-Buckets einzuschränken. Weitere Informationen zum Verteilen von privaten Inhalten finden Sie unter Bereitstellen privater Inhalte mit signierten URLs und signierten Cookies.
- Sie können Ihre CloudFront Distribution so konfigurieren, dass sie einen benutzerdefinierten Domainnamen verwendet (z. B. `www.example.com` anstelle von `d111111abcdef8.cloudfront.net`). Weitere Informationen finden Sie unter Verwenden benutzerdefinierter URLs.
- In diesem Tutorial wird ein Amazon-S3-Ursprung mit Ursprungszugriffssteuerung (OAC) verwendet. Sie können OAC jedoch nicht verwenden, wenn Ihr Ursprung ein S3-Bucket ist, der als Website-Endpunkt konfiguriert ist. Wenn das der Fall ist, müssen Sie Ihren Bucket CloudFront als benutzerdefinierten Ursprung einrichten. Weitere Informationen finden Sie unter Verwenden eines Amazon-S3-Buckets, der als Website-Endpunkt konfiguriert ist. Weitere Informationen zu OAC finden Sie unter Beschränken des Zugriffs auf einen Amazon-S3-Ursprung.

Erste Schritte mit einer Standarddistribution (AWS CLI)

Die Verfahren in diesem Abschnitt zeigen Ihnen, wie Sie mithilfe von AWS CLI with CloudFront eine grundlegende Konfiguration einrichten, die Folgendes umfasst:

- Erstellen eines Amazon-S3-Buckets zur Verwendung als Distributionsursprung
- Speichern der Originalversionen Ihrer Objekte in einem S3-Bucket

- Verwenden der Ursprungszugriffssteuerung (OAC), um authentifizierte Anforderungen an Ihren Amazon-S3-Ursprung zu senden. OAC sendet Anfragen durch CloudFront , um zu verhindern, dass Zuschauer direkt auf Ihren S3-Bucket zugreifen. Weitere Informationen zu OAC finden Sie unter Beschränken des Zugriffs auf einen Amazon-S3-Ursprung.
- Verwenden Sie den CloudFront Domainnamen in URLs für Ihre Objekte (z. B.`https://d111111abcdef8.cloudfront.net/index.html`).
- Halten Sie Ihre Objekte für die Standarddauer von 24 Stunden an CloudFront Randpositionen (die Mindestdauer beträgt 0 Sekunden).

Die meisten dieser Optionen sind anpassbar. Weitere Informationen dazu, wie Sie Ihre CloudFront -Verteilungsoptionen anpassen, finden Sie unter Eine Distribution erstellen.

Voraussetzungen

Bevor Sie beginnen, sollten Sie sicherstellen, dass Sie die in beschriebenen Schritte ausgeführt habe Richten Sie Ihre ein AWS-Konto.

Installieren Sie das AWS CLI und konfigurieren Sie es mit Ihren Anmeldeinformationen. Weitere Informationen finden Sie unter Erste Schritte mit dem AWS CLI im AWS CLI -Benutzerhandbuch.

Erstellen Sie einen Amazon S3 S3-Bucket

Ein Amazon S3 S3-Bucket ist ein Container für Dateien (Objekte) oder Ordner. CloudFront kann fast jeden Dateityp für Sie verteilen, wenn ein S3-Bucket die Quelle ist. CloudFront kann beispielsweise Text, Bilder und Videos verteilen. Die Menge der Daten, die Sie in Amazon S3 speichern können, ist nicht begrenzt.

In diesem Tutorial erstellen Sie einen S3-Bucket und laden eine HTML-Datei hoch, die Sie zur Erstellung einer einfachen Webseite verwenden.

```
aws s3 mb s3://amzn-s3-demo-bucket/ --region us-east-1
```

*amzn-s3-demo-bucket*Ersetzen Sie es durch einen weltweit eindeutigen Bucket-Namen. Für den empfehlen wir AWS-Region, eine Region zu wählen, die sich geografisch in Ihrer Nähe befindet. Dies reduziert die Latenz und die Kosten, aber die Auswahl einer anderen Region funktioniert auch. Eine solche Auswahl ist eventuell erforderlich, um gesetzliche Anforderungen zu berücksichtigen.

Hochladen des Inhalts in den Bucket

Laden Sie für dieses Tutorial die Beispielinhaltsdateien für eine einfache „Hello World“-Webseite herunter und extrahieren Sie sie.

```
# Create a temporary directory
mkdir -p ~/cloudfront-demo

# Download the sample Hello World files
curl -o ~/cloudfront-demo/hello-world-html.zip https://docs.aws.amazon.com/
AmazonCloudFront/latest/DeveloperGuide/samples/hello-world-html.zip

# Extract the zip file
unzip ~/cloudfront-demo/hello-world-html.zip -d ~/cloudfront-demo/hello-world
```

Dadurch wird ein Verzeichnis mit einer `index.html`-Datei und einem `css`-Ordner erstellt. Laden Sie diese Dateien in Ihren S3-Bucket hoch.

```
aws s3 cp ~/cloudfront-demo/hello-world/ s3://amzn-s3-demo-bucket/ --recursive
```

Erstellen einer Ursprungszugriffssteuerung (OAC)

In diesem Tutorial erstellen Sie eine Ursprungszugriffssteuerung (OAC, Origin Access Control). OAC hilft Ihnen dabei, authentifizierte Anforderungen sicher an Ihren Amazon-S3-Ursprung zu senden. Weitere Informationen zu OAC finden Sie unter Beschränken des Zugriffs auf einen Amazon-S3-Ursprung.

```
aws cloudfront create-origin-access-control \
    --origin-access-control-config Name="oac-for-
s3",SigningProtocol=sigv4,SigningBehavior=always,OriginAccessControlOriginType=s3
```

Speichern Sie die OAC-ID aus der Ausgabe als Umgebungsvariable. Ersetzen Sie den Beispielwert durch Ihre eigene OAC-ID. Sie verwenden diese im nächsten Schritt.

```
OAC_ID="E1ABCD2EFGHIJ"
```

Erstellen einer Standarddistribution

Erstellen Sie eine Konfigurationsdatei namens `distribution-config.json` für die Distribution. Ersetzen Sie den Beispielnamen des Buckets durch Ihren Bucket-Namen für die Werte `Id`, `DomainName` und `TargetOriginId`.

```
cat > distribution-config.json << EOF
{
    "CallerReference": "cli-example-$(date +%s)",
    "Origins": {
        "Quantity": 1,
        "Items": [
            {
                "Id": "S3-amzn-s3-demo-bucket",
                "DomainName": "amzn-s3-demo-bucket.s3.amazonaws.com",
                "S3OriginConfig": {
                    "OriginAccessIdentity": ""
                },
                "OriginAccessControlId": "$OAC_ID"
            }
        ]
    },
    "DefaultCacheBehavior": {
        "TargetOriginId": "S3-amzn-s3-demo-bucket",
        "ViewerProtocolPolicy": "redirect-to-https",
        "AllowedMethods": {
            "Quantity": 2,
            "Items": ["GET", "HEAD"],
            "CachedMethods": {
                "Quantity": 2,
                "Items": ["GET", "HEAD"]
            }
        },
        "DefaultTTL": 86400,
        "MinTTL": 0,
        "MaxTTL": 31536000,
        "Compress": true,
        "ForwardedValues": {
            "QueryString": false,
            "Cookies": {
                "Forward": "none"
            }
        }
```

```
    },
    "Comment": "CloudFront distribution for S3 bucket",
    "Enabled": true
}
EOF
```

Erstellen Sie die Standarddistribution.

```
aws cloudfront create-distribution --distribution-config file://distribution-
config.json
```

Speichern Sie die Distributions-ID und den Domainnamen aus der Ausgabe als Umgebungsvariablen. Ersetzen Sie die Beispielwerte durch Ihre eigenen Werte. Sie werden sie später in diesem Tutorial verwenden.

```
DISTRIBUTION_ID="EABCD1234XMPL"
DOMAIN_NAME="d111111abcdef8.cloudfront.net"
```

Bevor Sie die Distribution und den S3-Bucket aus diesem Tutorial in einer Produktionsumgebung verwenden, stellen Sie sicher, dass sie Ihren spezifischen Anforderungen entsprechen. Informationen zum Konfigurieren des Zugriffs in einer Produktionsumgebung finden Sie unter Konfigurieren des sicheren Zugriffs und Beschränken des Zugriffs auf Inhalte.

Aktualisieren Ihrer S3-Bucket-Richtlinie

Aktualisieren Sie Ihre S3-Bucket-Richtlinie, um den Zugriff auf die Objekte CloudFront zu ermöglichen. Ersetzen Sie Beispielnamen des Buckets durch den Namen Ihres Buckets.

```
# Get your AWS account ID
ACCOUNT_ID=$(aws sts get-caller-identity --query 'Account' --output text)

# Create the bucket policy
cat > bucket-policy.json << EOF
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowCloudFrontServicePrincipal",
            "Effect": "Allow",
            "Principal": {
```

```
                "Service": "cloudfront.amazonaws.com"
            },
            "Action": "s3:GetObject",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
            "Condition": {
                "StringEquals": {
                    "AWS:SourceArn": "arn:aws:cloudfront::$ACCOUNT_ID:distribution/
$DISTRIBUTION_ID"
                }
            }
        }
    ]
}
EOF

# Apply the bucket policy
aws s3api put-bucket-policy \
    --bucket amzn-s3-demo-bucket \
    --policy file://bucket-policy.json
```

Bestätigen der Bereitstellung der Distribution

Nachdem Sie Ihre Distribution erstellt haben, dauert es einige Zeit, bis sie bereitgestellt wird. Fahren Sie mit dem nächsten Schritt fort, wenn sich der Distributionsstatus von `InProgress` in `Deployed` ändert.

```
aws cloudfront get-distribution --id $DISTRIBUTION_ID --query 'Distribution.Status'
```

Alternativ können Sie auch den `wait`-Befehl verwenden, um auf die Bereitstellung der Distribution zu warten.

```
aws cloudfront wait distribution-deployed --id $DISTRIBUTION_ID
```

Greifen Sie auf Ihre Inhalte zu über CloudFront

Um über auf Ihre Inhalte zuzugreifen CloudFront, kombinieren Sie den Domainnamen für Ihre CloudFront Distribution mit der Hauptseite für Ihre Inhalte. Ersetzen Sie den CloudFront Beispiel-Domainnamen durch Ihren eigenen.

```
https://d111111abcdef8.cloudfront.net/index.html
```

Wenn Sie die vorherigen Schritte befolgt und die HTML-Datei erstellt haben, sollte eine Website mit der Aufschrift Hello world! angezeigt werden.

Wenn Sie weitere Inhalte in diesen S3-Bucket hochladen, können Sie auf die Inhalte zugreifen, CloudFront indem Sie den Namen der CloudFront Distributionsdomain mit dem Pfad zum Objekt im S3-Bucket kombinieren. Wenn Sie eine neue Datei mit dem Namen `new-page.html` zum Stammverzeichnis Ihres S3-Buckets hochladen, sieht die URL beispielsweise wie folgt aus:

`https://d111111abcdef8.cloudfront.net/new-page.html`.

Bereinigen

Wenn Sie die Distribution und den S3-Bucket nur zur Übung erstellt haben, löschen Sie sie, damit keine weiteren Gebühren für sie anfallen. Deaktivieren und löschen Sie zuerst die Distribution.

So deaktivieren und löschen Sie eine Standarddistribution (AWS CLI)

1. Deaktivieren Sie zunächst die Distribution.

```
# Get the current configuration and ETag
ETAG=$(aws cloudfront get-distribution-config --id $DISTRIBUTION_ID --query 'ETag'
 --output text)

# Create a modified configuration with Enabled=false
aws cloudfront get-distribution-config --id $DISTRIBUTION_ID | \
jq '.DistributionConfig.Enabled = false' > temp_disabled_config.json

# Update the distribution to disable it
aws cloudfront update-distribution \
    --id $DISTRIBUTION_ID \
    --distribution-config file://<(jq '.DistributionConfig'
 temp_disabled_config.json) \
    --if-match $ETAG
```

2. Warten Sie, bis die Distribution deaktiviert ist.

```
aws cloudfront wait distribution-deployed --id $DISTRIBUTION_ID
```

3. Löschen Sie die Distribution.

```
# Get the current ETag
```

```
ETAG=$(aws cloudfront get-distribution-config --id $DISTRIBUTION_ID --query 'ETag'
 --output text)

# Delete the distribution
aws cloudfront delete-distribution --id $DISTRIBUTION_ID --if-match $ETAG
```

So löschen Sie einen S3-Bucket (AWS CLI)

- Löschen Sie den S3-Bucket und seinen Inhalt. Ersetzen Sie Beispielnamen des Buckets durch Ihren eigenen.

```
# Delete the bucket contents
aws s3 rm s3://amzn-s3-demo-bucket --recursive

# Delete the bucket
aws s3 rb s3://amzn-s3-demo-bucket
```

Führen Sie die folgenden Befehle aus, um die für dieses Tutorial erstellten lokalen Dateien zu bereinigen:

```
# Clean up local files
rm -f distribution-config.json bucket-policy.json temp_disabled_config.json
rm -rf ~/cloudfront-demo
```

Optional können Sie auch die OAC löschen, die Sie für dieses Tutorial erstellt haben.

```
# Get the OAC ETag
OAC_ETAG=$(aws cloudfront get-origin-access-control --id $OAC_ID --query 'ETag' --
output text)

# Delete the OAC
aws cloudfront delete-origin-access-control --id $OAC_ID --if-match $OAC_ETAG
```

Erste Schritte mit einer sicheren statischen Website

Sie können mit Amazon beginnen, CloudFront indem Sie die in diesem Thema beschriebene Lösung verwenden, um eine sichere statische Website für Ihren Domainnamen zu erstellen. Eine statische

Website verwendet nur statische Dateien wie HTML, CSS JavaScript, Bilder und Videos und benötigt keine Server oder serverseitige Verarbeitung. Mit dieser Lösung erhält Ihre Website folgende Vorteile:

- Verwendet den dauerhaften Speicher von Amazon Simple Storage Service (Amazon S3) – Mit dieser Lösung wird ein Amazon S3-Bucket erstellt, um den Inhalt Ihrer statischen Website zu hosten. Um Ihre Website zu aktualisieren, laden Sie einfach Ihre neuen Dateien in den S3-Bucket hoch.
- Wird durch das Amazon CloudFront Content Delivery Network beschleunigt — Diese Lösung erstellt eine CloudFront Distribution, um Ihre Website Zuschauern mit geringer Latenz zur Verfügung zu stellen. Die Verteilung ist mit Origin Access Control (OAC) konfiguriert, um sicherzustellen, dass auf die Website nur über CloudFront S3 und nicht direkt von S3 aus zugegriffen werden kann.
- Ist durch HTTPS und Sicherheitsheader gesichert — Diese Lösung erstellt ein SSL/TLS Zertifikat in AWS Certificate Manager (ACM) und hängt es an die Distribution an. CloudFront Dieses Zertifikat ermöglicht es der Verteilung, die Website Ihrer Domäne sicher mit HTTPS zu bedienen.
- Wird konfiguriert und bereitgestellt mit AWS CloudFormation— Diese Lösung verwendet eine CloudFormation Vorlage, um alle Komponenten einzurichten, sodass Sie sich mehr auf den Inhalt Ihrer Website und weniger auf die Konfiguration der Komponenten konzentrieren können.

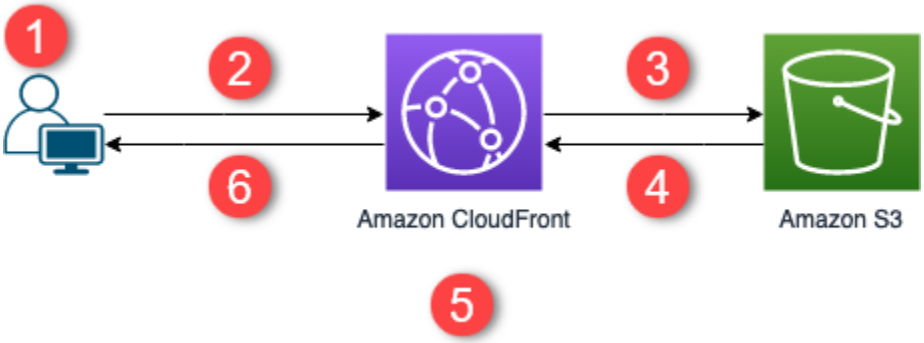
Diese Lösung ist Open Source auf GitHub. Um den Code anzuzeigen, eine Pull-Anforderung einzureichen oder ein Problemticket zu öffnen, gehen Sie zu https://github.com/aws-samples/amazon-cloudfront-secure-static-site.

Themen

- Übersicht über die Lösung
- Bereitstellen der Lösung

Übersicht über die Lösung

Das folgende Diagramm zeigt einen Überblick über die Funktionsweise dieser statischen Website-Lösung:



1. Der Betrachter fordert die Website unter www.example.com an.
2. Wenn das angeforderte Objekt zwischengespeichert ist, wird das Objekt aus seinem Cache an den Viewer CloudFront zurückgegeben.
3. Wenn sich das Objekt nicht im CloudFront Cache befindet, CloudFront wird das Objekt vom Ursprung (einem S3-Bucket) angefordert.
4. S3 gibt das Objekt an zurück CloudFront.
5. CloudFront speichert das Objekt im Cache.
6. Die Objekte werden an den Viewer zurückgegeben. Nachfolgende Anfragen für das Objekt, die an derselben CloudFront Edge-Position ankommen, werden vom CloudFront Cache aus bedient.

Bereitstellen der Lösung

Um diese sichere statische Website-Lösung bereitzustellen, können Sie eine der folgenden Optionen auswählen:

- Verwenden Sie die CloudFormation Konsole, um die Lösung mit Standardinhalten bereitzustellen, und laden Sie dann den Inhalt Ihrer Website auf Amazon S3 hoch.
- Klonen Sie die Lösung auf Ihren Computer, um Ihre Website-Inhalte hinzuzufügen. Stellen Sie dann die Lösung mit AWS Command Line Interface (AWS CLI) bereit.

> **Note**
>
> Sie müssen die Region USA Ost (Nord-Virginia) verwenden, um die CloudFormation Vorlage bereitzustellen.

Themen

- Voraussetzungen
- Verwenden Sie die CloudFormation Konsole
- Lokales Klonen der Lösung
- Suchen von Zugriffsprotokollen

Voraussetzungen

Um diese Lösung verwenden zu können, müssen Sie die folgenden Voraussetzungen haben:

- Ein registrierter Domänenname, z. B. example.com, der auf eine von Amazon Route 53 gehostete Zone verweist. Die gehostete Zone muss sich in demselben AWS-Konto befinden, in dem Sie diese Lösung bereitstellen. Wenn Sie keinen registrierten Domänennamen haben, können Sie einen bei Route 53 registrieren. Wenn Sie einen registrierten Domänennamen haben, der aber nicht auf eine von Route 53 gehostete Zone verweist, konfigurieren Sie Route 53 als Ihren DNS-Service.
- AWS Identity and Access Management (IAM) -Berechtigungen zum Starten von CloudFormation Vorlagen, die IAM-Rollen erstellen, und Berechtigungen zum Erstellen aller AWS Ressourcen in der Lösung. Weitere Informationen finden Sie unter Steuern des Zugriffs mit AWS Identity and Access Management im Benutzerhandbuch für AWS CloudFormation .

Sie sind für die Kosten verantwortlich, die bei der Nutzung dieser Lösung entstehen. Weitere Informationen zu den Kosten finden Sie auf den jeweiligen Preisseiten. AWS-Service

Verwenden Sie die CloudFormation Konsole

Zur Bereitstellung über die CloudFormation Konsole

1. Starten Sie diese Lösung in der CloudFormation Konsole. Melden Sie sich bei Bedarf bei Ihrem an AWS-Konto.
2. Der Assistent zum Erstellen von Stacks wird in der CloudFormation Konsole geöffnet. Er enthält vorausgefüllte Felder, die die CloudFormation Vorlage dieser Lösung angeben.

   Wählen Sie unten auf der Seite Next aus.
3. Geben Sie auf der Seite Specify stack details (Stackdetails angeben) Werte für die folgenden Felder ein:

- SubDomain— Geben Sie die Subdomain ein, die Sie für Ihre Website verwenden möchten. Wenn die Subdomain beispielsweise www ist, ist Ihre Website verfügbar unter *www.example.com.* (Ersetzen Sie example.com durch Ihren Domainnamen, wie im folgenden Punkt bullet).
- DomainName— Geben Sie Ihren Domainnamen ein, z. B. *example.com* Diese Domäne muss auf eine von Route 53 gehostete Zone verweisen.
- HostedZoneId— Die Route 53-Hosting-Zonen-ID Ihres Domainnamens.
- CreateApex— (Optional) Erstellen Sie in Ihrer CloudFront Konfiguration einen Alias für die Domain Apex (example.com).

4. Wenn Sie fertig sind, wählen Sie Next (Weiter).
5. (Optional) Auf der Seite Configure stack options (Stack-Optionen konfigurieren) können Sie Tags und andere Stack-Optionen hinzufügen.
6. Wenn Sie fertig sind, wählen Sie Next (Weiter).
7. Scrollen Sie auf der Seite Review (Überprüfen) zum Ende der Seite und wählen Sie dann die beiden Felder im Abschnitt Capabilities (Funktionen) aus. Diese Funktionen ermöglichen es CloudFormation , eine IAM-Rolle zu erstellen, die den Zugriff auf die Ressourcen des Stacks ermöglicht, und die Ressourcen dynamisch zu benennen.
8. Wählen Sie Create stack (Stack erstellen) aus.
9. Warten Sie, bis der Stack erstellt wurde. Der Stack erstellt einige verschachtelte Stacks. Dieser Vorgang kann einige Minuten dauern. Wenn der Vorgang fertig ist, wechselt der Status zu CREATE_COMPLETE.

   Wenn der Status CREATE_COMPLETE lautet, wechseln Sie zu https://, *www.example.com* um Ihre Website aufzurufen (ersetzen Sie www.example.com durch die Subdomain und den Domainnamen, die Sie in Schritt 3 angegeben haben). Sie sollten den Standardinhalt der Website sehen:



So ersetzen Sie den Standardinhalt der Website durch Ihren eigenen

1. Öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie den Bucket aus, dessen Name mit amazon-cloudfront-secure-static-site-s3bucketroot - beginnt.

   > **Note**
   >
   > Stellen Sie sicher, dass Sie den Bucket mit s3bucketroot im Namen auswählen, nicht s3bucketlogs. Der Bucket mit s3bucketroot im Namen enthält den Inhalt der Website. Der Bucket mit s3bucketlogs im Namen enthält nur Protokolldateien.

3. Löschen Sie den Standardinhalt der Website, und laden Sie dann Ihren eigenen hoch.

   > **Note**
   >
   > Wenn Sie Ihre Website mit dem Standardinhalt dieser Lösung angesehen haben, ist es wahrscheinlich, dass ein Teil des Standardinhalts an einem Edge-Speicherort zwischengespeichert wurde. CloudFront Um sicherzustellen, dass Zuschauer Ihre aktualisierten Website-Inhalte sehen, machen Sie die Dateien ungültig, um die zwischengespeicherten Kopien von CloudFront den Edge-Speicherorten zu entfernen. Weitere Informationen finden Sie unter Aufheben der Gültigkeit von Dateien zum Entfernen von Inhalten.

Lokales Klonen der Lösung

Voraussetzungen

Um Ihren Website-Inhalt hinzuzufügen, bevor Sie diese Lösung bereitstellen, müssen Sie die Artefakte der Lösung lokal verpacken, wofür Node.js und npm erforderlich sind. Weitere Informationen finden Sie unter https://www.npmjs.com/get-npm.

So fügen Sie Ihre Website-Inhalte hinzu und stellen die Lösung bereit

1. Klonen oder laden Sie die Lösung von herunte https://github.com/aws-samples/amazon-cloudfront-secure-static-site. Öffnen Sie nach dem Klonen oder Herunterladen eine Eingabeaufforderung oder ein Terminal, und navigieren Sie zum Ordner `amazon-cloudfront-secure-static-site`.

2. Führen Sie den folgenden Befehl aus, um die Artefakte der Lösung zu installieren und zu verpacken:

```
make package-static
```

3. Kopieren Sie den Inhalt Ihrer Website in den Ordner `www` und überschreiben Sie den Standard-Website-Inhalt.
4. Führen Sie den folgenden AWS CLI Befehl aus, um einen Amazon S3 S3-Bucket zum Speichern der Lösungsartefakte zu erstellen. *amzn-s3-demo-bucket-for-artifacts*Ersetzen Sie es durch Ihren eigenen Bucket-Namen.

```
aws s3 mb s3://amzn-s3-demo-bucket-for-artifacts --region us-east-1
```

5. Führen Sie den folgenden AWS CLI Befehl aus, um die Artefakte der Lösung als CloudFormation Vorlage zu verpacken. *amzn-s3-demo-bucket-for-artifacts*Ersetzen Sie es durch den Namen des Buckets, den Sie im vorherigen Schritt erstellt haben.

```
aws cloudformation package \
    --region us-east-1 \
    --template-file templates/main.yaml \
    --s3-bucket amzn-s3-demo-bucket-for-artifacts \
    --output-template-file packaged.template
```

6. Führen Sie den folgenden Befehl aus, um die Lösung bereitzustellen CloudFormation, und ersetzen Sie dabei die folgenden Werte:

   - *your-CloudFormation-stack-name*— Durch einen Namen für den CloudFormation Stack ersetzen.
   - *example.com*— Ersetze es durch deinen Domainnamen. Diese Domain muss auf eine von Route 53 gehostete Zone im selben AWS-Konto verweisen.
   - *www*— Ersetzen Sie durch die Subdomain, die Sie für Ihre Website verwenden möchten. Wenn die Subdomäne beispielsweise www ist, ist Ihre Website unter www.example.com verfügbar.
   - *hosted-zone-ID*— Ersetzen Sie es durch die Route 53-Hosting-Zonen-ID Ihres Domainnamens.

```
aws cloudformation deploy \
    --region us-east-1 \
    --stack-name your-CloudFormation-stack-name \
```

```
    --template-file packaged.template \
    --capabilities CAPABILITY_NAMED_IAM CAPABILITY_AUTO_EXPAND \
    --parameter-overrides DomainName=example.com SubDomain=www HostedZoneId=hosted-
zone-ID
```

- (Optional) Um den Stack mit einem Domain-Apex bereitzustellen, führen Sie stattdessen den folgenden Befehl aus.

```
aws --region us-east-1 cloudformation deploy \
    --stack-name your-CloudFormation-stack-name \
    --template-file packaged.template \
    --capabilities CAPABILITY_NAMED_IAM CAPABILITY_AUTO_EXPAND \
    --parameter-overrides  DomainName=example.com SubDomain=www
 HostedZoneId=hosted-zone-ID CreateApex=yes
```

7. Warten Sie, bis die Erstellung des CloudFormation Stacks abgeschlossen ist. Der Stack erstellt einige verschachtelte Stacks. Dieser Vorgang kann einige Minuten dauern. Wenn der Vorgang fertig ist, wechselt der Status zu CREATE_COMPLETE.

   Wenn sich der Status in CREATE_COMPLETE ändert, gehen Sie zu, https://www.example.com um Ihre Website aufzurufen (ersetzen Sie www.example.com durch die Subdomain und den Domainnamen, die Sie im vorherigen Schritt angegeben haben). Sie sollten den Inhalt Ihrer Website sehen.

Suchen von Zugriffsprotokollen

Diese Lösung ermöglicht Zugriffsprotokolle für die CloudFront-Verteilung. Führen Sie die folgenden Schritte aus, um die Zugriffsprotokolle der Verteilung zu finden.

So finden Sie die Zugriffsprotokolle der Verteilung

1. Öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie den Bucket aus, dessen Name mit amazon-cloudfront-secure-static-site-s3bucketlogs - beginnt.

> **Note**
>
> Stellen Sie sicher, dass Sie den Bucket mit s3bucketlogs im Namen auswählen, nicht s3bucketroot. Der Bucket mit s3bucketlogs im Namen enthält Protokolldateien. Der Bucket mit s3bucketroot im Namen enthält den Inhalt der Website.

3. Der Ordner mit dem Namen cdn enthält die Zugriffsprotokolle. CloudFront

CloudFront für E-Commerce- und Marktplatz-Anwendungen

Eine E-Commerce- oder Marketplace-Anwendung bietet drei verschiedene Inhaltstypen über eine einzige CloudFront Distribution: statische Inhalte (Produktbilder, CSS, JavaScript), dynamische Seiten (Produktangebote, Suchergebnisse) und API-Antworten (Warenkorb, Inventar, Preise). Jeder Inhaltstyp erfordert ein anderes Cacheverhalten, eine andere TTL- und Quellkonfiguration, um ein ausgewogenes Verhältnis zwischen Aktualität und Leistung herzustellen.

Die Sicherheitskontrollen, die für Marketplace- und E-Commerce-Distributionen gelten, finden Sie unter. Sicherheitscheckliste für Marketplace-Distributionen

Übersicht über die Architektur

Eine CloudFront Marketplace-Distribution verwendet mehrere Cache-Verhaltensweisen, um Anfragen anhand von URL-Pfadmustern an den richtigen Ursprung weiterzuleiten. Statische Ressourcen werden vom Amazon Simple Storage Service (Amazon S3) bereitgestellt, während dynamische Inhalte und API-Anfragen an ein Elastic Load Balancing (Application Load Balancer) weitergeleitet werden, das sich vor Ihren Anwendungsservern befindet. Ein ElastiCache (Valkey-) Cluster bietet einen Cache auf Anwendungsebene zwischen dem ALB und Ihrer Datenbank für Produktdaten, auf die häufig zugegriffen wird.

Komponenten der Distributionsarchitektur

Komponente	Bereitgestellter Inhalt	Zweck
CloudFront Verteilung	Alle Inhalte (einzelne Domain)	Globales Edge-Caching, TLS-Termi nierung, Anforderungsrouting unter Verwendung des Cache-Verhaltens
Amazon S3-Ursprung (statische Ressourcen)	Produktbilder, CSS, JS, Schriftarten	Dauerhafter Objektspeicher mit Origin Access Control (OAC). Unveränderlicher Inhalt mit langen TTLs.
ALB-Ursprung (dynamisc her Inhalt)	Produktseiten, Suche, API-Endpunkte	Routen zu Anwendungsservern. Kurze TTLs oder kein Caching für personalisierte Inhalte.

Komponente	Bereitgestellter Inhalt	Zweck
ElastiCache Cluster (Valkey)	Produktkatalog, Inventarzahlen, Sitzungsdaten	Application-level Cache zwischen ALB und Datenbank. Sub-millisecond liest Daten, auf die häufig zugegriffen wird.

Verhalten für E-Commerce zwischenspeichern

Das Cache-Verhalten bestimmt, wie Anfragen auf der Grundlage von URL-Pfadmustern CloudFront behandelt werden. Konfigurieren Sie für einen Marktplatz separate Verhaltensweisen für jeden Inhaltstyp mit den entsprechenden Cache-Richtlinien und TTLs.

Empfohlenes Cache-Verhalten

Pfadmuster	Origin	TTL	Cache-Richtlinie	Begründung
`/static/*`	Amazon S3	365 Tage	CachingOptimized	Bei versionierten Dateinamen (z. B. `app.a1b2c3.js`) ist ein maximaler TTL-Wert zulässig. Eine Cache-Invalidierung ist nie erforderlich — stellen Sie neue Versionen mit neuen Dateinamen bereit.
`/images/*`	Amazon S3	30 Tage	CachingOptimized	Produktbilder ändern sich selten. Verwenden Sie die Cache-Tag-Invalidierung, wenn ein Verkäufer ein Bild aktualisiert. Niedrigerer TTL-Wert als bei statischen Assets, da Bild-URLs möglicherweise wiederverwendet werden.

Pfadmuster	Origin	TTL	Cache-Richtlinie	Begründung
`/api/products/*`	ALB	60 Sekunden	Benutzerdefiniert (einschließlich Abfragezeichenfolgen)	Produktlisten- und Such-APIs ändern sich häufig, tolerieren aber kurzzeitige Veraltbarkeit. Fügen Sie Abfragezeichenfolgenparameter für Paginierung und Filter in den Cache-Schlüssel ein.
`/api/cart/*`	ALB	0 (kein Cache)	CachingDisabled	Warenkorboperationen sind benutzerspezifisch und müssen immer den Ursprung erreichen. Deaktivieren Sie das Caching vollständig.
`/api/inventory/*`	ALB	5 Sekunden	Benutzerdefiniert (einschließlich Abfragezeichenfolgen)	Die Anzahl der Lagerbestände ändert sich bei Verkaufsveranstaltungen schnell. Ein sehr kurzes TTL verhindert Überverkäufe und reduziert gleichzeitig die Ursprungslast.

Pfadmuster	Origin	TTL	Cache-Richtlinie	Begründung
`Default (*)`	ALB	0 (kein Cache)	CachingDisabled	Das Standardverhalten leitet nicht zwischengespeicherte Anfragen zum serverseitigen Rendern an die Anwendung weiter. Seiten, die personalisierte Inhalte (Empfehlungen, Benutzername) enthalten, sollten nicht am Edge zwischengespeichert werden.

> **Note**
>
> Ordne das Verhalten im Cache von den spezifischsten bis hin zu den unspezifischsten an. CloudFront wertet Pfadmuster in der angegebenen Reihenfolge aus und verwendet die erste Übereinstimmung. Platziere `/api/cart/*` davor`/api/*`, um sicherzustellen, dass Warenkorbanfragen das Caching umgehen.

Weitere Informationen zu Cache-Verhalten finden Sie unter the section called “Einstellungen für das Cache-Verhalten”.

TTL-Strategie für Marketplace-Inhalte

Die TTL-Konfiguration bestimmt, wie lange CloudFront zwischengespeicherte Inhalte bereitgestellt werden, bevor der Ursprung auf Aktualisierungen überprüft wird. E-commerce Anwendungen erfordern unterschiedliche TTL-Strategien, je nachdem, wie sich die Aktualität der Inhalte auf das Kundenerlebnis auswirkt.

Unveränderliche Ressourcen (CSS, JS, Schriftarten)

Stellen Sie die maximale TTL (365 Tage) ein. Verwenden Sie Dateinamen mit Inhalts-Hash, sodass bei jeder Bereitstellung neue URLs erstellt werden. Niemals ungültig machen — alte

Versionen laufen auf natürliche Weise ab, wenn Benutzer die neuen Dateinamen laden. `Cache-Control: public, max-age=31536000, immutable`Am Ursprung festgelegt.

Bilder des Produkts

Stellen Sie TTL auf 30 Tage ein, wobei das Cache-Tag beim Update ungültig wird. Wenn ein Verkäufer ein neues Bild hochlädt, wird es anhand des Cache-Tags des Produkts ungültig gemacht und nicht anhand des Pfads. Dies zielt nur auf die betroffenen Objekte ab, ohne eine pauschale Invalidierung durch Platzhalter. `Cache-Control: public, max-age=2592000`Am Ursprung festgelegt.

Produktkatalog-API

Stellen Sie TTL auf 60 Sekunden ein. Produktangebote, Beschreibungen und Preise tolerieren kurzzeitige Veraltbarkeit. Dadurch wird die Quelllast bei Verkehrsspitzen (Blitzverkäufe, Produkteinführungen) reduziert und gleichzeitig die Daten einigermaßen aktuell gehalten. Wird so eingestelltCache-Control: public, max-age=60, stale-while-revalidate=30, dass veraltete Inhalte bereitgestellt werden, während im CloudFront Hintergrund eine neue Antwort abgerufen wird.

API für Inventar und Preisgestaltung

Stellen Sie TTL auf 5 Sekunden ein. Diese Werte ändern sich häufig und wirken sich auf Kaufentscheidungen aus. Durch sehr kurzes Caching wird die Ausgangslast bei Datenverkehrsspitzen immer noch um ein Vielfaches reduziert. Tausende von gleichzeitigen Benutzern erhalten dieselbe zwischengespeicherte Antwort von 5 Sekunden, anstatt dass jeder den Ursprung erreicht.

User-specific Inhalt (Warenkorb, Empfehlungen)

Deaktivieren Sie das Caching (TTL = 0). Inhalte, die je nach Benutzer unterschiedlich sind, müssen immer den Ursprung erreichen. Wenn Sie versehentlich benutzerspezifische Inhalte zwischenspeichern, werden CloudFront möglicherweise die Daten eines Benutzers an einen anderen Benutzer weitergegeben. Verwenden Sie die `CachingDisabled` verwaltete Richtlinie für diese Verhaltensweisen.

Origin-Konfiguration

Amazon S3-Ursprung für statische Assets

Verwenden Sie einen Amazon S3-Bucket als Ursprung für Produktbilder und statische Assets. Konfigurieren Sie Origin Access Control (OAC), um den direkten Zugriff auf den Bucket einzuschränken — alle Anfragen müssen CloudFront durchgehen.

Amazon S3-Ursprungseinstellungen

Einstellung	Wert	Begründung
Ursprungsdomäne	`marketplace-assets.s3.amazonaws.com`	Verwenden Sie den Amazon S3-REST-API-Endpunkt (nicht den Website-Endpunkt), um OAC zu unterstützen.
Ursprungszugriff	Origin Access Control (OAC)	Beschränkt den Bucket-Zugriff nur auf CloudFront . Verhindert den direkten S3-URL-Zugriff auf Ihre Assets.
Origin Shield	Aktiviert (Region, die dem Bucket am nächsten liegt)	Fügt eine zentrale Caching-Ebene zwischen den Kantenpositionen und dem Ursprung hinzu. Reduziert die Kosten für S3-Anfragen und verbessert die Cache-Trefferquote für Bilder.
Verbindungsversuche	3	Default (Standard). Wiederholungen bei vorübergehenden S3-Fehlern.

ALB-Ursprung für dynamische Inhalte

Verwenden Sie einen Application Load Balancer als Ursprung für dynamische Produktseiten, Such- und API-Endpunkte. Das ALB leitet mithilfe von Zielgruppen an Ihre Anwendungsserver weiter (Amazon EC2-Instances, Amazon ECS-Aufgaben oder Lambda-Funktionen).

ALB-Ursprungseinstellungen

Einstellung	Wert	Begründung
Ursprungsdomäne	ALB-DNS-Name	Verwenden Sie den ALB-DNS-N amen direkt. Verwenden Sie keine IP-Adresse — ALB-IPs ändern sich.
Protocol (Protokoll)	Nur HTTPS	Verschlüsselt den Verkehr zwischen CloudFront und dem Ursprung. Erforderlich für sensible Daten (Benutzersitzungen, Zahlungsi nformationen).
Benutzerdefinierter Origin-Header	`X-Origin-Verify: <secret-value>`	Schränkt den ALB-Zugriff auf Anfragen von ein. CloudFront Das ALB sucht nach diesem Header und lehnt direkte Zugriffsversuche ab.
Verbindungstimeout	10 Sekunden	Kürzer als Standard (30 Sekunden). Versagen Sie schnell bei Problemen mit dem Ursprung, anstatt Edge-Verb indungen warten zu lassen.
Antwort-Timeout	30 Sekunden	Ermöglicht Zeit für komplexe Suchabfragen und Katalogop erationen. Erhöhen Sie den Wert, wenn Ihre API lang andauernde Operationen hat.
Keep-alive Zeitlimit	5 Sekunden	Verwendet Verbindungen zum ALB erneut. Reduziert den TLS-Hands hake-Overhead für nachfolgende Anfragen.

Application-level Caching mit ElastiCache (Valkey)

CloudFront Zwischenspeicherung von Inhalten am Edge, aber Daten, auf die häufig zugegriffen wird und für die Datenbankabfragen erforderlich sind, profitieren von einem zusätzlichen Cache auf Anwendungsebene. Ein ElastiCache Cluster, auf dem Valkey ausgeführt wird, befindet sich zwischen Ihren Anwendungsservern und der Datenbank und ermöglicht Lesevorgänge im Sub-Millisekundenbereich für Produktkatalogdaten, Inventarzählungen und Sitzungsstatus.

Two-tier Caching-Strategie

Ebene	Was es zwischens peichert	TTL	Fehlverhalten beim Zwischens peichern
CloudFront Rand	Vollständige HTTP-Antworten (API JSON, HTML-Seiten, Bilder)	5s — 365d (nach Verhalten)	Leitet die Anfrage an den ALB-Ursprung weiter
ElastiCache (Tal)	Anwendungsdatenobj ekte (Produktdatensätze , Inventar, Sitzungen)	30 Sekunden — 5 Minuten (je nach Datentyp)	Die Anwendung fragt die Datenbank ab und schreibt das Ergebnis in den Cache

Bei einer Anfrage auf einer Produktseite:

1. CloudFront überprüft seinen Edge-Cache. Gibt bei einem Treffer sofort die zwischengespeicherte Antwort zurück.
2. Bei Fehlschlag CloudFront leitet er zur ALB weiter.
3. Die Anwendung ElastiCache sucht nach den Produktdaten. Bei einem Treffer wird die Antwort aus zwischengespeicherten Daten (unter einer Millisekunde) erstellt.
4. Bei ElastiCache Fehlschlag wird die Datenbank abgefragt, in die Datenbank geschrieben und die ElastiCache Antwort zurückgegeben.
5. CloudFront speichert die Antwort gemäß der TTL-Einstellung des Verhaltens am Edge im Cache.

Wenn ein Verkäufer ein Produkt aktualisiert:

1. Die Anwendung macht den Produktschlüssel ungültig. ElastiCache
2. Die Anwendung sendet eine Cache-Tag-Invalidierung CloudFront für das Tag des Produkts an.

3. Die nächste Anfrage löst eine neue Antwort durch beide Cache-Ebenen aus.

Häufig gestellte Fragen

In den folgenden Abschnitten werden häufig gestellte Fragen zu Caching-Strategien für E-Commerce- und Marketplace-Workloads beantwortet.

Wie gehe ich beim Caching mit personalisierten Inhalten um?

Trennen Sie personalisierte Elemente von zwischenspeicherbaren Inhalten. Stellen Sie die Seiten-Shell (Produktdetails, Bilder, Beschreibungen) aus dem CloudFront Cache bereit und laden Sie personalisierte Elemente (Empfehlungen, Anzahl der Einkaufswagen, Benutzername), indem Sie clientseitige API-Aufrufe verwenden, die das Caching umgehen. Mit diesem Ansatz können Sie das teure Rendern der Seite zwischenspeichern und die Personalisierung auf dem neuesten Stand halten.

Wie wähle ich zwischen Invalidierung und kurzen TTLs?

Verwenden Sie kurze TTLs (5—60 Sekunden) für Inhalte, die sich häufig und vorhersehbar ändern (Inventar, Preise). Verwende die Ungültigkeitserklärung für Inhalte, die sich selten ändern, aber sofort aktualisiert werden müssen, wenn sie sich ändern (Produktbilder nach einer Bearbeitung durch den Verkäufer, Produktbeschreibungen nach einer Konformitätsprüfung). Für eine Ungültigerklärung fallen Kosten pro Anfrage an und es gibt ein Limit für die Parallelität — verwenden Sie sie nicht als Ersatz für angemessene TTLs.

Wie bereite ich mich auf schnelle Verkäufe und Traffic-Spitzen vor?

CloudFront skaliert automatisch, um Verkehrsspitzen zu bewältigen. Um die Cache-Trefferquote während eines Verkaufs zu maximieren: Warmen Sie die Produktseiten vor, indem Sie sie vor Beginn des Events aufrufen, erhöhen Sie vorübergehend die API-TTLs (erhöhen Sie beispielsweise die TTL für Inventar von 5 Sekunden auf 15 Sekunden), um mehr Traffic am Edge abzufangen, und stellen Sie sicher, dass Ihr ElastiCache Cluster über genügend Speicherplatz für erhöhte Cache-Schreibvorgänge verfügt. Überwachen Sie die Metrik zur CloudFront Cache-Trefferquote während des Events.

Sollte ich Origin Shield für meinen Marktplatz verwenden?

Ja. Origin Shield fügt eine zentrale Cache-Ebene zwischen regionalen Edge-Caches und deinem Origin hinzu. Bei E-Commerce-Workloads reduziert es die Ursprungsanfragen für beliebte Produkte

(weil ein Cache-Treffer bei Origin Shield alle Edge-Standorte bedient), reduziert die S3-GET-Kosten für Bilder und glättet Traffic-Spitzen zum ALB. Aktiviere es in der Region, die deinem Ursprung am nächsten liegt. Weitere Informationen finden Sie unter the section called “Verwenden von Origin Shield”.

Sicherheitscheckliste für Marketplace-Distributionen

E-commerce und Marketplace-Anwendungen verarbeiten sensible Daten — Kundensitzungen, Zahlungsströme und Premium-Inhalte. Diese Checkliste behandelt die Sicherheitskontrollen, die für eine CloudFront Distribution mit einer ElastiCache (Valkey-) Caching-Ebene spezifisch sind. Wenden Sie diese Kontrollen zusätzlich zu den unter beschriebenen allgemeinen bewährten CloudFront Sicherheitsmethoden an. Konfigurieren des sicheren Zugriffs und Beschränken des Zugriffs auf Inhalte

Diese Checkliste ergänzt die Konfigurationshinweise unter. CloudFront für E-Commerce- und Marktplatz-Anwendungen

Signierte URLs und Cookies für Premium-Inhalte

Verwenden Sie signierte URLs oder signierte Cookies, um den Zugriff auf Inhalte einzuschränken, für die eine Autorisierung erforderlich ist — Premium-Produktvorschauen, gekaufte digitale Waren, zeitlich begrenzte Werbeinhalte oder verkäuferspezifische Inhalte.

Vergleich signierter URL und signierter Cookies

Methode	Anwendungsfall	E-commerce Beispiel
Signierte URL	Beschränkung des Zugriffs auf einzelne Dateien. Download-Links teilen. Clients, die keine Cookies unterstützen.	One-time Download-Link für ein gekauftes digitales Produkt. Time-limitedZugriff auf ein hochauflösendes Produktbild für einen verifizierten Verkäufer.
Signierter Cookie	Bereitstellung des Zugriffs auf mehrere Dateien (z. B. alle Inhalte einer Premium-Stufe). Vermeidung von URL-Änder ungen für bestehende Inhalte.	Premium-Mitgliedschaft: Zugriff auf einen gesamten Katalog digitaler Inhalte. Subscriber-only Produktvorschauen auf mehreren Seiten.

Anforderungen an die Konfiguration:

- Erstellen Sie eine Schlüsselgruppe mit einem öffentlichen RSA-Schlüssel (mindestens 2048-Bit). Speichern Sie den privaten Schlüssel, damit Ihre Anwendung AWS Secrets Manager Signaturen generieren kann.
- Stellen Sie für das Cache-Verhalten die Option Zuschauerzugriff einschränken auf Ja ein und ordnen Sie die Schlüsselgruppe zu.
- Legen Sie eine dem Inhaltstyp entsprechende Ablaufzeit fest: Minuten für Download-Links, Stunden für Browsersitzungen, Tage für Abonnementzugriff.
- Verwenden Sie eine benutzerdefinierte Richtlinie (keine vorgefertigte Richtlinie), wenn Sie den IP-Bereich einschränken oder Platzhalterpfade zulassen müssen.

Weitere Informationen zur Implementierung finden Sie in the section called “Beschränken von Inhalten mit signierten URLs und signierten Cookies”.

Origin Access Control (OAC) für Amazon S3

Die Origin-Zugriffskontrolle verhindert, dass Benutzer die Umgehung umgehen und CloudFront direkt auf Ihren Amazon S3-Bucket zugreifen. Ohne OAC kann jeder, der Ihre Bucket-URL entdeckt, auf Ressourcen zugreifen, ohne Ihre signierten URL/cookie Durchsetzungsmaßnahmen, WAF-Regeln oder geografischen Einschränkungen zu umgehen. Vollständige Konfigurationsschritte finden Sie unter. the section called “Beschränken Sie den Zugriff auf ein AWS Ursprungsserver”

Aktivieren Sie OAC für alle Amazon S3-Ursprünge

Erstellen Sie ein OAC und verknüpfen Sie es mit jedem Amazon S3-Ursprung in Ihrer Distribution. Aktualisieren Sie die Bucket-Richtlinie, sodass nur der CloudFront Service Principal (`cloudfront.amazonaws.com`) mit einer Bedingung zulässig ist, die Ihrem Verteilungs-ARN entspricht.

Öffentlichen Zugriff entfernen

Aktivieren Sie in der Amazon Simple Storage Service-Konsole Amazon S3 Block Public Access für den Bucket. Dies ist eine Amazon S3-Einstellung, keine CloudFront Einstellung. Die Bucket-Richtlinie gewährt nur Zugriff auf CloudFront — es sind keine öffentlichen ACLs oder Richtlinien erforderlich. Weitere Informationen finden Sie unter Blockieren des öffentlichen Zugriffs auf Ihren Amazon S3-Speicher im Amazon Simple Storage Service-Benutzerhandbuch.

Verwenden Sie den Amazon S3-REST-API-Endpunkt

Konfigurieren Sie den Ursprung mithilfe des REST-API-Endpunkts (`bucket-name.s3.amazonaws.com`) des Buckets, nicht mit dem Endpunkt der Amazon S3-Website. OAC funktioniert nur mit dem REST-API-Endpunkt.

Verschlüsselung während der Übertragung

Verschlüsseln Sie alle Verbindungen im Anforderungspfad — vom Viewer zum CloudFront, CloudFront zum Ursprung und von der Anwendung zum ElastiCache Cache.

Konfiguration der Verschlüsselung bei der Übertragung

Connection (Verbindung)	Einstellung	Konfiguration
Zuschauer → CloudFront	Viewer-Protokollrichtlinie: Nur HTTPS	Stellen Sie die Viewer-Protokollrichtlinie `https-only` auf `redirect-to-https` oder für alle Cache-Verhaltensweisen ein. Verwenden Sie eine TLS 1.2-Mindestsicherheitsrichtlinie (`TLSv1.2_2021`).
CloudFront → S3-Ursprung	Ursprungsprotokoll: HTTPS	S3-REST-API-Endpunkte unterstützen standardmäßig HTTPS. CloudFront verwendet HTTPS, wenn Sie den Ursprung mit dem REST-API-Endpunkt und OAC konfigurieren.
CloudFront → ALB-Ursprung	Ursprungsprotokoll: nur HTTPS	Stellen Sie die Ursprungsprotokollrichtlinie auf ein`https-only` . Installieren Sie ein gültiges TLS-Zertifikat auf dem ALB (verwenden AWS Certificate Manager). CloudFront validiert das Zertifikat bei der Verbindung.

Connection (Verbindung)	Einstellung	Konfiguration
Bewerbung → ElastiCache	In-transit Verschlüsselung: Aktiviert	Aktivieren Sie in der ElastiCache Konsole `TransitEncryptionEnabled` die Replikationsgruppe. Dies ist eine ElastiCache Einstellung. Verwenden Sie TLS-Verbindungen aus Ihrem Anwendungscode. Valkey und Redis OSS 6.0 und höher unterstützen nativ die Verschlüsselung bei der Übertragung.

VPC-Sicherheitsgruppen für den Cache-Cluster

Der ElastiCache Cluster wird in einer VPC ausgeführt und darf nur Verbindungen von Ihren Anwendungsservern akzeptieren. Verwenden Sie die Verkettung von Sicherheitsgruppen, um den Zugriff einzuschränken.

Sicherheitsgruppenkonfiguration für ElastiCache

Sicherheitsgruppe	Regel für eingehenden Datenverkehr	Begründung
Cache-Cluster SG	TCP 6379 nur von der Anwendung SG	Nur Anwendungsserver können den Cache erreichen. Kein Internetzugang, kein Zugriff von anderen Diensten. Verwenden Sie Port 6379 (Valkey/Redis Standardport) oder Ihren konfigurierten Port.
Anwendung SG	TCP 443 nur von ALB SG	Anwendungsserver akzeptieren nur Datenverkehr von der ALB. In Kombination mit dem Cache-Cluster SG entsteht eine Kette: CloudFront → ALB → App → Cache.

Zusätzliche VPC-Härtung:

- Stellen Sie den ElastiCache Cluster in privaten Subnetzen ohne Internet-Gateway-Route bereit.
- Verwenden Sie eine Subnetzgruppe, die sich über mindestens 2 Availability Zones erstreckt, um eine hohe Verfügbarkeit zu gewährleisten.
- Aktivieren Sie das ElastiCache Authentifizierungstoken (Passwort) für eine zusätzliche Authentifizierungsebene, die über Sicherheitsgruppen hinausgeht.
- Deaktivieren Sie den öffentlichen Zugriff auf den ElastiCache Cluster (Standardeinstellung, aber überprüfen Sie dies).

Zusätzliche Sicherheitskontrollen

Beschränken Sie den ALB-Zugriff nur auf CloudFront

Fügen Sie einen benutzerdefinierten Origin-Header (`X-Origin-Verify`) hinzu, der bei jeder Anfrage CloudFront gesendet wird. Konfigurieren Sie den ALB oder die Anwendung so, dass Anfragen abgelehnt werden, die diesen Header nicht enthalten. Speichern Sie den Header-Wert darin AWS Secrets Manager und rotieren Sie ihn regelmäßig. Verwenden Sie alternativ die Liste der CloudFront verwalteten Präfixe, um die Regeln für eingehende ALB-Sicherheitsgruppen auf CloudFront IP-Bereiche zu beschränken.

Aktivieren Sie diese Option AWS WAF für die Distribution

Ordnen Sie einer AWS WAF Web-ACL Regeln zur Ratenbegrenzung zu, um sich vor Missbrauch im Einkaufswagen, dem Auslesen von Anmeldedaten auf den Anmelde-APIs und dem Aussortieren von Inventar zu schützen. Verwenden Sie verwaltete Regelgruppen für gängige Bedrohungen (SQL Injection, XSS) und benutzerdefinierte Regeln für anwendungsspezifische Muster.

Geo-restrictions zur Einhaltung der Vorschriften

Wenn Ihr Marketplace in bestimmten Ländern tätig ist, aktivieren Sie die geografische Beschränkung, um Anfragen aus Regionen zu blockieren, in denen Sie rechtlich nicht zum Verkauf oder Versand befugt sind. Bewerben Sie sich auf Vertriebsebene. Für geografische Einschränkungen pro Pfad verwenden Sie AWS WAF Geo-Match-Bedingungen in einer Web-ACL-Regel mit einer URL-Pfadbedingung.

Protokollierung des Zugriffs

Aktivieren Sie die CloudFront Standardprotokollierung oder die Echtzeitprotokollierung, um ungewöhnliche Zugriffsmuster zu erkennen — plötzliche Spitzen bei der Nutzung von Download-Links, Zugriffe aus unerwarteten Regionen oder wiederholte fehlgeschlagene Authentifizierungsversuche an API-Endpunkten.

CloudFront Pauschalpreispläne

CloudFront Pauschaltarife kombinieren das Amazon CloudFront Global Content Delivery Network (CDN) mit mehreren AWS-Services Funktionen zu einem monatlichen Preis, ohne dass zusätzliche Gebühren anfallen, unabhängig von Verkehrsspitzen oder Angriffen.

Flat-rate Die Preispläne beinhalten die folgenden Funktionen zu einem einfachen monatlichen Preis:

- CloudFront CDN
- AWS WAF und DDoS-Schutz
- Bot-Management und Analytik
- Amazon Route 53 DNS
- Amazon CloudWatch protokolliert die Aufnahme
- TLS-Zertifikat
- Serverloses Edge-Computing
- Amazon S3-Speicherguthaben jeden Monat

Die Tarife sind in den Stufen Free, Pro, Business und Premium erhältlich, um den Anforderungen Ihrer Anwendung gerecht zu werden. Für Tarife ist kein jährliches Abonnement erforderlich, um die besten verfügbaren Tarife zu erhalten. Beginnen Sie mit dem kostenlosen Tarif und führen Sie ein Upgrade durch, um von mehr Funktionen und größeren Nutzungskonditionen zu profitieren.

Themen

- Vorteile von CloudFront Flatrate-Tarifen
- Funktionen nach Preisstufen
- Monatliche Nutzungsgebühren
- Kosten, die durch Ihren Tarif abgedeckt werden
- Insgesamt reduzieren AWS Kosten mit Preisplänen
- Verwalten Sie Ihre Pauschalpreispläne
- Berechtigungen
- Flat-rate Kontingente für Preispläne
- Nicht unterstützte Funktionen

Vorteile von CloudFront Flatrate-Tarifen

Der CloudFront Preisplan bietet mehrere wichtige Vorteile:

- Konsolidierte Dienstleistungen und Preisgestaltung

  Kombinieren Sie mehrere AWS-Services Funktionen in einem einzigen Plan zu einem Pauschalpreis. Entwickelt, um separate Servicekäufe und Preisberechnungen im Voraus zu vermeiden.

- Keine Überschreitungen

  Unabhängig von Verkehrsspitzen oder Angriffen fallen keine Gebühren für Überschreitungen an.

- Klare Nutzungsberechtigungen

  Jeder Plan beinhaltet veröffentlichte Nutzungsberechtigungen, die für eine optimale Leistung auf dieser Ebene konzipiert sind. Überwachen Sie Ihre Nutzung, erhalten Sie proaktive Benachrichtigungen und führen Sie je nach den Anforderungen Ihrer Anwendung ein Upgrade durch, ohne langfristige Verpflichtungen.

- Schützen Sie sich vor DDoS-Angriffen

  CloudFront und AWS WAF absorbieren und blockieren Angriffe, bevor sie Ihre Infrastruktur erreichen. Reserviert Ihre Rechen-, Datenbank- und Infrastrukturnutzung nur für legitimen Datenverkehr. Blockierte DDoS-Angriffe und Anfragen, die von blockiert wurden, werden AWS WAF niemals auf Ihr Nutzungslimit angerechnet.

- Reduzieren Sie Ihre Gesamtkosten AWS

  Die Datenübertragung von AWS Anwendungen, die auf Diensten wie Amazon S3, AWS Application Load Balancer (ALB) oder Amazon API Gateway ausgeführt werden, ist CloudFront weiterhin kostenlos. Wenn Sie Ihre AWS Anwendungen nicht direkt im Internet, CloudFront sondern über das Internet bereitstellen, deckt Ihr Pauschaltarif die Datenübertragungskosten zwischen Ihren Anwendungen und Ihren Zuschauern zu einem einfachen monatlichen Preis ab, ohne dass Sie sich Gedanken über Überschreitungen machen müssen. Weniger Anfragen, die Ihren Ursprung erreichen, reduzieren auch Ihre Kosten für Dienste, die nutzungsabhängig abgerechnet werden.

Funktionen nach Preisstufen

Jeder Preisplan umfasst eine CloudFront Distribution mit bis zu einer Apex-Domain (Root), die wichtige Funktionen und Dienste zu einem monatlichen Preis vereint. Jeder Plan beinhaltet auch zusätzliche S3-Speicherguthaben.

Tarife für höhere Tarife beinhalten alle Funktionen der Tarife niedrigerer Tarife sowie zusätzliche Funktionen.

- Kostenlos — Für Hobbyisten, Lernende und Entwickler, die gerade erst loslegen.
- Pro — Starten und erweitern Sie kleine Websites, Blogs und Anwendungen.
- Unternehmen — Schützen und beschleunigen Sie Geschäftsanwendungen.
- Premium — Skalieren und schützen Sie geschäfts- und unternehmenskritische Anwendungen.

Wählen Sie eine Tarifstufe aus, die Funktionen und Konfigurationen umfasst, die Sie für Ihre Anwendungen benötigen. Sehen Sie sich die folgenden Funktionen pro Preisplan an.

Funktionen des Preisplans

Die folgende Tabelle zeigt die CloudFront Funktionen von DDoS, Amazon Route 53 CloudWatch, Amazon und Amazon S3, die in jeder Preisstufe enthalten sind. AWS WAF

Leistung und Lieferung	Kostenfrei	Pro	Geschäft	Prämie
Globales CDN Nutzen Sie CloudFront über 750 globale Edge-Standorte als riesige, verteilte zentrale Anlaufstelle für Ihre Webanwendung.	Ja ✓	Ja ✓	Ja ✓	Ja ✓

Leistung und Lieferung	Kostenfrei	Pro	Geschäft	Prämie
Beschleunigen Sie statische, dynamische und nicht zwischens peicherbare Anwendungen.				

Leistung und Lieferung	Kostenfrei	Pro	Geschäft	Prämie
Zwischens peichern von Inhalten	Ja 	Ja 	Ja 	Ja 
Speichern Sie Kopien Ihrer Inhalte an CloudFront über 750 Edge-Standorten weltweit und stellen Sie sie Benutzern vom nächstgelegenen Standort aus zur Verfügung . Reduziert die Ladezeiten, schützt Ihre Anwendung vor Datenverk ehrsspitzen und spart Kosten, da wiederholte Anfragen lokal statt von Ihren Anwendung sservern aus bearbeitet werden.				

Leistung und Lieferung	Kostenfrei	Pro	Geschäft	Prämie
Schnelle Cache-Invalidierungen Entfernen oder aktualisieren Sie zwischeng espeicherte Inhalte an allen Edge-Standorten innerhalb von Sekunden.	Ja 	Ja 	Ja 	Ja 
Ungültige rklärungen durch Cache-Tag Weisen Sie Ihren Inhalten benutzerd efinierte Tags zu und entfernen Sie alle zwischeng espeicherten Inhalte, die ein bestimmtes Tag gemeinsam nutzen, ohne einzelne Dateipfade zu verfolgen.		Ja 	Ja 	Ja 

Leistung und Lieferung	Kostenfrei	Pro	Geschäft	Prämie
Intelligentes Routing Leitet Benutzer mithilfe von Netzwerkdaten in Echtzeit intelligent an den optimalen Edge-Standort weiter und stellt für eine bessere Leistung über das AWS private Netzwerk eine Verbindung zu Ihrem AWS Ausgangspunkt her.	Ja 	Ja 	Ja 	Ja 

Leistung und Lieferung	Kostenfrei	Pro	Geschäft	Prämie
Mehrstufiges Caching [Regionale Edge-Caches befinden] sich zwischen Edge-Stan dorten und Ihrer Anwendung, um Inhalte länger zu speichern. Dadurch wird Ihre Anwendung entlastet und eine schnelle Bereitstellung gewährleistet.	Ja ✓	Ja ✓	Ja ✓	Ja ✓
Standard-Cache-Regeln Trifft effektive Caching-E ntscheidungen, um die meisten Webanwend ungen ohne benutzerd efinierte Konfiguration zwischenz uspeichern.	Ja ✓	Ja ✓	Ja ✓	Ja ✓

Leistung und Lieferung	Kostenfrei	Pro	Geschäft	Prämie
Benutzerdefinierte Caching-Regeln Steuern Sie, wie Inhalte CloudFront zwischengespeichert werden, indem Sie angeben, welche Anforderungswerte verwendet werden sollen, und optimieren Sie mithilfe von Cache-Richtlinien die Anforderungen Ihrer Anwendung an Leistung, Personalisierung und Aktualität.			Ja 	Ja 

Leistung und Lieferung	Kostenfrei	Pro	Geschäft	Prämie
High-speed Routing am Ursprung Mit Origin Shield werden dynamische Anfragen über ein CloudFront privates Netzwerk von Edge-Standorten zu Ihrem Ursprung weitergeleitet, um einen leistungsstarken Pfad zu Ihrem Ursprung zu gewährleisten.				Ja 

Leistung und Lieferung	Kostenfrei	Pro	Geschäft	Prämie
Reduzierung der Ursprungslast Fügt mithilfe von Origin Shield eine zusätzliche Caching-Ebene in der Nähe Ihrer Webanwendung hinzu. Origin Shield konsolidiert Anfragen von allen Edge-Standorten und reduziert so die Belastung Ihrer Anwendung, insbesondere bei Datenverkehrsspitzen.				Ja 

Leistung und Lieferung	Kostenfrei	Pro	Geschäft	Prämie
Automatisches Origin-Failover Leitet den Datenverkehr automatisch an einen Backup-Ursprung weiter, falls Ihr primärer Ursprung ausfällt, und sorgt so für eine hohe Verfügbarkeit, ohne die Benutzer zu stören.				Ja 

Leistung und Lieferung	Kostenfrei	Pro	Geschäft	Prämie
Standardregeln für Anfragen von der Herkunft Steuern Sie, welche Informationen aus Zuschauer anfragen automatisch in Anfragen an Ihren Absender aufgenommen werden. Verwenden Sie dazu AWS verwaltete Richtlinien für Absenderanfragen, die für gängige Szenarien optimiert sind.	Ja 	Ja 	Ja 	Ja 

Leistung und Lieferung	Kostenfrei	Pro	Geschäft	Prämie
Standardregeln für Antwort-Header Verwenden Sie Richtlinien für AWS verwaltete Antwortheader, um HTTP-Header in Antworten an Zuschauer hinzuzufügen oder zu entfernen. Diese sind für allgemeine Sicherheitsheader, CORS-Einstellungen und andere Standardanwendungsfälle vorkonfiguriert.	Ja 	Ja 	Ja 	Ja 

Leistung und Lieferung	Kostenfrei	Pro	Geschäft	Prämie
Benutzerdefinierte Regeln für die Herkunft der Anfrage			Ja 	Ja 
Erstellen Sie Ihre eigenen Richtlinien für Quellanfragen, um genau festzulegen, welche URL-Abfragezeichen folgen, Header und Cookies an Ihren Ursprung weitergeleitet werden. So können Sie benutzerdefinierte Analysen und die Bearbeitung von Anfragen ermöglichen.				

Leistung und Lieferung	Kostenfrei	Pro	Geschäft	Prämie
Benutzerdefinierte Regeln für Antwort-Header			Ja	Ja
Erstellen Sie Ihre eigenen Richtlinien für Antwortheader, um genau zu steuern, welche HTTP-Header in Antworten an Zuschauer CloudFront hinzugefügt oder entfernt werden, z. B. Sicherheitsheader, Content Security Policy (CSP), CORS-Einstellungen und benutzerdefinierte Anwendungs-Header.				

Leistung und Lieferung	Kostenfrei	Pro	Geschäft	Prämie
Anzahl der Cache-Verhaltensweisen	5	10	50	100
Konfigurieren Sie das Cache-Verhalten, um zu steuern, wie Anfragen nach bestimmten URL-Mustern CloudFront behandelt werden, einschließlich welcher Herkunft der Inhalt bereitgestellt wird, wie Inhalte zwischengespeichert werden und ob HTTPS oder signierte URLs erforderlich sind.				
Sicherheit und Schutz				

Leistung und Lieferung	Kostenfrei	Pro	Geschäft	Prämie
Always-on DDoS-Schutz Schützen Sie sich vor DDoS-Angriffen, die auf Ihre Websites oder Anwendungen abzielen.	Ja 	Ja 	Ja 	Ja 
Erweiterter DDoS-Schutz Identifizieren und blockieren Sie DDoS-Angriffe in Sekundenschnelle mit dem AntiDDoS-AMR. AWS lernt Ihre individuellen Anwendungsmuster kennen, um zwischen Angriffen und natürlichen Überspannungen legitimer Benutzer zu unterscheiden.			Ja 	Ja 

Leistung und Lieferung	Kostenfrei	Pro	Geschäft	Prämie
Webanwendungsfirewall (WAF) Schützen Sie sich auf der Grundlage interner Bedrohungsinformationen von Amazon vor häufigen Anwendungsschwachstellen und potenziellen Bedrohungen. Anfragen werden blockiert, bevor sie Ihre Server erreichen.	Ja 	Ja 	Ja 	Ja 

Leistung und Lieferung	Kostenfrei	Pro	Geschäft	Prämie
Anzahl der WAF-Regeln	5	25	50	75
Gesamtzahl der Sicherheitsregeln, die Sie in Ihrer WAF-Konfiguration erstellen und aktivieren können, einschließlich benutzerdefinierter Regeln und AWS verwalteter Regeln.				

Leistung und Lieferung	Kostenfrei	Pro	Geschäft	Prämie
Schutzmaßnahmen für PHP WordPress - und SQL-Datenbanken		Ja 	Ja 	Ja 
Use-case basierte Sicherheitsregeln zum Schutz gängiger Anwendungen und Betriebssysteme wie PHP WordPress, SQL-Datenbanken, Linux und Windows.				

Leistung und Lieferung	Kostenfrei	Pro	Geschäft	Prämie
KI-Verkehrsanalyse Überwachen Sie Zugriffsmuster, Anforderungsvolumen und beliebte Pfade von KI-Bots, die mit Ihren Inhalten interagieren. KI-Bots werden anhand der Absicht klassifiziert, Aktivitäten wie potenziell unbefugtes Scraping zu Trainingszwecken zu erkennen.		Ja 	Ja 	Ja 

Leistung und Lieferung	Kostenfrei	Pro	Geschäft	Prämie
IP-based Ratenbegrenzung Blockiert automatisch IP-Adressen, die eine konfigurierbare Anzahl von Anfragen über einen Zeitraum von 5 Minuten überschreiten, und schützt so vor HTTP-Flood-Angriffen und Denial-of-Service (DoS) -Versuchen.	Ja	Ja	Ja	Ja
Blockierung des geografischen Datenverkehrs Blockieren Sie Anfragen aus ausgewählten Ländern oder Regionen.	Ja	Ja	Ja	Ja

Leistung und Lieferung	Kostenfrei	Pro	Geschäft	Prämie
CAPTCHA-Herausforderung Erfordern Sie Anfragen, die bestimmten Sicherheitsregeln entsprechen, um ein CAPTCHA-Rätsel zu lösen, um zu beweisen, dass ein Mensch die Anfrage sendet.		Ja 	Ja 	Ja 
Header-based Filterung von Bedrohungen Erstellen Sie WAF-Sicherheitsregeln, die Bedrohungen anhand von HTTP-Anforderungsheadern filtern.		Ja 	Ja 	Ja 

Leistung und Lieferung	Kostenfrei	Pro	Geschäft	Prämie
Regex-based Filterung von Bedrohungen			Ja 	Ja 
Erstellen Sie WAF-Sicherheitsregeln mithilfe regulärer Ausdrücke, um URI-Pfade und HTTP-Anforderungsattribute abzugleichen.				
JavaScript Herausforderung			Ja 	Ja 
Blockieren Sie automatische Bedrohungen, indem Sie von Browsern verlangen, JavaScript Herausforderungen zu bewältigen, bei denen legitime Benutzer überprüft werden.				

Leistung und Lieferung	Kostenfrei	Pro	Geschäft	Prämie
Bot-Management und -Analytik (+KI-Bots) Erkennen und analysieren Sie den Bot-Traffic mit AWS WAF Bot Control für gängige Bots. Bietet Steuerelemente, um unverifizierte Bots zu blockieren, herauszufordern oder zuzulassen und gleichzeitig verifizierte Bots wie Suchmaschinen zu identifizieren und zu unterscheiden. Überwachen Sie KI-Bots und ergreifen Sie Maßnahmen, die auf der Absicht des Bots basieren.			Ja 	Ja 

Leistung und Lieferung	Kostenfrei	Pro	Geschäft	Prämie
Benutzerdefinierte WAF-Antwort		Ja 	Ja 	Ja 
Legen Sie einen bestimmten HTTP-Statuscode und optional eine benutzerdefinierte HTML-, Klartext- oder JSON-Antwort fest, wenn Anfragen durch eine Regel blockiert werden.				

Leistung und Lieferung	Kostenfrei	Pro	Geschäft	Prämie
Einfügen einer Kopfzeile	Ja	Ja	Ja	Ja
Fügen Sie benutzerdefinierte HTTP-Header zu Anfragen hinzu, die die WAF-Prüfung bestehen, sodass nachgelagerte Anwendungen Anfragen unterschiedlich verarbeiten oder zur Analyse kennzeichnen können.	✓	✓	✓	✓
Leichenin spektion anfordern	16 KB	16 KB	64 KB	64 KB
Maximale Größe des HTTP-Anfragetexts, der auf Bedrohungen und bösartige Muster überprüft werden kann. AWS WAF				

Leistung und Lieferung	Kostenfrei	Pro	Geschäft	Prämie
Private Herkunft innerhalb von VPC Erhöhen Sie die Sicherheit, indem Sie Ihre Anwendung in einem privaten VPC-Subnetz aufbewahren, auf das nur über Ihre CloudFront Distributionen zugegriffen werden kann und das über VPC-Ursprünge vor dem öffentlichen Internet verborgen ist. ???			Ja 	Ja 

Leistung und Lieferung	Kostenfrei	Pro	Geschäft	Prämie
Gegenseitiges TLS (Ursprung) Beschränken Sie den unbefugten Zugriff auf Ihre Anwendung (Herkunft) mithilfe von TLS-based Client-Zertifikaten und stellen Sie sicher, dass nur Ihre autorisierten CloudFront Distributionen Verbindungen zu Ihrer Anwendung herstellen können.			Ja 	Ja 

Leistung und Lieferung	Kostenfrei	Pro	Geschäft	Prämie
Origin Access Control (OAC) Sorgen Sie für einen privaten S3-Bucket und gewähren Sie den Zugriff nur über die von Ihnen angegebene CloudFront Distribution. Stellen Sie dabei sicher, dass Ihre Inhalte durch Ihre WAF-Regeln, Ratenlimits und andere in Ihrer CloudFront Distribution konfigurierte Sicherheitskontrollen geschützt sind.	Ja 	Ja 	Ja 	Ja 

Leistung und Lieferung	Kostenfrei	Pro	Geschäft	Prämie
Kostenloses TLS-Zertifikat Kostenloses TLS-Zertifikat für Ihre Domain mit automatischer Verlängerung AWS Certificate Manager.	Ja 	Ja 	Ja 	Ja 
Signierte URLs Erstellen Sie sichere URLs, die bestimmten Benutzern temporären Zugriff auf private Inhalte bieten. Wird häufig verwendet, um private Dokumente mit autorisierten Benutzern zu teilen oder nach Bestätigung der Zahlung sicheren Zugriff auf geschützte Inhalte zu gewähren.	Ja 	Ja 	Ja 	Ja 

Leistung und Lieferung	Kostenfrei	Pro	Geschäft	Prämie
Gegenseitiges TLS (mTLS) Beschränken Sie den Zugriff auf Ihre Anwendung mithilfe der mTLS-Auth entifizierung und stellen Sie sicher, dass nur vertrauen swürdige Clients mit gültigen Zertifikaten eine Verbindun g herstellen können.				Ja 
Edge Compute				

Leistung und Lieferung	Kostenfrei	Pro	Geschäft	Prämie
Serverloses Edge-Computing Führen Sie Lightweight JavaScript at the Edge aus, um URLs, HTTP-Header und request/response Elemente mithilfe von Funktionen innerhalb von Millisekunden zu ändern. CloudFront Lambda @Edge kann auch mit allen Tarifstufen verwendet werden, aber im Gegensatz zu CloudFront Functions werden Lambda @Edge -Aufrufe auf Pay-as-you-go-Basis abgerechnet.	Ja ✓	Ja ✓	Ja ✓	Ja ✓

Leistung und Lieferung	Kostenfrei	Pro	Geschäft	Prämie
Edge-Schlüsselwertspeicher Speichern Sie Daten am Edge und ermöglichen Sie so KeyValueStore eine schnelle und dynamische Anpassung von Inhalten mit CloudFront Functions.		Ja 	Ja 	Ja 
Netzwerk- und Protokollunterstützung				

Leistung und Lieferung	Kostenfrei	Pro	Geschäft	Prämie
IPv6 Stellen Sie Inhalte sowohl über moderne IPv6- als auch über herkömmliche IPv4-Verbindungen für Zuschauer und CloudFront Absender bereit. Ermöglicht eine durchgängige IPv6-Unterstützung für Ihre Anwendungen.	Ja 	Ja 	Ja 	Ja 

Leistung und Lieferung	Kostenfrei	Pro	Geschäft	Prämie
HTTP/2 Ermöglichen Sie schnellere Seitenladevorgänge durch moderne Protokollfunktionen wie Multiplexing, Header-Komprimierung und Stream-Priorisierung. Wird automatisch verwendet, wenn dies von Browsern und Clients unterstützt wird.	Ja 	Ja 	Ja 	Ja 

Leistung und Lieferung	Kostenfrei	Pro	Geschäft	Prämie
HTTP/3 Stellen Sie Inhalte mithilfe von QUIC für Browser und Clients bereit, die dies unterstützen, und ermöglichen Sie so schnellere Verbindungen und eine verbesserte Leistung. Dies kommt insbesondere mobilen Benutzern zugute und gewährleistet die Aufrechterhaltung der Verbindungen, wenn sich die Netzwerkbedingungen ändern.	Ja 	Ja 	Ja 	Ja 

Leistung und Lieferung	Kostenfrei	Pro	Geschäft	Prämie
TLS 1.3 Stellen Sie schnellere HTTPS-Verbindungen über einen Handshake-Prozess bereit, der einen Roundtrip erfordert, verglichen mit zwei in TLS 1.2. Reduziert die Latenz beim ersten Byte im Vergleich zu früheren TLS-Versionen um bis zu 33%. Ende-zu-Ende für Ihre Anwendungen aktiviert.	Ja 	Ja 	Ja 	Ja 

Leistung und Lieferung	Kostenfrei	Pro	Geschäft	Prämie
WebSockets Ermöglichen Sie die persistente bidirektionale Kommunikation in Echtzeit zwischen Browsern und Servern. Ideal für KI-Chat-Anwendungen, Mehrspieler-Spiele, kollaborative Arbeitsbereiche und Echtzeit-Datenfeeds wie Finanzhandelsplattformen.	Ja 	Ja 	Ja 	Ja 
Protokollieren und überwachen				

Leistung und Lieferung	Kostenfrei	Pro	Geschäft	Prämie
Auf Protokolle zugreifen Greifen Sie auf detaillierte CloudFront Anforderungsprotokolle zu, um die Sicherheits- und Versanddatenverkehrsmuster zu verstehen. Die Erfassung von Amazon CloudWatch Logs ist ohne zusätzliche Kosten enthalten.		Ja 	Ja 	Ja 

Leistung und Lieferung	Kostenfrei	Pro	Geschäft	Prämie
WAF-Anforderungsprotokolle Greifen Sie auf detaillierte AWS WAF Anforderungsprotokolle zu, um die Sicherheits- und Übertragungsmuster zu verstehen. Die Aufnahme von Amazon CloudWatch Logs ist ohne zusätzliche Kosten enthalten.		Ja 	Ja 	Ja 

Leistung und Lieferung	Kostenfrei	Pro	Geschäft	Prämie
Sicherheits-Dashboard Überwachen Sie Sicherheitsereignisse, untersuchen Sie Bedrohungen und ergreifen Sie sofortige Blockierungsmaßnahmen mithilfe visueller Analysen, ohne Sicherheitsregeln zu schreiben. Pro and above beinhaltet einen visuellen Protokollanalysator, mit dem Sie Verkehrsmuster schnell verstehen können, ohne Protokolle abfragen zu müssen.	Ja 	Ja 	Ja 	Ja 
DNS				

Leistung und Lieferung	Kostenfrei	Pro	Geschäft	Prämie
Amazon Route 53 DNS Schneller, zuverlässiger öffentlicher autoritativer DNS-Dienst mithilfe von Route 53.	Ja 	Ja 	Ja 	Ja 
Datensätze pro gehosteter Zone Die maximale Anzahl von DNS-Einträgen, die in der gehosteten Zone zulässig sind.	50	100	1000	5000

Leistung und Lieferung	Kostenfrei	Pro	Geschäft	Prämie
DNSSEC Schützen Sie Ihre Domain vor DNS-Spoofing und Man-in-the-Middle-Angriffen, bei denen Angreifer DNS-Abfragen abfangen und Besucher auf gefälschte Websites umleiten. Sichert den DNS-Verkehr, indem Ihre DNS-Einträge kryptografisch signiert werden.	Ja 	Ja 	Ja 	Ja 
Speicherung				

Leistung und Lieferung	Kostenfrei	Pro	Geschäft	Prämie
Amazon S3-Speicher	5 GB	50 GB	1 TB	5 TB
Amazon S3-Speichergutschriften, die alle S3-Standard-Speicherkosten in Ihrem Konto ausgleichen AWS-Konto. Nicht auf CloudFront Inhalte beschränkt und unterliegt nicht den Nutzungsbeschränkungen des Tarifs.				
Support und Zuverlässigkeit				

Leistung und Lieferung	Kostenfrei	Pro	Geschäft	Prämie
Konto- und Abrechnungssupport rund um die Uhr	Ja 	Ja 	Ja 	Ja 
One-on-one Antworten auf Konto- und Rechnungsfragen.				
Wenn Sie einen kostenpflichtigen Supportplan haben, haben Sie Anspruch auf Support für alle Flatrate-Tarife.				

Leistung und Lieferung	Kostenfrei	Pro	Geschäft	Prämie
Dokumentation und Foren AWS Support	Ja 	Ja 	Ja 	Ja 
Greifen Sie auf Produktdokumentationen, technische Unterlagen, Leitfäden mit bewährten Methoden, AWS re:Post Community -Foren und Informationen zum Servicestatus zu, um Sie bei der Planung und Fehlerbehebung zu unterstützen.				

Leistung und Lieferung	Kostenfrei	Pro	Geschäft	Prämie
Verfügbarkeit (SLA) Service Level Agreements (SLA) für Amazon CloudFront AWS WAF, Amazon Route 53 und Amazon enthalten Verpflichtungen CloudWatch zur Serviceverfügbarkeit. Falls die damit verbundenen SLA-Verpflichtungen AWS nicht eingehalten werden, haben Sie Anspruch auf eine Servicegutschrift.			Ja 	Ja 

Monatliche Nutzungsgebühren

Jeder Pauschaltarif beinhaltet ein monatliches Nutzungsentgelt, das für eine optimale Leistung auf dieser Stufe konzipiert ist. Bei den Nutzungsfreibeträgen handelt es sich nicht um feste Grenzwerte — sie entsprechen der Grundnutzung, die Ihr Tarif unterstützen soll. Sie können Ihre Nutzung

jederzeit in der CloudFront Konsole verfolgen und erhalten automatische E-Mail-Benachrichtigungen, wenn Sie 50%, 80% oder 100% Ihres Nutzungskontingents erreichen.

Monatliche Nutzungskontingente pro Abonnementstufe

	Kostenfrei	Pro	Geschäft	Premium
Anforderungen	1 M	10 M	125 M	500 M (konfigurierbar bis zu 6 B für einen neuen Pauschalpreis)
Datenübertragung	100 GB	50 TB	50 TB	50 TB (konfigurierbar bis zu 600 TB für einen neuen Pauschalpreis)

Was wird auf Ihre Nutzungsgebühr angerechnet

Blockierte DDoS-Angriffe und Anfragen, die von blockiert wurden, werden AWS WAF niemals auf Ihr Nutzungslimit angerechnet. Nur Verkehr, der Ihre AWS WAF Sicherheitsregeln überschreitet, zählt zu Ihrem Freibetrag. Dies bedeutet, dass Angriffe und unerwünschter Datenverkehr nicht zu Ihren Lasten angerechnet werden und Sie weiterhin die Möglichkeit haben, genau zu definieren, welchen Datenverkehr Ihre Anwendung blockiert oder zulässt.

Was passiert, wenn die Nutzung das Limit überschreitet

Die Tarife sind so konzipiert, dass sie flexibel sind und realen Verkehrsmustern wie Verkehrsschwankungen, organischem Wachstum und viralen Ereignissen Rechnung tragen. Und das Wichtigste: Es fallen keine Gebühren für Überschreitungen an, unabhängig davon, um wie viel Sie Ihr Limit überschreiten. Wenn Ihre Nutzung aufgrund eines anhaltenden Wachstums über mehrere Monate oder aufgrund einer ungewöhnlich hohen Nutzung in einem einzigen Monat Ihr Limit übersteigt, wird Ihr Traffic-Angebot möglicherweise angepasst. Bei den meisten Kunden kommt es nie zu Leistungsanpassungen.

Funktionsweise:

- Ihr erster Traffic-Anstieg, der das Dreifache Ihres monatlichen Tarifs ausmacht, hat keine Auswirkungen auf Ihren Service in diesem Monat. Diese einmalige Unterkunft bewältigt unerwartete Ereignisse wie virale Inhalte oder erfolgreiche Produkteinführungen, ohne Ihren Erfolg zu beeinträchtigen.
- Eine anhaltende Nutzung, die Ihr Kontingent übersteigt, wird über einen Zeitraum von 2-3 Monaten oder länger geprüft, nicht sofort. Geringfügige Schwankungen und ein moderates Wachstum werden erwartet und berücksichtigt. Lediglich ein erheblicher, anhaltender Überlastungsgrad deutet darauf hin, dass Ihre Anwendung aus Ihrer Stufe herausgewachsen ist.
- Sie erhalten jeden Monat Benachrichtigungen, wenn Sie Ihr Limit erreichen und überschreiten. Wenn Ihre Nutzung Ihren Tarif konstant und deutlich übersteigt, empfehlen wir Ihnen, auf eine Stufe umzusteigen, die Ihrem Wachstum entspricht und eine optimale Leistung bei der Skalierung gewährleistet. Sie kontrollieren Ihre Leistung, indem Sie ein Upgrade durchführen, wenn sich Ihre grundlegenden Nutzungsmuster ändern.
- Wenn Sie die Nutzungsmenge Ihres Tarifs ohne Upgrade weiterhin erheblich überschreiten, wird Ihre Traffic-Bereitstellung möglicherweise angepasst. Beispielsweise könnte Ihr Traffic von weniger oder weiter entfernten Edge-Standorten aus bedient werden, oder die Leistung könnte angepasst werden. Der Grad der Anpassung ist proportional — bei geringer übermäßiger Nutzung werden nur minimale Änderungen vorgenommen, bei größeren anhaltenden Überlastungen treten spürbarere Änderungen auf. Durch ein Upgrade wird die volle Leistung wiederhergestellt.

Bei den meisten Kunden kommt es nie zu Leistungsanpassungen. Die Pläne sind so konzipiert, dass sie normalen Wachstumsmustern Rechnung tragen.

Konfigurierbare Nutzungsberechtigungen im Premium-Tarif

Der Premium-Tarif bietet konfigurierbare monatliche Nutzungsberechtigungen pro Distribution. Wenn Sie einen Premium-Tarif abonnieren oder verwalten, können Sie aus den folgenden Stufen ein höheres monatliches Nutzungskontingent auswählen:

Premium-Nutzungsstufen

Premium-Nutzungsstufe	Monatliche Datenübertragung	Monatliche Anfragen	Flat-rate Preis pro Monat
Premium (Standard)	50 TB	500 M	1.000$
Prämie	75 TB	750 M	1.450$

Premium-Nutzungsstufe	Monatliche Datenübertragung	Monatliche Anfragen	Flat-rate Preis pro Monat
Prämie	125 TB	1,25 B	2.250$
Prämie	200 TB	2 B	3.500$
Prämie	350 TB	3,5 B	6.000$
Prämie	600 TB	6 B	10.000$

Wenn Sie eine höhere Nutzungsstufe wählen, erhöht sich Ihr monatlicher Preis und Ihr monatliches Nutzungsvolumen entsprechend. Die im Premium-Tarif enthaltenen Funktionen und Dienste bleiben auf jeder Nutzungsstufe gleich. Sie ändern nur Ihr Nutzungskontingent und Ihren Pauschalpreis. Für alle Nutzungsstufen gelten dieselben Richtlinien für Nutzungsgebühren. Details hierzu finden Sie unter Was passiert, wenn die Nutzung das Limit überschreitet.

Sie können Ihre Nutzungsstufe jederzeit in der CloudFront Konsole ändern. Wenn Sie Ihre Nutzungsstufe erhöhen, werden die Änderungen sofort wirksam. Ihr Preis und Ihr Nutzungsentgelt werden anteilig berechnet. Wenn Sie Ihre Nutzungsstufe verringern, wird die Änderung zu Beginn des nächsten Abrechnungszeitraums wirksam.

Wenn die Grundnutzung Ihrer Anwendung 6 Mrd. Anfragen oder 600 TB pro Monat übersteigt, kontaktieren Sie uns für individuelle Preise.

> **Note**
>
> Konfigurierbare Nutzungskontingente sind nur im Premium-Tarif verfügbar. Die Nutzungskontingente der Tarife Free, Pro und Business sind nicht konfigurierbar.

Die Teilnahmeberechtigung basiert auf der bisherigen Nutzung

Ihre bisherige CloudFront Nutzung kann sich darauf auswirken, ob Sie berechtigt sind, sich für bestimmte Abonnementstufen zu registrieren oder ein Downgrade auf bestimmte Tarife vorzunehmen. Wenn Ihre aktuelle Nutzung die Nutzungsgrenzen einer Tarifstufe übersteigt, müssen Sie möglicherweise eine höhere Stufe wählen, die besser zu Ihrer Arbeitsbelastung passt.

Kosten, die durch Ihren Tarif abgedeckt werden

Ihr Plan deckt die Kosten für:

- Ihr CloudFront Vertrieb
- Die AWS WAF Web-ACL, benutzerdefinierte Regeln, AWS verwaltete Regeln und die Anfragegebühren für die Web-ACL, die mit Ihrer Distribution verknüpft sind
- CloudWatch Protokolliert die Erfassung der CloudFront Zugriffsprotokolle Ihrer Distribution und der zugehörigen WAF-Protokolle
- Die von Route 53 gehostete Zone, DNS-Einträge und DNS-Abfragen, wenn sie dem Plan Ihrer Distribution beigefügt sind

Sie erhalten außerdem S3-Credits, um die S3-Standard-Speichernutzung in Ihrem Zahlungskonto auszugleichen, unabhängig davon, ob ein S3-Bucket als Ausgangspunkt für Ihre CloudFront Verteilung verwendet wird oder nicht.

Route 53, DNS-Verwaltung und Ihr Tarif

Wenn Sie Route 53 für DNS verwenden und die Zone an Ihren Plan anhängen, kann Ihr Pauschaltarif die Kosten für die gehostete Route 53-Zone beinhalten. Sie können die Zone Ihrem Plan im Abschnitt „Tarif verwalten“ Ihrer CloudFront Distribution hinzufügen. Wenn Ihre Zone an den Plan angehängt ist, deckt Ihr Plan die Standardkosten Ihrer gehosteten Zone ab, einschließlich der monatlichen Gebühr für gehostete Zonen, DNS-Einträge und DNS-Abfragegebühren, abhängig von den jeweiligen Zulagen pro Stufe, die unten aufgeführt sind. Die gehostete Zone muss die folgenden Anforderungen erfüllen:

- Existieren Sie im selben AWS Konto wie Ihre CloudFront Distribution
- Behalten Sie die Anzahl der Datensätze bei, die pro gehosteter Zone für Ihre Tarifstufe zulässig sind
- Geben Sie die Domain an, die von Ihrer CloudFront Distribution verwendet wird

Wenn Ihre gehostete Zone nicht mit Ihrem Tarif verknüpft ist, gilt weiterhin die nutzungsbasierte Bezahlung, bei der Sie für alle Standardkosten von Route 53 verantwortlich sind.

Wissenswertes über die monatlichen Zulagen für DNS-Abfragen

Wenn Ihre gehostete Zone an Ihren Tarif angehängt ist, erhalten Sie:

1. DNS-Abfragen an ALIAS-Einträge, die auf Ihre CloudFront Distribution und andere unterstützte Einträge verweisen AWS-Services
2. Ein zusätzliches monatliches Kontingent für andere DNS-Eintragstypen

	Kostenfrei	Pro	Geschäft	Prämie
DNS-Abfragen an ALIAS-Einträge (CloudFront und andere unterstützte Einträge AWS-Services) pro Monat	Kein Limit	Kein Limit	Kein Limit	Kein Limit
Zusätzliche Zulage für DNS-Abfragen pro Monat	1 M	5 M	20 M	100 M

> **Note**
>
> Um die Vorteile Ihres Plans zu maximieren, verwenden Sie ALIAS-Datensätze, um auf Ihre CloudFront Distribution zu verweisen. ALIAS-Einträge, CloudFront die auf andere unterstützte Daten verweisen, werden AWS-Services nicht auf Ihr monatliches Kontingent für DNS-Abfragen angerechnet. Alle anderen DNS-Abfragen, einschließlich CNAME-Einträge an CloudFront, werden auf Ihr DNS-Abfragevolumen angerechnet.

Überschreitung der zulässigen DNS-Abfragewerte

Wenn die Nutzung Ihrer DNS-Abfragen das monatliche Kontingent Ihres Tarifs überschreitet, werden Sie AWS möglicherweise benachrichtigt. Zu diesem Zeitpunkt können Sie Ihre gehostete Zone im Abschnitt „Abonnement verwalten“ Ihrer CloudFront Distribution vom Plan trennen, um für

die gehostete Zone wieder nutzungsbasierte Preise anzubieten. Wenn Sie Ihre gehostete Zone nach Erhalt dieser Benachrichtigung nicht trennen, wird die gehostete Zone AWS möglicherweise automatisch auf nutzungsbasierte Preise umgestellt. Wenn eine gehostete Zone auf Pay-as-you-go-Preise umgestellt wird, sind Sie für alle Standardkosten von Route 53 verantwortlich. Ihr CloudFront Vertrieb und alle anderen Tarife bleiben unverändert.

Insgesamt reduzieren AWS Kosten mit Preisplänen

CloudFront Pauschalpreispläne können Ihre AWS Gesamtkosten auf drei Arten senken:

Erstens entfallen die Kosten für die Datenübertragung zwischen CloudFront und Ihren AWS Anwendungen, die auf Diensten wie Amazon S3, AWS Application Load Balancer (ALB) oder Amazon API Gateway ausgeführt werden, automatisch. Wenn Sie Ihre AWS Anwendungen nicht direkt im Internet, CloudFront sondern über das Internet bereitstellen, deckt Ihr Pauschaltarif die Datenübertragungskosten zwischen Ihren Anwendungen und Ihren Zuschauern zu einem einfachen monatlichen Preis ab, ohne dass Sie sich Gedanken über Überschreitungen machen müssen.

Zweitens CloudFront reduziert es Ihre Rechen- und Datenbankkosten, indem Sie Ihre Anwendungsinfrastruktur schützen und die Anzahl der Anfragen reduzieren, die Ihren Ursprung erreichen. Es stellt zwischengespeicherte Inhalte von Edge-Standorten oder regionalen Edge-Caches bereit, reduziert doppelte Anfragen und blockiert bösartigen und unerwünschten Datenverkehr, bevor er Ihre Backend-Dienste erreicht. Das bedeutet, dass weniger Anfragen auf Ihre Anwendungsserver, Datenbanken und andere Server gelangen, AWS-Services die je nach Nutzung berechnet werden, was Ihre Kosten senkt.

Schließlich beinhaltet jeder Plan Amazon S3 Standard-Speicherguthaben, um die Speichernutzung für Sie auszugleichen AWS-Konto.

Um diese Einsparungen zu maximieren, konfigurieren Sie Ihre AWS Startplätze so, dass nur Traffic von akzeptiert wirdCloudFront. Verwenden Sie für S3 Origin Access Control OAC mit privaten Buckets, um Zugriff auf Ihre angegebene CloudFront Distribution zu gewähren. Beschränken Sie für Application Load Balancer-, Network Load Balancer- und Amazon EC2-Instances in privaten Subnetzen den Zugriff auf Ihre angegebene Distribution mithilfe von VPC Origins. CloudFront

Verwalten Sie Ihre Pauschalpreispläne

Sie können Ihre CloudFront Pauschalpreispläne über die CloudFront Konsole oder programmgesteuert mithilfe der AWS Command Line Interface (AWS CLI) oder der API verwalten.

PricingPlanManager In diesem Abschnitt wird beschrieben, wie Sie mithilfe der Konsole Preispläne abonnieren, Abonnementstufen ändern und Abonnements kündigen. CloudFront

Informationen zur programmgesteuerten Verwaltung von Preisplänen finden Sie unter Erste Schritte mit der API. PricingPlanManager

Abonnieren Sie einen Preisplan für eine neue Distribution

Wenn Sie eine neue Distribution erstellen, können Sie einen Preisplan abonnieren.

Um einen neuen Vertrieb für einen Preisplan zu abonnieren

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home.
2. Wählen Sie im Navigationsbereich Distributionen aus und folgen Sie dann den Schritten, um eine Distribution zu erstellen.
3. Wählen Sie den Preisplan Ihrer Distribution aus. Beachten Sie, dass einige Funktionen nicht pro Preisstufe verfügbar sind. Sehen Sie sich die Funktionen der einzelnen Tarife an und wählen Sie den Preisplan aus, den Sie für Ihre Anwendung benötigen.
4. Gehen Sie wie folgt vor, um Ihre Distribution zu erstellen.

Abonnieren Sie für eine bestehende Distribution einen Preisplan

Wenn Sie eine Distribution aktualisieren, können Sie einen Preisplan abonnieren. Bevor Sie einen Preisplan auswählen, stellen Sie sicher, dass Ihre Vertriebskonfiguration mit dem gewünschten Plan kompatibel ist.

> **Tip**
>
> Wenn Ihre aktuelle Distribution Funktionen verwendet, die nicht unterstützt werden, müssen Sie diese Funktionen deaktivieren, bevor Sie den Preisplan abonnieren können. Dazu gehört auch das Deaktivieren von Funktionen wie Zugriffsprotokollen in Echtzeit.

Sobald Ihre Distributionskonfiguration kompatibel ist, können Sie beim Aktualisieren einer Distribution den gewünschten Preisplan auswählen.

Um für eine bestehende Distribution einen Preisplan zu abonnieren

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home.
2. Wählen Sie im Navigationsbereich Distributionen aus und folgen Sie dann den Schritten, um eine vorhandene Distribution zu aktualisieren.
3. Wählen Sie den Preisplan Ihrer Distribution aus. Beachten Sie, dass einige Funktionen nicht pro Preisstufe verfügbar sind. Sehen Sie sich die Funktionen der einzelnen Tarife an und wählen Sie den Preisplan aus, den Sie für Ihre Anwendung benötigen.
4. Führen Sie die Schritte aus, um Ihre Distribution zu aktualisieren.

Aktualisieren Sie einen Preisplan

Wir empfehlen Ihnen, einen Tarif zu aktualisieren, wenn Sie sich Ihrem monatlichen Nutzungskontingent nähern oder dieses überschritten haben oder wenn Sie eine Funktion aktivieren möchten, die in der nächsten Stufe verfügbar ist.

Wenn Sie ein Upgrade auf eine höhere Tarifstufe durchführen, werden die Änderungen sofort wirksam. Ihr Preis und Ihr Nutzungskontingent werden anteilig berechnet. Ihre Distribution und die damit verbundenen Ressourcen haben Zugriff auf die verfügbaren Funktionen und das höhere Nutzungsvolumen Ihres neuen Tarifs.

Um einen Preisplan zu aktualisieren

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home.
2. Rufen Sie im Navigationsbereich Distributions auf.
3. Wählen Sie Ihre Distribution aus, die einen bestehenden Preisplan abonniert hat.
4. Folgen Sie den Anweisungen, um den Preisplan Ihrer Distribution zu aktualisieren.
5. Führen Sie die Schritte aus, um eine bestehende Distribution zu aktualisieren.

Ändern Sie die Nutzungsstufe Ihres Premium-Tarifs

Wenn Sie den Premium-Tarif nutzen, können Sie Ihre Nutzungsstufe ändern, um Ihr Nutzungskontingent zu erhöhen oder zu verringern.

Wenn Sie Ihre Nutzungsstufe erhöhen, wird die Änderung sofort wirksam und Ihr Preis wird anteilig berechnet. Wenn Sie Ihr Nutzungsniveau verringern, wird die Änderung zu Beginn des nächsten Abrechnungszeitraums wirksam.

Um die Nutzungsstufe Ihres Premium-Tarifs zu ändern

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home.
2. Rufen Sie im Navigationsbereich Distributions auf.
3. Wählen Sie Ihre Distribution aus, die einen Premium-Preisplan abonniert hat.
4. Wählen Sie die Schaltfläche „Plan verwalten“.
5. Wählen Sie im Abschnitt Abo verwalten die Option Abo ändern aus.
6. Folgen Sie den Anweisungen, um Ihre Nutzungsstufe für den Premium-Preisplan zu ändern. In der Konsole werden die verfügbaren Nutzungsstufen und die entsprechenden monatlichen Preise angezeigt.

Einen Preisplan herabstufen

Wir empfehlen Ihnen, auf eine niedrigere Tarifstufe herunterzustufen, wenn Sie die zusätzlichen Funktionen Ihrer bestehenden Tarife nicht benötigen. Beispielsweise könnten Sie ein Downgrade durchführen, wenn Sie davon ausgehen, dass Ihre Anwendung weniger Traffic haben wird.

Wenn Sie auf eine niedrigere Stufe herabstufen, werden Ihre Abrechnungsänderungen zu Beginn des nächsten Abrechnungszeitraums wirksam.

Wenn Ihre Distribution derzeit die Nutzungsmenge für einen Tarif überschreitet, können Sie ein Downgrade durchführen, sobald Ihre Nutzung innerhalb der Nutzungsmenge für Ihre gewünschte Stufe liegt. Um zu vermeiden, dass Ihnen Ihre bestehende Tarifstufe im nächsten Abrechnungszeitraum in Rechnung gestellt wird, führen Sie ein Downgrade vor Ende des Monats durch.

Um einen Preisplan herabzustufen

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home.
2. Rufen Sie im Navigationsbereich Distributions auf.
3. Wählen Sie Ihre Distribution aus, die einen bestehenden Preisplan abonniert hat.

4. Folgen Sie den Anweisungen, um den Preisplan Ihrer Distribution herabzustufen. Wenn Sie Funktionen haben, die nicht unterstützt werden, müssen Sie entweder die Funktion oder Ressource aus der Distribution entfernen.
5. Führen Sie die Schritte aus, um eine vorhandene Distribution zu aktualisieren.

Stornieren Sie einen Preisplan

Wenn Sie einen kostenpflichtigen Tarif kündigen, behalten Sie Ihren Pauschalpreis bis zum Ende Ihres aktuellen Abrechnungszeitraums bei. Ihr Vertrieb und alle damit verbundenen Tarifressourcen werden dann zu Beginn des nächsten Abrechnungszeitraums auf nutzungsbasierte Bezahlung umgestellt. Kostenlose Tarife werden sofort storniert.

Um einen Preisplan zu kündigen

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home.
2. Rufen Sie im Navigationsbereich Distributions auf.
3. Wählen Sie Ihre Distribution aus, die einen bestehenden Preisplan abonniert hat.
4. Folgen Sie den Anweisungen, um den Preisplan Ihrer Distribution zu kündigen.
5. Führen Sie die Schritte aus, um eine bestehende Distribution zu aktualisieren.

Stornieren Sie eine ausstehende Planänderung

Wenn Sie Ihren Pauschaltarif herabgestuft oder gekündigt haben, müssen Sie bis zum Ende des aktuellen Abrechnungszeitraums warten, bevor Ihre Änderungen in Kraft treten. Um Ihren bestehenden Pauschaltarif beizubehalten oder Ihren Tarif erneut hochzustufen oder herabzustufen, müssen Sie zunächst Ihre ausstehende Planänderung stornieren.

Um eine ausstehende Änderung des Preisplans zu stornieren

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home.
2. Rufen Sie im Navigationsbereich Distributions auf.
3. Wählen Sie Ihre Distribution aus, die einen bestehenden Preisplan abonniert hat.
4. Folgen Sie den Anweisungen, um die ausstehende Planänderung Ihrer Distribution zu stornieren.

5. Wählen Sie den Preisplan, den Sie für Ihren Vertrieb wünschen.
6. Führen Sie die Schritte aus, um eine bestehende Distribution zu aktualisieren.

Löschen einer Verteilung mit einem Preisplan

Sie können keine Distribution löschen, die einen Preisplan abonniert hat. Sie müssen zuerst den Preisplan kündigen. Stornierungen von kostenpflichtigen Tarifen werden nach dem aktuellen Abrechnungszeitraum wirksam. Kostenlose Tarife werden sofort storniert. Nachdem der Plan gekündigt wurde, können Sie die Verteilung löschen.

Um eine Verteilung mit einem Preisplan zu löschen

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home.
2. Rufen Sie im Navigationsbereich Distributions auf.
3. Folgen Sie den vorherigen Schritten, um den Preisplan der Distribution zu kündigen.
4. Folgen Sie den Schritten, um die Verteilung zu löschen.

> **Note**
>
> Sie können eine Distribution deaktivieren, die einen Preisplan abonniert hat, aber für diesen Plan fallen trotzdem Gebühren an. Damit für Ihren Tarif keine Gebühren mehr anfallen, müssen Sie ihn zunächst kündigen.

Berechtigungen

Um Preisplan-Abonnements für Ihre CloudFront Distributionen einsehen oder verwalten zu können, benötigen Sie die erforderlichen Berechtigungen. Weitere Informationen erhalten Sie unter AWS verwaltete Richtlinie: CloudFrontFullAccess und AWS verwaltete Richtlinie: CloudFrontReadOnlyAccess.

Flat-rate Kontingente für Preispläne

Die folgende Tabelle zeigt die Kontingente und Einschränkungen für CloudFront Pauschalpreispläne.

> **Note**
>
> Diese Kontingente können für Sie AWS-Konto nicht erhöht werden.

Account-level Kontingente	Kontingente
Preispläne pro AWS-Konto	100
Kostenlose Pläne pro AWS-Konto	3
Apex-level Domains pro Plan	1

Nicht unterstützte Funktionen

Bevor Sie eine Distribution einem Preisplan zuordnen können, müssen Sie sicherstellen, dass bestimmte Funktionen deaktiviert und Verknüpfungen entfernt werden.

> **Hinweise**
>
> - Wenn für Ihre Distribution oder Ihr Konto eine dieser Einschränkungen gilt, müssen Sie diese beheben, bevor Sie Preispläne verwenden können. Nachdem Sie Änderungen an Ihrer Distribution vorgenommen haben, warten Sie, bis die Änderungen auf alle Edge-Standorte übertragen werden.
> - Wenn Sie einen Preisplan verwenden, muss Ihrer Distribution eine AWS WAF Web-ACL zugeordnet sein. Diese Ressource kann nicht entfernt oder von Ihrer Distribution getrennt werden, es sei denn, Sie wechseln für diese Distribution zur nutzungsbasierten Preisgestaltung.

Nicht unterstützte Funktionen

Sie können keinen Preisplan für Distributionen abonnieren, deren Konfiguration die folgenden nicht unterstützten Funktionen enthält. Sie können die nicht unterstützte Funktion deaktivieren und eine alternative Option verwenden oder die nutzungsbasierte Bezahlung für Ihre Distribution beibehalten.

Nicht unterstützte Funktionen	Alternative Optionen	AWS-Service
Multi-tenantVerteilungen	Verwenden Sie eine Standarddistribution oder nutzungsbasierte Preisgestaltung	CloudFront
Kontinuierliche Bereitstellung und Staging-Distributionen ???	Verwenden Sie nutzungsbasierte Preise	CloudFront
Konfiguration der Anycast-IP-Liste	Verwenden Sie Preise mit nutzungsbasierter Bezahlung	CloudFront
Real-time auf Protokolle zugreifen	Verwenden Sie Standardzugriffsprotokolle oder nutzungsbasierte Preise	CloudFront
Gezielte Bots	Verwenden Sie gängige Bots oder nutzungsbasierte Preise	AWS WAF
Vom Partner verwaltete Regeln	Verwenden Sie Preise mit nutzungsbasierter Bezahlung	AWS WAF
Kontoerstellung und Betrugsprävention	Verwenden Sie nutzungsbasierte Preise	AWS WAF
Schutz vor Kontoübernahmen	Nutzen Sie nutzungsbasierte Preise	AWS WAF
Regelgruppen	Erstellen Sie individuelle Regeln (Regelgruppen sind gemeinsam genutzte AWS WAF Regeln, die auf eine Web-ACL angewendet werden	AWS WAF

Nicht unterstützte Funktionen	Alternative Optionen	AWS-Service
	können, ähnlich wie Richtlinien auf) CloudFront	
Ältere Funktionen		
ForwardedValues-Konfiguration	Verwende Origin-Anforderungsrichtlinien	CloudFront
Dediziert IP/SSL	Verwenden Sie nutzungsbasierte Preise	CloudFront
Verschlüsselung auf Feldebene	Verwenden Sie nutzungsbasierte Preise	CloudFront
AWS Identity and Access Management (IAM)-Serverzertifikate	Verwenden Sie (ACM) -Zertifikate AWS Certificate Manager	CloudFront
Origin Access Identity (OAI)	Verwenden Sie Origin Access Control (OAC)	CloudFront
Ältere Cache-Einstellungen	Verwenden Sie Cache-Richtlinien und Origin-Request-Richtlinien.	CloudFront

Nicht unterstützte Zuordnungen

Sie können für eine Verteilung keinen Preisplan abonnieren, wenn die Verteilung bereits mit einer der folgenden Ressourcen verknüpft ist, die bereits anderen Distributionen zugeordnet sind. Ressourcen, die einer Verteilung zugeordnet sind, die einen Preisplan abonniert hat, können nur für diese Verteilung verwendet werden. Wenn Sie beispielsweise eine CloudFront Funktion haben, die einen Schlüsselwertspeicher verwendet, können weder die Funktion noch der Schlüsselwertspeicher für eine Verteilung gemeinsam genutzt werden, die in einem Preisplan enthalten ist.

- CloudFront Funktionen
- CloudFront Funktionen, die einem Schlüsselwertspeicher zugeordnet sind
- AWS WAF Web-ACLs

Um für eine Distribution einen Preisplan zu abonnieren, entfernen Sie entweder die zugehörige Ressource oder ersetzen Sie sie durch eine andere.

Account-level Einschränkungen

AWS-Konten kommen nicht für Preispläne in Frage, wenn sie eine der folgenden Bedingungen erfüllen:

- Sie haben die maximal zulässige Anzahl an Abonnements erreicht. Siehe Flat-rate Kontingente für Preispläne.
- Ihr Konto wird verwendet Kostenloses AWS-Kontingent.

Resource-level Einschränkungen

Ausschüttungen kommen nicht für Preispläne in Frage, wenn sie eine der folgenden Bedingungen erfüllen:

- Ihre Distribution wurde aktiviert AWS Shield Advanced
- Ihre Distribution hat den Firewall Manager-Dienst für Ihre Web-ACL aktiviert. Firewall Manager verwaltet die WebACL Ihrer CloudFront Distribution nicht im Rahmen eines Preisplans.

Zusätzliche Funktionen, die sich auf Ihren Preisplan auswirken können

Flat-rate Preispläne ermöglichen es Ihnen, einen Pauschalpreis für Ihre CloudFront Distribution und die oben aufgeführten Funktionen zu zahlen, die sowohl in Ihrem Plan enthalten als auch mit Ihrer CloudFront Distribution verknüpft sind. Für alle anderen Funktionen können zusätzliche Gebühren anfallen, einschließlich, aber nicht beschränkt auf die folgenden:

CloudFront

- Aufrufe der Lambda @Edge -Funktion

AWS WAF

- CAPTCHA-Puzzles, die mit der JavaScript API erstellt wurden, werden mit nutzungsbasierter Bezahlung in Rechnung gestellt. CAPTCHA-Antworten, die in Ihren AWS WAF Regeln konfiguriert sind (der häufigste Anwendungsfall), sind in Ihrem Plan ohne zusätzliche Kosten enthalten.

Route 53

- Route 53 DNSSEC ist kostenpflichtig AWS KMS
- Route 53 IP (CIDR) -Blöcke (die ersten 1.000 sind pro Block kostenlos) AWS-Konto
- Route 53 Health Checks (die ersten 50 sind jeweils kostenlos) AWS-Konto

Funktionen zur Protokollierung

- Route 53 DNS-Abfrageprotokolle, CloudFront Funktionsprotokolle und CloudFront Verbindungsfunktionsprotokolle
- AWS WAF Übermittlung von Protokollen an Amazon S3
- CloudFront oder die Lieferung von AWS WAF Protokollen an Amazon Data Firehose
- Zusätzliche CloudWatch Metriken für CloudFront
- CloudFront Zugriffsprotokolle im Parquet-Format

> **Note**
>
> Ihr Plan beinhaltet die Aufnahme von Amazon CloudWatch Logs für CloudFront Standardprotokolle (Zugriffsprotokolle) und WAF-Protokolle ohne zusätzliche Kosten. Alle anderen CloudWatch Kosten wie Speicherung und Abfragen sind nicht in Ihrem Plan enthalten. Alle anderen Protokolle werden ebenfalls separat in Rechnung gestellt.

> **Note**
>
> Ihr Plan beinhaltet öffentlich autoritatives DNS von Route 53. Wenn Ihre gehostete Route 53-Zone mit Ihrem Tarif verknüpft ist, deckt Ihr Tarif die Standardkosten Ihrer gehosteten Zone ab, einschließlich der monatlichen Gebühr für gehostete Zonen, DNS-Einträge und DNS-Abfragegebühren, sofern die jeweiligen Gebühren pro Stufe anfallen. Alle anderen Kosten,

die sich aus der Nutzung von Route 53 ergeben, und Funktionen, die oben nicht in Ihrem Plan enthalten sind, werden nicht von Ihrem Plan abgedeckt.

Preispläne im Vergleich zu Pay-as-you-go-Preisen

Flat-rate Tarife und nutzungsbasierte Preise bieten je nach Bedarf unterschiedliche Vorteile. Bei Flatrate-Tarifen zahlen Sie einen Preis, der die mehrfache Erfassung von AWS-Services Like-CloudFront, AWS WAF Route 53- und CloudWatch Logs beinhaltet, und es fallen keine Gebühren für Überschreitungen an, auch nicht bei Verkehrsspitzen oder Angriffen.

Bei nutzungsbasierten Preisen wird Ihnen jeder Dienst und jede Funktion auf der Grundlage Ihrer tatsächlichen Nutzung separat in Rechnung gestellt. Dies bietet zwar vollständige Flexibilität bei der Serviceauswahl und -konfiguration, Ihre Kosten können jedoch je nach Verkehrsmuster von Monat zu Monat variieren. Sie müssen die Nutzung mehrerer Dienste überwachen, um die Kosten verwalten zu können.

Flat-rate Tarife sind ideal, wenn Sie eine kombinierte monatliche Abrechnung, eine vereinfachte Servicekonfiguration und integrierte Sicherheitsfunktionen wünschen, ohne sich Gedanken über Mehrkosten machen zu müssen. Pay-as-you-go Die Preisgestaltung ist die bessere Wahl, wenn Sie die vollständige Kontrolle über einzelne Servicefunktionen und benutzerdefinierte Konfigurationen benötigen, Zugriff auf Funktionen benötigen, die in Pauschaltarifen nicht verfügbar sind, oder wenn Sie erwarten, dass große, vorhersehbare Datenverkehrsspitzen bewältigt werden müssen. Die CloudFront Pauschalpreise von Amazon können nicht mit anderen Angeboten, Aktionen oder Rabatten kombiniert werden.

Konfigurieren von Distributionen

Sie erstellen eine CloudFront Amazon-Distribution, um mitzuteilen, CloudFront von wo aus Inhalte geliefert werden sollen, und um zu erfahren, wie Sie die Inhaltszustellung verfolgen und verwalten können.

Wenn Sie Ihre CloudFront Distribution erstellen, konfiguriert die meisten Vertriebseinstellungen CloudFront automatisch für Sie, basierend auf der Herkunft Ihrer Inhalte. Weitere Informationen zu vorkonfigurierten Einstellungen finden Sie unter Referenz für vorkonfigurierte Distributionseinstellungen. Optional können Sie Ihre Distributionseinstellungen manuell bearbeiten. Weitere Informationen finden Sie unter Referenz für alle Distributionseinstellungen.

Folgende Einstellungen können konfiguriert werden:

- Ihr Inhaltsursprung — Der Amazon S3 S3-Bucket, AWS Elemental MediaPackage Kanal, AWS Elemental MediaStore Container, Elastic Load Balancing Load Balancer oder HTTP-Server, von dem die CloudFront zu verteilenden Dateien abgerufen werden. Sie können eine beliebige Kombination von bis zu 25 Ursprüngen für eine einzelne Verteilung angeben.
- Zugriff – Ob Sie möchten, dass die Dateien öffentlich verfügbar sind, oder ob Sie den Zugriff auf bestimmte Benutzer beschränken möchten.
- Sicherheit – Ob Sie den AWS WAF -Schutz aktivieren und möchten, dass Benutzer für den Zugriff auf Ihre Inhalte HTTPS verwenden müssen. Bei Distributionen mit mehreren Mandanten werden nur AWS WAF V2-Webzugriffskontrolllisten (ACLs) unterstützt.
- Cache-Schlüssel – Welche Werte, wenn überhaupt, in den Cache-Schlüssel eingefügt werden sollen. Der Cache-Schlüssel identifiziert eindeutig jede Datei im Cache für eine bestimmte Verteilung.
- Einstellungen für Origin-Anfragen — Gibt CloudFront an, ob Sie HTTP-Header, Cookies oder Abfragezeichenfolgen in Anfragen einbeziehen möchten, die an Ihren Ursprung gesendet werden.
- Geografische Einschränkungen — Ob Sie verhindern CloudFront möchten, dass Nutzer in ausgewählten Ländern auf Ihre Inhalte zugreifen.
- Protokolle — Ob Sie Standardprotokolle oder Echtzeit-Zugriffsprotokolle erstellen CloudFront möchten, die die Zuschaueraktivitäten aufzeigen.

Weitere Informationen finden Sie unter Referenz für alle Distributionseinstellungen.

Die aktuelle maximale Anzahl von Verteilungen, die Sie für jede Verteilung erstellen können AWS-Konto, finden Sie unter. Allgemeine Kontingente für Verteilungen Die Anzahl von Dateien, die Sie pro Verteilung bereitstellen können, ist nicht begrenzt.

Sie können Verteilungen verwenden, um die folgenden Inhalte über HTTP oder HTTPS bereitzustellen:

- Statische und dynamische Inhalte wie HTML- JavaScript, CSS- und Bilddateien werden über HTTP oder HTTPS heruntergeladen.
- Video-on-Demand in verschiedenen Formaten, wie Apple HTTP Live Streaming (HLS) und Microsoft Smooth Streaming. (Bei Multi-Tenant-Distributionen wird Smooth Streaming nicht unterstützt.) Weitere Informationen finden Sie unter Stellen Sie Video auf Abruf bereit mit CloudFront.
- Live-Events, z. B. ein Meeting, eine Konferenz oder ein Konzert – in Echtzeit. Für Live-Streaming können Sie die Distribution mithilfe eines CloudFormation Stacks automatisch erstellen. Weitere Informationen finden Sie unter Stellen Sie Videostreaming mit CloudFront und bereit AWS Mediendienste.

In den folgenden Themen finden Sie weitere Informationen zu CloudFront Distributionen und dazu, wie Sie sie entsprechend Ihren Geschäftsanforderungen konfigurieren können. Weitere Informationen zum Erstellen einer Verteilung finden Sie unter Eine Distribution erstellen.

Themen

- Verständnis der Funktionsweise von Multi-Tenant-Distributionen
- Eine Distribution erstellen
- Referenz für vorkonfigurierte Distributionseinstellungen
- Referenz für alle Distributionseinstellungen
- Testen einer Distribution
- Eine Verteilung aktualisieren
- Kennzeichnen Sie eine Ressource CloudFront
- Löschen einer -Verteilung
- Verwenden Sie bei Verteilungen verschiedene Ursprünge CloudFront
- IPv6 für CloudFront Distributionen aktivieren
- Verwenden Sie CloudFront Continuous Deployment, um CDN-Konfigurationsänderungen sicher zu testen

- Verwenden von benutzerdefinierten URLs durch Hinzufügen von alternativen Domainnamen (CNAMEs)
- WebSockets Mit CloudFront Distributionen verwenden
- Anfordern statischer Anycast-IPs zur Verwendung für Zulassungslisten
- gRPC mit CloudFront Distributionen verwenden

Verständnis der Funktionsweise von Multi-Tenant-Distributionen

Sie können CloudFront Mehrmandantenverteilungen mit Einstellungen erstellen, die für mehrere Verteilungsmandanten wiederverwendet werden können. Bei einer Verteilung mit mehreren Mandanten können Sie Ihre Verteilungseinstellungen auf der Grundlage Ihres Inhaltstyps CloudFront konfigurieren lassen. Weitere Informationen zu vorkonfigurierten Einstellungen finden Sie unter Referenz für vorkonfigurierte Distributionseinstellungen.

Informationen zur Umstellung einer vorhandenen Standardverteilung auf eine Mehrmandantenverteilung finden Sie unter. Migrieren zu einer Multi-Tenant-Distribution

Die Verwendung einer Multi-Tenant-Distribution anstelle einer Standarddistribution bietet unter anderem folgende Vorteile:

- Verringerung des betrieblichen Aufwands
- Wiederverwendbare Konfigurationen für Webadministratoren und Softwareanbieter zur Verwaltung der CloudFront Verteilung mehrerer Webanwendungen, die Inhalte für Endbenutzer bereitstellen.
- Verbesserte Integrationen mit anderen AWS-Services , um automatisiertes Zertifikatsmanagement, einheitliche Sicherheitskontrollen und problemlose Konfigurationskontrolle im großen Maßstab zu ermöglichen.
- Aufrechterhaltung konsistenter Ressourcenmuster in Ihren Implementierungen. Definieren Sie Einstellungen, die übernommen werden müssen, und geben Sie dann Anpassungen an, um die Einstellungen zu überschreiben.
- Anpassbare Ursprungs- und Sicherheitseinstellungen, um spezifische Anforderungen auf der Ebene des Distributionsmandanten zu erfüllen
- Einteilung Ihrer Distributionsmandanten in verschiedene Stufen. Wenn beispielsweise nur einige der Distributionsmandanten Origin Shield benötigen, können Sie die Distributionsmandanten in verschiedene Multi-Tenant-Distributionen gruppieren.
- Nutzung derselben DNS-Konfiguration für mehrere Domains

Im Gegensatz zu einer Standardverteilung kann auf eine Multi-Tenant-Distribution nicht direkt zugegriffen werden, da sie keinen Routing-Endpunkt hat. CloudFront Daher muss sie in Verbindung mit einer Verbindungsgruppe und einem oder mehreren Distributionsmandanten verwendet werden. Standardverteilungen haben zwar ihren eigenen CloudFront Endpunkt und können von Endbenutzern direkt aufgerufen werden, sie können jedoch nicht als Vorlage für andere Distributionen verwendet werden.

Informationen zu Multi-Tenant-Distributionskontingenten finden Sie unter Kontingente zu Multi-Tenant-Distributionen.

Themen

- Funktionsweise
- Bedingungen
- Nicht unterstützte Funktionen
- Anpassungen von Distributionsmandanten
- Fordern Sie Zertifikate für Ihren CloudFront Distributionsmandanten an
- Erstellen einer benutzerdefinierten Verbindungsgruppe (optional)
- Migrieren zu einer Multi-Tenant-Distribution

Funktionsweise

In einer Standarddistribution enthält die Distribution alle Einstellungen, die Sie für Ihre Website oder Anwendung aktivieren möchten, z. B. die Ursprungskonfigurationen, das Cacheverhalten und die Sicherheitseinstellungen. Wenn Sie eine separate Website erstellen und zu einem Großteil die gleichen Einstellungen verwenden möchten, müssten Sie eine neue Distribution erstellen.

CloudFront Mehrmandantenverteilungen unterscheiden sich darin, dass Sie eine anfängliche Mehrmandantenverteilung erstellen können. Für jede neue Website erstellen Sie einen Distributionsmandanten, der automatisch die definierten Werte der Quelldistribution erbt. Anschließend passen Sie spezifische Einstellungen für Ihren Distributionsmandanten an.

-Übersicht

1. Zu Beginn erstellen Sie zunächst eine Mehrmandantenverteilung. CloudFront konfiguriert Ihre Verteilungseinstellungen für Sie auf der Grundlage Ihres Inhaltstyps. Sie können die Einstellungen für alle Ursprünge außer VPC-Ursprüngen anpassen. Die VPC-Ursprungseinstellungen werden

auf der VPC-Ursprungsressource selbst angepasst. Weitere Informationen zu den anpassbaren Einstellungen für Multi-Tenant-Distributionen finden Sie unter Referenz für vorkonfigurierte Distributionseinstellungen.

- Das TLS-Zertifikat, das Sie für die Multi-Tenant-Distribution verwenden, kann von Ihren Distributionsmandanten übernommen werden. Die Multi-Tenant-Distribution selbst ist nicht routingfähig, sodass ihr kein Domainname zugeordnet ist.

2. CloudFront Erstellt standardmäßig eine Verbindungsgruppe für Sie. Die Verbindungsgruppe steuert, mit welcher Verbindung Zuschaueranfragen nach Inhalten verbunden CloudFront werden. Sie können einige Routing-Einstellungen in der Verbindungsgruppe anpassen.

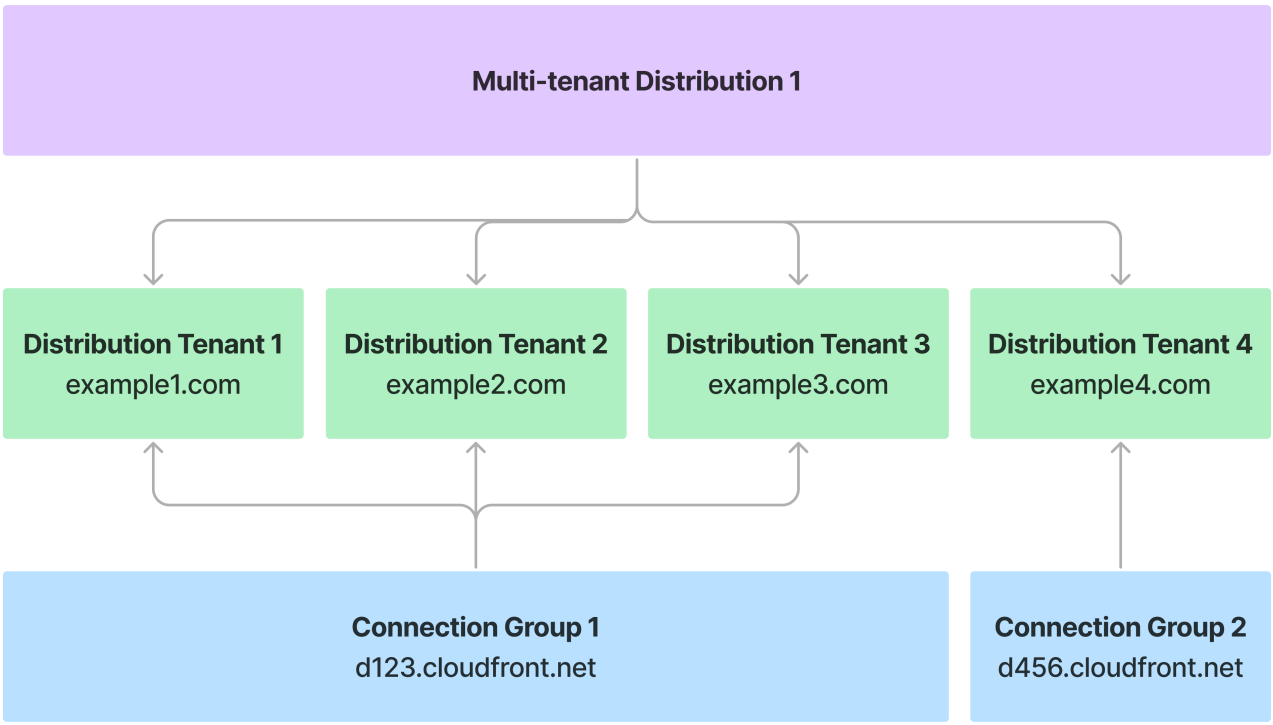
   Sie können dies ändern, indem Sie manuell Ihre eigene Verbindungsgruppe erstellen. Weitere Informationen finden Sie unter Erstellen einer benutzerdefinierten Verbindungsgruppe (optional).
3. Erstellen Sie anschließend einen oder mehrere Distributionsmandanten. Der Distributionsmandant ist die „Eingangstür“, über die Viewer auf Ihre Inhalte zugreifen können. Jeder Verteilungsmandant verweist auf die Mehrmandantenverteilung und wird automatisch der Verbindungsgruppe zugeordnet, die für Sie CloudFront erstellt wurde. Der Distributionsmandant unterstützt eine einzelne Domain oder Subdomain.
4. Anschließend können Sie einige Einstellungen für den Distributionsmandanten anpassen, z. B. Vanity-Domains und Ursprungspfade. Weitere Informationen finden Sie unter Anpassungen von Distributionsmandanten.
5. Schließlich müssen Sie den DNS-Eintrag in Ihrem DNS-Host aktualisieren, um den Datenverkehr an den Distributionsmandanten weiterzuleiten. Rufen Sie dazu den CloudFront Endpunktwert aus Ihrer Verbindungsgruppe ab und erstellen Sie einen CNAME-Eintrag, der auf den CloudFront Endpunkt verweist.

Example Beispiel

Die folgende Grafik zeigt, wie eine Multi-Tenant-Distribution, Distributionsmandanten und Verbindungsgruppen zusammenarbeiten, um Ihren Viewern Inhalte für mehrere Domains bereitzustellen.

1. Die Multi-Tenant-Distribution definiert für jeden Distributionsmandanten die übernommenen Einstellungen. Sie verwenden die Multi-Tenant-Distribution als Vorlage.
2. Jeder aus der Multi-Tenant-Distribution erstellte Distributionsmandant hat seine eigene Domain.
3. Die Verteilungsmandanten werden automatisch der Verbindungsgruppe hinzugefügt, die Sie bei der CloudFront Erstellung der Mehrmandantenverteilung für Sie erstellt haben.

Verbindungsgruppen steuern, wie Viewer-Anfragen mit dem CloudFront Netzwerk verbunden werden.



Eine ausführliche Anleitung zur Erstellung von Multi-Tenant-Distributionen finden Sie unter Erstellen Sie eine CloudFront Distribution in der Konsole.

Bedingungen

Die folgenden Konzepte beschreiben Komponenten von Multi-Tenant-Distributionen:

Multi-tenant Verteilung

Eine Blueprint-Verteilung für mehrere Mandanten, die alle gemeinsam genutzten Konfigurationseinstellungen für alle Verteilungsmandanten spezifiziert, einschließlich Cache-Verhalten, Sicherheitsvorkehrungen und Herkunft. Multi-tenant Verteilungen können den Datenverkehr nicht direkt bedienen. Sie müssen in Verbindung mit Verbindungsgruppen und Distributionsmandanten verwendet werden.

Standarddistribution

Eine Distribution, die keine Multi-Tenant-Funktionalität bietet. Diese Distributionen eignen sich am besten für die Unterstützung einzelner Websites oder Apps.

Distributionsmandant

Ein Distributionsmandant übernimmt die Konfiguration der Multi-Tenant-Distribution. Einige Konfigurationseinstellungen können auf der Ebene des Distributionsmandanten angepasst werden. Der Distributionsmandant muss über ein gültiges TLS-Zertifikat verfügen, das von der Multi-Tenant-Distribution übernommen werden kann, sofern es die Domain oder Subdomain des Distributionsmandanten abdeckt.

Der Verteilungsmandant muss einer Verbindungsgruppe zugeordnet sein. CloudFront erstellt eine Verbindungsgruppe für Sie, wenn Sie einen Verteilungsmandanten erstellen, und weist dieser Verbindungsgruppe automatisch alle Verteilungsmandanten zu.

Multi-tenancy

Sie können die Multi-Tenant-Distribution verwenden, um Domain-übergreifend Inhalte bereitzustellen und die gleiche Konfiguration und Infrastruktur zu verwenden. Durch diesen Ansatz können verschiedene Domains (sogenannte Mandanten) gemeinsame Einstellungen aus der Multi-Tenant-Distribution verwenden und gleichzeitig ihre individuellen Anpassungen beibehalten.

Verbindungsgruppe

Stellt den CloudFront Routing-Endpunkt bereit, der Inhalte für Zuschauer bereitstellt. Sie müssen jeden Verteilungsmandanten einer Verbindungsgruppe zuordnen, um den entsprechenden CloudFront Routing-Endpunkt für den CNAME-Datensatz zu erhalten, den Sie für Ihre Verteilermandantendomäne oder -subdomäne erstellen. Verbindungsgruppen können von mehreren Distributionsmandanten gemeinsam genutzt werden. Verbindungsgruppen verwalten die Routing-Einstellungen für Distributionsmandanten, wie z. B. IPv6- und Anycast-IP-Listeneinstellungen.

Parameters

Eine Liste von Schlüssel-Wert-Paaren für Platzhalterwerte, z. B. Ursprungspfade und Domainnamen. Sie können Parameter in Ihrer Multi-Tenant-Distribution definieren und Werte für diese Parameter auf der Ebene des Distributionsmandanten angeben. Sie wählen aus, ob die Parameterwerte für den Distributionsmandanten eingegeben werden müssen.

Wenn Sie für einen optionalen Parameter in einem Distributionsmandanten keinen Wert angeben, wird der Standardwert aus der Multi-Tenant-Distribution verwendet.

CloudFront Routing-Endpunkt

Kanonisches DNS für die Verbindungsgruppe, wie z. B. `d123.cloudfront.net`. Wird im CNAME-Datensatz für Ihre Distributionsmandanten-Domain oder -Subdomain verwendet.

Anpassungen

Sie können Ihre Distributionsmandanten so anpassen, dass sie andere Einstellungen als die Multi-Tenant-Distribution verwenden. Für jeden Verteilungsmandanten können Sie eine andere AWS WAF Web Access Control List (ACL), TLS-Zertifikate und geografische Einschränkungen angeben.

Nicht unterstützte Funktionen

Die folgenden Features können nicht mit einer Multi-Tenant-Distribution verwendet werden. Wenn Sie eine neue Multi-Tenant-Distribution mit denselben Einstellungen wie Ihre Standarddistribution erstellen möchten, beachten Sie, dass einige Einstellungen nicht verfügbar sind.

> **Hinweise**
>
> - Derzeit gelten AWS Firewall Manager Richtlinien nur für Ihre Standardverteilungen. Firewall Manager wird in einer zukünftigen Version Multi-Tenant-Distributionen unterstützen.
> - Im Gegensatz zu Standarddistributionen geben Sie Ihren Domainnamen (Alias) auf der Ebene des Distributionsmandanten an. Weitere Informationen finden Sie unter Fordern Sie Zertifikate für Ihren CloudFront Distributionsmandanten an und zur CreateDistributionTenantAPI-Operation.

- Fortlaufende Bereitstellung
- Ursprungszugriffsidentität (OAI) – Verwenden Sie stattdessen Ursprungszugriffssteuerung (OAC).
- Dedizierte benutzerdefinierte IP-SSL-Unterstützung – Nur die `sni-only`-Methode wird unterstützt.
- AWS WAF Klassische Web-ACL (V1) — Nur AWS WAF V2-Web-ACLs werden unterstützt.
- Standardprotokollierung (Legacy)
- Minimale TTL
- Standard-TTL
- Maximale TTL

- ForwardedValues
- PriceClass
- Vertrauenswürdige Signaturgeber
- Smooth Streaming
- AWS Identity and Access Management (IAM) -Serverzertifikate
- Dedizierte IP-Adressen
- Mindestprotokollversion SSLv3

Die folgenden Einstellungen können in einer Multi-Tenant-Distribution oder einem Distributionsmandanten nicht konfiguriert werden. Legen Sie die gewünschten Werte stattdessen in einer Verbindungsgruppe fest. Alle Distributionsmandanten, die dieser Verbindungsgruppe zugeordnet sind, verwenden diese Einstellungen. Weitere Informationen finden Sie unter Erstellen einer benutzerdefinierten Verbindungsgruppe (optional).

- Aktivieren von IPv6 (Viewer-Anforderungen)
- Statische Anycast-IP-Liste

Anpassungen von Distributionsmandanten

Wenn Sie eine Multi-Tenant-Distribution verwenden, übernehmen Ihre Distributionsmandanten die Konfiguration der Multi-Tenant-Distribution. Sie können jedoch einige Einstellungen auf der Ebene der Distributionsmandanten anpassen.

Sie können Folgendes anpassen:

- Parameter – Parameter sind Schlüssel-Wert-Paare, die Sie für die Ursprungsdomain oder die Quellpfade verwenden können. Siehe So funktionieren Parameter für Distributionsmandanten.
- AWS WAF Web-ACL (V2) — Sie können eine separate Web-ACL für den Verteilungsmandanten angeben, wodurch die für die Mehrmandantenverteilung verwendete Web-ACL außer Kraft gesetzt wird. Sie können diese Einstellung auch für einen bestimmten Distributionsmandanten deaktivieren, was zur Folge hat, dass der Distributionsmandant den Web-ACL-Schutz nicht von der Multi-Tenant-Distribution übernimmt. Weitere Informationen finden Sie unter AWS WAF Web-ACL.
- Geografische Einschränkungen – Geografische Einschränkungen, die Sie für einen Distributionsmandanten angeben, überschreiben alle geografischen Einschränkungen für die Multi-Tenant-Distribution. Wenn Sie beispielsweise Deutschland (DE) in Ihrer Multi-Tenant-Distribution

blockieren, blockieren alle zugehörigen Vertriebsmandanten auch DE. Wenn Sie DE jedoch für einen bestimmten Distributionsmandanten zulassen, überschreibt diese Einstellung für den Distributionsmandanten die Einstellungen für die Multi-Tenant-Distribution. Weitere Informationen finden Sie unter Einschränken der geografischen Distribution Ihrer Inhalte.

- Aufhebungspfade – Geben Sie die Dateipfade zu den Inhalten an, die Sie für den Distributionsmandanten als ungültig erklären möchten. Weitere Informationen finden Sie unter Aufheben der Gültigkeit von Dateien.
- Benutzerdefinierte TLS-Zertifikate — AWS Certificate Manager (ACM) -Zertifikate, die Sie für Verteilungsmandanten angeben, ergänzen das in der Mehrmandantenverteilung bereitgestellte Zertifikat. Wenn jedoch das Zertifikat der Multi-Tenant-Distribution und das Zertifikat des Distributionsmandanten dieselbe Domain abdecken, wird das Mandantenzertifikat verwendet. Weitere Informationen finden Sie unter Fordern Sie Zertifikate für Ihren CloudFront Distributionsmandanten an.
- Domainnamen – Sie müssen mindestens einen Domainnamen pro Distributionsmandant angeben.

So funktionieren Parameter für Distributionsmandanten

Ein Parameter ist ein Schlüssel-Wert-Paar, das Sie für Platzhalterwerte verwenden können. Definieren Sie die Parameter, die Sie in Ihrer Multi-Tenant-Distribution verwenden möchten, und geben Sie an, ob sie erforderlich sind.

Wenn Sie die Parameter in Ihrer Multi-Tenant-Distribution definieren, wählen Sie aus, ob diese Parameter auf Ebene der Distributionsmandanten eingegeben werden müssen.

- Wenn Sie die Parameter in der Multi-Tenant-Distribution als erforderlich definieren, müssen sie auf Ebene der Distributionsmandanten eingegeben werden. (Sie werden nicht übernommen.)
- Wenn die Parameter nicht erforderlich sind, können Sie in der Multi-Tenant-Distribution einen Standardwert angeben, der vom Distributionsmandanten übernommen wird.

Sie können Parameter für die folgenden Eigenschaften verwenden:

- Domänenname des Ursprungs
- Ursprungspfad

In der Multi-Tenant-Distribution können Sie bis zu zwei Parameter für jede der vorherigen Eigenschaften definieren.

Beispielparameter

Es folgen Beispiele zur Verwendung von Parametern für den Domainnamen und den Ursprungspfad.

Parameter für Domainnamen

In der Konfiguration für die Multi-Tenant-Distribution können Sie einen Parameter für den Ursprungsdomainnamen gemäß den folgenden Beispielen definieren:

Amazon S3

- {{*parameter1*}}.amzn-s3-demo-logging-bucket.s3.us-east-1.amazonaws.com
- {{*parameter1*}}-amzn-s3-demo-logging-bucket.s3.us-east-1.amazonaws.com

Benutzerdefinierte Ursprünge

- {{*parameter1*}}.lambda-url.us-east-1.on.aws
- {{*parameter1*}}.mediapackagev2.ap-south-1.amazonaws.com

Wenn Sie einen Distributionsmandanten erstellen, geben Sie den Wert an, der für *parameter1* verwendet werden soll.

```
"Parameters": [
  {
    "Name": "parameter1",
    "Value": "mycompany-website"
  }
]
```

Bei Verwendung der vorherigen Beispielangaben in der Multi-Tenant-Distribution wird der Ursprungsdomainname für den Distributionsmandanten wie folgt aufgelöst:

- *mycompany-website*.amzn-s3-demo-bucket3.s3.us-east-1.amazonaws.com
- *mycompany-website*-amzn-s3-demo-bucket3.s3.us-east-1.amazonaws.com
- *mycompany-website*.lambda-url.us-east-1.on.aws
- *mycompany-website*.mediapackagev2.ap-south-1.amazonaws.com

Parameter für den Ursprungspfad

In ähnlicher Weise können Sie die Parameter für den Ursprungspfad in der Multi-Tenant-Distribution gemäß den folgenden Beispielen definieren:

- /{{*parameter2*}}
- /{{*parameter2*}}/test
- /public/{{*parameter2*}}/test
- /search?name={{*parameter2*}}

Wenn Sie einen Distributionsmandanten erstellen, geben Sie den Wert an, der für *parameter2* verwendet werden soll.

```
"Parameters": [
  {
    "Name": "parameter2",
    "Value": "myBrand"
  }
]
```

Bei Verwendung der vorherigen Beispielangaben in der Multi-Tenant-Distribution wird der Ursprungspfad für den Distributionsmandanten wie folgt aufgelöst:

- /*myBrand*
- /*myBrand*/test
- /public/*myBrand*/test
- /search?name=*myBrand*

Example Beispiel

Sie möchten mehrere Websites (Mandanten) für Ihre Kunden erstellen und müssen sicherstellen, dass jede Verteilungsmandantenressource die richtigen Werte verwendet.

1. Sie erstellen eine Multi-Tenant-Distribution und geben zwei Parameter für die Konfiguration des Distributionsmandanten an.
2. Für den Namen der Ursprungsdomäne erstellen Sie einen Parameter mit dem Namen *customer-name* und geben an, dass er erforderlich ist. Sie geben den Parameter vor dem S3-Bucket ein, sodass er wie folgt angezeigt wird:

`{{customer-name}}.amzn-s3-demo-bucket3.s3.us-east-1.amazonaws.com`.

3. Für den ursprünglichen Pfad erstellen Sie einen zweiten Parameter mit dem Namen *my-theme* und geben an, dass er optional ist. Der Standardwert ist*basic*. Ihr Ursprungspfad wird wie folgt angezeigt: `/{{my-theme}}`
4. Wenn Sie einen Distributionsmandanten erstellen:
   - Für den Domänennamen müssen Sie einen Wert für angeben*customer-name*, da dieser in der Mehrmandantenverteilung als erforderlich gekennzeichnet ist.
   - Für den Quellpfad können Sie optional einen Wert für angeben *my-theme* oder den Standardwert verwenden.

Fordern Sie Zertifikate für Ihren CloudFront Distributionsmandanten an

Wenn Sie einen Verteilungsmandanten erstellen, erbt der Mandant das gemeinsame Zertifikat AWS Certificate Manager (ACM) von der Mehrmandantenverteilung. Dieses gemeinsame Zertifikat stellt HTTPS für alle Mandanten bereit, die der Multi-Tenant-Distribution zugeordnet sind.

Wenn Sie einen CloudFront Verteilungsmandanten erstellen oder aktualisieren, um Domänen hinzuzufügen, können Sie ein verwaltetes CloudFront Zertifikat von ACM hinzufügen. CloudFront erhält dann in Ihrem HTTP-validated Namen ein Zertifikat von ACM. Sie können dieses ACM-Zertifikat auf Mandantenebene für benutzerdefinierte Domänenkonfigurationen verwenden. CloudFront optimiert den Verlängerungsablauf, um Zertifikate auf dem neuesten Stand zu halten und die unterbrechungsfreie Bereitstellung von Inhalten zu gewährleisten.

> **Note**
>
> Sie besitzen das Zertifikat, aber es kann nur mit CloudFront Ressourcen verwendet werden und der private Schlüssel kann nicht exportiert werden.

Sie können das Zertifikat anfordern, wenn Sie den Distributionsmandanten erstellen oder aktualisieren.

Wählen Sie aus, wie die Token zur Zertifikatsvalidierung bereitgestellt werden sollen

Wenn Sie über die API oder AWS CLI ein verwaltetes Zertifikat für einen Verteilungsmandanten anfordern, geben Sie `ValidationTokenHost` in an`ManagedCertificateRequest`. Wählen Sie den Wert basierend darauf aus, wer das HTTP-Validierungstoken bereitstellt.

Für beide Optionen ist ein CloudFront-managed ACM-Zertifikat erforderlich. Der Parameter steuert nur, wo das Validierungstoken bereitgestellt wird.

CloudFront-hosted (`cloudfront`)

CloudFront stellt das HTTP-Validierungstoken automatisch bereit. Wählen Sie diese Option, wenn für die Domain kein Datenverkehr vorhanden ist oder wenn Sie ihr DNS CloudFront sofort darauf verweisen können.

Self-hosted (`self-hosted`)

Sie stellen das HTTP-Validierungstoken von Ihrer vorhandenen Infrastruktur aus bereit. Wählen Sie diese Option, wenn Sie eine Domain mit vorhandenem Traffic migrieren. Das Zertifikat kann ausgestellt werden, bevor Sie DNS darauf CloudFront verweisen. Sie können die Tokendatei auf Ihren vorhandenen Server hochladen oder eine (permanente) HTTP-301-Umleitung konfigurieren, die Validierungsanfragen an den ACM-Validierungsendpunkt weiterleitet.

Anweisungen zum Abschließen der Einrichtung des Domainbesitzes finden Sie unter. Abschließen der Domaineinrichtung Wenn Sie möchten`cloudfront`, wählen Sie Ich habe keinen Traffic. Wenn du dich dafür entschieden hast`self-hosted`, wähle Ich habe bestehenden Traffic aus.

Weitere Informationen zu den API-Werten finden Sie ManagedCertificateRequestin der CloudFront API-Referenz.

Themen

- Hinzufügen einer Domain und eines Zertifikats (Distributionsmandant)
- Abschließen der Domaineinrichtung
- Verweisen Sie Domains auf CloudFront
- Überlegungen zur Domain (Distributionsmandant)
- Platzhalter-Domains (Distributionsmandant)

Hinzufügen einer Domain und eines Zertifikats (Distributionsmandant)

Das folgende Verfahren veranschaulicht, wie Sie eine Domain hinzufügen und das Zertifikat für einen Distributionsmandanten aktualisieren.

So fügen Sie eine Domain und ein Zertifikat hinzu (Distributionsmandant)

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home.
2. Wählen Sie unter SaaS die Option Distributionsmandanten aus.
3. Suchen Sie nach dem Distributionsmandanten. Verwenden Sie das Dropdown-Menü in der Suchleiste, um nach Domain, Name, Distributions-ID, Zertifikat-ID, Verbindungsgruppen-ID oder Web-ACL-ID zu filtern.
4. Wählen Sie den Namen des Distributionsmandanten aus.
5. Wählen Sie für Domains die Option Domain verwalten aus.
6. Wählen Sie unter Zertifikat aus, ob ein benutzerdefiniertes TLS-Zertifikat für Ihren Distributionsmandanten verwendet werden soll. Das Zertifikat prüft, ob Sie berechtigt sind, den Domainnamen zu verwenden. Das Zertifikat muss in der Region USA Ost (Nord-Virginia) vorhanden sein.
7. Wählen Sie für Domains die Option Domain hinzufügen aus und geben Sie den Domainnamen ein. Abhängig von Ihrer Domain werden die folgenden Meldungen unter dem eingegebenen Domainnamen angezeigt.

   - Diese Domain ist durch das Zertifikat abgedeckt.
   - Diese Domain ist durch das Zertifikat abgedeckt, Validierung ausstehend.
   - Diese Domain ist nicht durch ein Zertifikat abgedeckt. (Das bedeutet, dass Sie die Inhaberschaft der Domain verifizieren müssen.)
8. Wählen Sie Distributionsmandant aktualisieren aus.

   Auf der Seite mit den Mandantendetails finden Sie unter Domains die folgenden Felder:

   - Domaininhaberschaft – der Status der Inhaberschaft der Domain. Bevor Inhalte CloudFront bereitgestellt werden können, muss Ihr Domaininhaber mithilfe der TLS-Zertifikatsvalidierung verifiziert werden.
   - DNS-Status — Die DNS-Einträge Ihrer Domain müssen auf diese verweisen, CloudFront um den Datenverkehr korrekt weiterzuleiten.
9. Wenn Ihre Domaininhaberschaft nicht bestätigt wurde, wählen Sie auf der Seite mit den Mandantendetails unter Domains die Option Domaineinrichtung abschließen aus und führen Sie dann das folgende Verfahren aus, um den DNS-Eintrag auf Ihren CloudFront Domainnamen zu verweisen.

Abschließen der Domaineinrichtung

Gehen Sie wie folgt vor, um zu überprüfen, ob Sie Inhaber der Domain für Ihre Distributionsmandanten sind. Wählen Sie je nach Domain eines der folgenden Verfahren aus.

Note

Wenn auf Ihre Domain bereits CloudFront mit einem Amazon Route 53-Aliaseintrag verwiesen wird, müssen Sie Ihren DNS-TXT-Eintrag mit `_cf-challenge.` vor dem Domainnamen hinzufügen. Dieser TXT-Eintrag bestätigt, dass Ihr Domainname mit verknüpft ist. CloudFront Wiederholen Sie diesen Schritt für jede Domain. Nachfolgend wird gezeigt, wie Sie Ihren TXT-Eintrag aktualisieren:

- Eintragsname: `_cf-challenge.`*`DomainName`*
- Eintragstyp: `TXT`
- Eintragswert: *`CloudFrontRoutingEndpoint`*

Ihr TXT-Eintrag könnte beispielsweise so aussehen: `_cf-challenge.example.com TXT d111111abcdef8.cloudfront.net`
Sie finden Ihren CloudFront Routing-Endpunkt in der Konsole auf der Detailseite des Distributionsmandanten oder verwenden Sie die ListConnectionGroupsAPI-Aktion in der Amazon CloudFront API-Referenz, um ihn zu finden.

Tip

Wenn Sie ein SaaS-Anbieter sind und die Ausstellung von Zertifikaten zulassen möchten, ohne dass Ihre Kunden (Mandanten) einen TXT-Eintrag direkt zu ihrem DNS hinzufügen müssen, gehen Sie wie folgt vor:

1. Wenn Sie Inhaber der Domain `example-saas-provider.com` sind, weisen Sie Ihren Mandanten Subdomains zu, z. B. `customer-123.example-saas-provider.com`.
2. Fügen Sie in Ihrem DNS den TXT-Eintrag `_cf-challenge.customer-123.example-saas-provider.com TXT d111111abcdef8.cloudfront.net` zu Ihrer DNS-Konfiguration hinzu.

3. Als Nächstes können Ihre Kunden (die Mandanten) ihren eigenen DNS-Eintrag aktualisieren, um ihren Domainnamen der von Ihnen angegebenen Subdomain zuzuordnen.

   ```
   www.customer-domain.com CNAME customer-123.example-saas-provider.com
   ```

I have existing traffic

Wählen Sie diese Option, wenn Ihre Domain keine Ausfallzeiten toleriert. Sie müssen Zugriff auf Ihren origin/web Server haben. Gehen Sie wie folgt vor, um die Domaininhaberschaft zu überprüfen.

So schließen Sie die Einrichtung der Domain bei vorhandenem Datenverkehr ab

1. Wählen Sie unter Spezifizieren Sie Ihren Web-Datenverkehr die Option Ich habe vorhandenen Datenverkehr und dann Weiter aus.
2. Wählen Sie für Domain-Inhaberschaft verifizieren eine der folgenden Optionen aus:

   - Vorhandenes Zertifikat verwenden – Suchen Sie nach einem vorhandenen ACM-Zertifikat oder geben Sie den Zertifikat-ARN ein, der die aufgelisteten Domains abdeckt.
   - Manueller Datei-Upload – Wählen Sie aus, ob Sie direkten Zugriff auf das Hochladen von Dateien auf Ihren Webserver haben.

     Erstellen Sie für jede Domain eine Klartextdatei, die Ihr Validierungstoken vom Token-Speicherort enthält, und laden Sie sie an Ihren Ursprung unter dem angegebenen Dateipfad auf Ihrem vorhandenen Server hoch. Der Pfad zu dieser Datei könnte wie folgt aussehen: `/.well-known/pki-validation/acm_9c2a7b2ec0524d09fa6013efb73ad123.txt`. Nachdem Sie diesen Schritt abgeschlossen haben, verifiziert ACM das Token und stellt dann das TLS-Zertifikat für die Domain aus.
   - HTTP-Weiterleitung – Wählen Sie aus, ob Sie keinen direkten Zugriff zum Hochladen von Dateien auf Ihren Webserver haben oder ob Sie einen CDN- oder Proxydienst verwenden.

     Erstellen Sie für jede Domain eine 301-Weiterleitung auf Ihrem vorhandenen Server. Kopieren Sie den bekannten Pfad unter Umleiten von und verweisen Sie

auf den angegebenen Zertifikatsendpunkt unter Umleiten zu. Ihre Umleitung könnte folgendermaßen aussehen:

```
If the URL matches: example.com/.well-known/pki-validation/
leabe938a4fe077b31e1ff62b781c123.txt
Then the settings are:Forwarding URL
Then 301 Permanent Redirect:To validation.us-east-1.acm-
validations.aws/123456789012/.well-known/pki-validation/
leabe938a4fe077b31e1ff62b781c123.txt
```

> **Note**
>
> Sie können Zertifikatsstatus überprüfen wählen, um zu überprüfen, wann ACM das Zertifikat für die Domain ausstellt.

3. Wählen Sie Weiter aus.
4. Führen Sie die Schritte unter Verweisen Sie Domains auf CloudFront aus.

I don't have traffic

Wählen Sie diese Option, wenn Sie neue Domains hinzufügen. CloudFront verwaltet die Zertifikatsvalidierung für Sie.

So schließen Sie die Einrichtung der Domain ohne vorhandenen Datenverkehr ab

1. Wählen Sie unter Spezifizieren Sie Ihren Web-Datenverkehr die Option Ich habe noch keinen Datenverkehr aus.
2. Führen Sie für jeden Domainnamen die Schritte unter Verweisen Sie Domains auf CloudFront aus.
3. Nachdem Sie Ihre DNS-Datensätze für jeden Domainnamen aktualisiert haben, wählen Sie Weiter aus.
4. Warten Sie, bis das Zertifikat ausgestellt wurde.

> **Note**
>
> Sie können Zertifikatsstatus überprüfen wählen, um zu überprüfen, wann ACM das Zertifikat für die Domain ausstellt.

5. Wählen Sie Absenden aus.

Verweisen Sie Domains auf CloudFront

Aktualisieren Sie Ihre DNS-Einträge, um den Verkehr von jeder Domain zum CloudFront Routing-Endpunkt weiterzuleiten. Sie können mehrere Domainnamen haben, aber sie müssen alle zu diesem Endpunkt aufgelöst werden.

Um Domains zu verweisen CloudFront

1. Kopieren Sie den Wert des CloudFront Routing-Endpunkts, z. B. d111111abcdef8.cloudfront.net.
2. Aktualisieren Sie Ihre DNS-Einträge, um den Verkehr von jeder Domain zum Routing-Endpunkt weiterzuleiten. CloudFront
   1. Melden Sie sich bei der Domainvergabestelle oder der Managementkonsole für Ihren DNS-Anbieter an.
   2. Navigieren Sie zum DNS-Verwaltungsbereich für Ihre Domain.
      - Für Subdomains – Erstellen Sie einen CNAME-Datensatz. Beispiel:
        - Name – Ihre Subdomain (z. B. `www` oder `app`)
        - Value//Target — Der CloudFront Routing-Endpunkt
        - Datensatztyp – CNAME
        - TTL – 3600 (bzw. was für Ihren Anwendungsfall geeignet ist)
      - Für apex/root Domänen — Dies erfordert eine eindeutige DNS-Konfiguration, da CNAME-Standardeinträge nicht auf Stamm- oder Apex-Domänenebene verwendet werden können. Da die meisten DNS-Anbieter ALIAS-Einträge nicht unterstützen, empfehlen wir, einen ALIAS-Eintrag in Route 53 zu erstellen. Beispiel:
        - Name – Ihre Apex-Domain (z. B. `example.com`)
        - Eintragstyp – A
        - Alias – Ja

- Alias-Ziel — Ihr Routing-Endpunkt CloudFront
- Routing-Richtlinie – Einfach (bzw. was für Ihren Anwendungsfall geeignet ist)

3. Stellen Sie sicher, dass die DNS-Änderung übernommen wurde. (Dies geschieht normalerweise, wenn die TTL abgelaufen ist. Manchmal kann es 24–48 Stunden dauern.) Verwenden Sie ein Tool wie `dig` oder `nslookup`.

```
dig www.example.com
# Should eventually return a CNAME pointing to your CloudFront routing endpoint
```

3. Kehren Sie zur CloudFront Konsole zurück und wählen Sie Submit. Wenn Ihre Domain aktiv ist, CloudFront aktualisiert sie den Domain-Status, um anzuzeigen, dass Ihre Domain bereit ist, Traffic bereitzustellen.

Weitere Informationen finden Sie in der Dokumentation zu Ihrem DNS-Anbieter.

- Cloudflare
- ClouDNS
- DNSimple
- Gandi.net
- GoDaddy
- Google Cloud DNS
- Nennen Sie cheap

Überlegungen zur Domain (Distributionsmandant)

Wenn eine Domain aktiv ist, wurde die Domainkontrolle eingerichtet und CloudFront beantwortet alle Anfragen von Zuschauern an diese Domain. Nach der Aktivierung kann eine Domain nicht deaktiviert oder in einen inaktiven Status geändert werden. Die Domain kann keiner anderen CloudFront Ressource zugeordnet werden, solange sie bereits verwendet wird. Um die Domain einer anderen Distribution zuzuordnen, verwenden Sie die UpdateDomainAssociationAnfrage, um die Domain von einer CloudFront Ressource auf eine andere zu verschieben.

Wenn eine Domain inaktiv ist, reagiert sie CloudFront nicht auf Zuschaueranfragen an die Domain. Beachten Sie Folgendes, wenn die Domain inaktiv ist:

- Wenn Sie eine ausstehende Zertifikatsanforderung haben, CloudFront beantwortet sie Anfragen für den bekannten Pfad. Solange die Anfrage aussteht, kann die Domain keinen anderen CloudFront Ressourcen zugeordnet werden.
- Wenn Sie keine ausstehende Zertifikatsanforderung haben, beantwortet sie CloudFront keine Anfragen für die Domain. Sie können die Domain mit anderen CloudFront Ressourcen verknüpfen.
- Pro Distributionsmandant ist nur eine ausstehende Zertifikatsanforderung erlaubt. Um ein weiteres Zertifikat für andere Domains anzufordern, müssen Sie die bestehende ausstehende Anforderung stornieren. Durch das Stornieren einer vorhandenen Zertifikatsanforderung wird das zugehörige ACM-Zertifikat nicht gelöscht. Sie können es mithilfe der ACM-API löschen.
- Wenn Sie ein neues Zertifikat auf Ihren Distributionsmandanten anwenden, wird die Zuordnung zum vorherigen Zertifikat aufgehoben. Sie können das Zertifikat wiederverwenden, um die Domain für einen anderen Distributionsmandanten abzudecken.

Wie bei Verlängerungen von DNS-validated Zertifikaten werden Sie benachrichtigt, wenn die Zertifikatserneuerung erfolgreich ist. Sie müssen jedoch nichts weiter tun. CloudFront verwaltet die Zertifikatsverlängerung für Ihre Domain automatisch.

> **Note**
>
> Sie müssen die ACM-API-Operationen nicht aufrufen, um Ihre Zertifikatsressourcen zu erstellen oder zu aktualisieren. Sie können Ihre Zertifikate verwalten, indem Sie die Operationen CreateDistributionTenantund UpdateDistributionTenantAPI verwenden, um die Details für Ihre verwaltete Zertifikatsanforderung anzugeben.

> **Note**
>
> Kompatibilität mit geografischen Einschränkungen — Die CloudFront verwaltete Zertifikatsvalidierung ist mit geografischen Einschränkungen kompatibel. Wenn Sie geografische Beschränkungen für eine Distribution aktivieren, die ein CloudFront verwaltetes Zertifikat verwendet, werden Validierungsanfragen von den Regeln für geografische Beschränkungen ausgeschlossen. Zertifizierungsstellen überprüfen den Domainbesitz aus der Perspektive mehrerer geografisch verteilter Netzwerke (Multi Perspective Issuance Corroboration), um sich vor BGP-Hijacking zu schützen. CloudFront stellt sicher, dass diese Überprüfungsanfragen erfolgreich sind, unabhängig davon, welche Länder Sie blockieren oder zulassen.

Für die Nutzung dieser Funktion sind keine Maßnahmen erforderlich. Wenn Sie zuvor geografische Beschränkungen aufgehoben haben, um Fehler bei der Zertifikatsvalidierung zu umgehen, können Sie diese problemlos erneut anwenden.

Platzhalter-Domains (Distributionsmandant)

Platzhalter-Domains werden in folgenden Situationen für Distributionsmandanten unterstützt:

- wenn der Platzhalter in dem gemeinsamen Zertifikat enthalten ist, das von der übergeordneten Multi-Tenant-Distribution übernommen wurde
- wenn Sie ein gültiges vorhandenes benutzerdefiniertes TLS-Zertifikat für Ihren Distributionsmandanten verwenden

Erstellen einer benutzerdefinierten Verbindungsgruppe (optional)

CloudFront Erstellt standardmäßig eine Verbindungsgruppe für Sie, wenn Sie eine Verteilung mit mehreren Mandanten erstellen. Die Verbindungsgruppe steuert, mit welchen Inhaltsanfragen von Zuschauern eine Verbindung hergestellt wird. CloudFront

Wir empfehlen Ihnen, die Standardverbindungsgruppe zu verwenden. Wenn Sie jedoch Unternehmensanwendungen isolieren oder Gruppen von Distributionsmandanten separat verwalten müssen, können Sie eine benutzerdefinierte Verbindungsgruppe erstellen. Beispielsweise müssen Sie möglicherweise einen Distributionsmandanten in eine separate Verbindungsgruppe verschieben, wenn er von einem DDoS-Angriff betroffen ist. Auf diese Weise können Sie andere Distributionsmandanten vor den Auswirkungen schützen.

Erstellen einer benutzerdefinierten Verbindungsgruppe (optional)

Optional können Sie eine benutzerdefinierte Verbindungsgruppe für Ihre Distributionsmandanten erstellen.

So erstellen Sie eine benutzerdefinierte Verbindungsgruppe (optional)

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home.
2. Wählen Sie im Navigationsbereich Settings (Einstellungen).
3. Aktivieren Sie die Einstellungen für die Verbindungsgruppe.

4. Wählen Sie im Navigationsbereich Verbindungsgruppen und dann Verbindungsgruppe erstellen aus.
5. Geben Sie unter Verbindungsgruppenname einen Namen für die Verbindungsgruppe ein. Sie können diesen Namen nicht mehr aktualisieren, nachdem Sie die Verbindungsgruppe erstellt haben.
6. Geben Sie für IPv6 an, ob Sie dieses IP-Protokoll aktivieren möchten. Weitere Informationen finden Sie unter Aktivieren von IPv6 (Viewer-Anforderungen).
7. Geben Sie für Statische Anycast-IP-Liste an, ob Sie Datenverkehr von einer Reihe von IP-Adressen aus an Ihre Distributionsmandanten weiterleiten möchten. Weitere Informationen finden Sie unter Statische Anycast-IP-Liste.
8. (Optional) Fügen Sie Ihrer Verbindungsgruppe Tags hinzu.
9. Wählen Sie Verbindungsgruppe erstellen aus.

Wenn Ihre Verbindungsgruppe erstellt wurde, werden die von Ihnen angegebenen Einstellungen sowie der ARN und der Endpunkt angezeigt.

- Der ARN sieht wie im folgenden Beispiel aus: `arn:aws:cloudfront::123456789012:connection-group/cg_2uVbA9KeWaADTbKzhj9lcKDoM25`
- Der Endpunkt sieht wie im folgenden Beispiel aus: d111111abcdef8.cloudfront.net

Sie können Ihre benutzerdefinierte Verbindungsgruppe löschen, nachdem Sie sie erstellt haben. Bevor Sie eine Verbindungsgruppe löschen können, müssen Sie jedoch zunächst alle zugehörigen Distributionsmandanten löschen. Sie können die Standardverbindungsgruppe, die Sie bei der CloudFront Erstellung Ihrer Mehrmandantenverteilung für Sie erstellt haben, nicht deaktivieren oder löschen.

> **Important**
>
> Wenn Sie die Verbindungsgruppe für einen Verteilungsmandanten ändern, CloudFront wird weiterhin Datenverkehr für den Verteilungsmandanten übertragen, allerdings mit erhöhter Latenz. Es wird empfohlen, den DNS-Eintrag für den Verteilungsmandanten so zu aktualisieren, dass er den CloudFront Routingendpunkt der neuen Verbindungsgruppe verwendet.

Solange Sie den DNS-Eintrag nicht aktualisieren, basiert CloudFront das Routing auf den Einstellungen, die für den Routing-Endpunkt definiert sind, auf den die Website derzeit mit DNS verweist. Angenommen, Ihre Standard-Verbindungsgruppe verwendet keine statischen Anycast-IPs, Ihre neue benutzerdefinierte Verbindungsgruppe jedoch schon. Sie müssen den DNS-Eintrag aktualisieren, bevor CloudFront Sie statische Anycast-IPs für die Verteilungsmandanten in Ihrer benutzerdefinierten Verbindungsgruppe verwenden können.

Migrieren zu einer Multi-Tenant-Distribution

Wenn Sie über eine CloudFront Standarddistribution verfügen und zu einer Multi-Tenant-Distribution migrieren möchten, gehen Sie wie folgt vor.

So migrieren Sie von einer Standarddistribution zu einer Multi-Tenant-Distribution

1. Überprüfen Sie das Nicht unterstützte Funktionen.
2. Erstellen Sie eine Multi-Tenant-Distribution mit derselben Konfiguration wie Ihre Standarddistribution, abgesehen von den nicht unterstützten Features. Weitere Informationen finden Sie unter Erstellen Sie eine CloudFront Distribution in der Konsole.
3. Erstellen Sie einen Distributionsmandanten und fügen Sie einen alternativen Domainnamen hinzu, dessen Inhaber Sie sind.

   Warning

   Verwenden Sie nicht den aktuellen Domainnamen, der Ihrer Standarddistribution zugeordnet ist. Fügen Sie stattdessen eine Platzhalter-Domain hinzu. Sie werden Ihre Domain später verschieben. Weitere Informationen zum Erstellen eines Distributionsmandanten finden Sie unter Erstellen Sie eine CloudFront Distribution in der Konsole.

4. Stellen Sie ein vorhandenes Zertifikat für die Domain des Distributionsmandanten bereit. Dies ist das Zertifikat, das die Platzhalterdomain und die Domain, die Sie verschieben möchten, abdeckt.
5. Kopieren Sie den CloudFront Routing-Endpunkt von der Detailseite des Verteilungsmandanten in der Konsole. Sie können es auch mithilfe der ListConnectionGroupsAPI-Aktion in der Amazon CloudFront API-Referenz finden.

6. Um die Inhaberschaft der Domain zu verifizieren, erstellen Sie einen DCV-TXT-Eintrag mit einem Unterstrich (_), der auf den CloudFront Routing-Endpunkt für Ihren Distributionsmandanten verweist. Weitere Informationen finden Sie unter Verweisen Sie Domains auf CloudFront.
7. Wenn Ihre Änderungen übernommen wurden, aktualisieren Sie Ihren Distributionsmandanten so, dass er die Domain verwendet, die Sie zuvor für Ihre Standarddistribution verwendet haben.

   - Konsole – Detaillierte Anleitungen finden Sie unter Hinzufügen einer Domain und eines Zertifikats (Distributionsmandant).
   - API — Verwenden Sie die UpdateDomainAssociationAPI-Aktion in der Amazon CloudFront API-Referenz.

   > **Important**
   >
   > Dadurch wird der Cache-Schlüssel für Ihre Inhalte zurückgesetzt. Danach CloudFront beginnt das Zwischenspeichern Ihrer Inhalte mithilfe des neuen Cache-Schlüssels. Weitere Informationen finden Sie unter Grundlegendes zum Cache-Schlüssel.

8. Aktualisieren Sie Ihren DNS-Eintrag so, dass Ihre Domain auf den CloudFront Routing-Endpunkt für Ihren Verteilungsmandanten verweist. Nachdem Sie diesen Schritt abgeschlossen haben, ist Ihre Domain bereit, den Datenverkehr an Ihren Distributionsmandanten weiterzuleiten. Weitere Informationen finden Sie unter Verweisen Sie Domains auf CloudFront.
9. (Optional) Nachdem Sie Ihre Domain erfolgreich zu einem Verteilungsmandanten migriert haben, können Sie ein anderes CloudFront verwaltetes Zertifikat verwenden, das den Domainnamen für Ihren Verteilungsmandanten abdeckt. Um ein verwaltetes Zertifikat anzufordern, erstellen Sie einen separaten TXT-Eintrag, um das Zertifikat auszustellen, und folgen Sie den Schritten unter Abschließen der Domaineinrichtung.

Eine Distribution erstellen

In diesem Thema wird erklärt, wie Sie mit der CloudFront Konsole eine Distribution erstellen.

-Übersicht

1. Erstellen Sie mindestens einen Amazon-S3-Bucket oder konfigurieren Sie HTTP-Server als Ursprungs-Server. Ein Ursprung ist der Speicherort, an dem Sie die Originalversion Ihrer Inhalte speichern. Wenn CloudFront Sie eine Anfrage für Ihre Dateien erhalten, geht sie an

den Ursprung, um die Dateien abzurufen, die sie an Edge-Standorten verteilt. Sie können jede Kombination von Amazon S3-Buckets und HTTP-Servern als Ursprungsserver verwenden.

- Wenn Sie Amazon S3 verwenden, darf der Name Ihres Buckets nur Kleinbuchstaben enthalten und er darf keine Leerzeichen enthalten.
- Wenn Sie einen Amazon-EC2-Server oder einen anderen benutzerdefinierten Ursprung verwenden, lesen Sie Verwenden von Amazon EC2 (oder eines anderen benutzerdefinierten Ursprungs).
- Informationen zur aktuell gültigen maximalen Anzahl von Ursprüngen, die Sie für eine Verteilung erstellen können, oder zum Anfordern eines höheren Kontingents finden Sie unter Allgemeine Kontingente für Verteilungen.

2. Laden Sie Ihre Inhalte auf Ihre Ursprungsserver hoch. Sie machen Ihre Objekte öffentlich lesbar, oder Sie können CloudFront signierte URLs verwenden, um den Zugriff auf Ihre Inhalte einzuschränken.

   > **Important**
   >
   > Sie sind dafür verantwortlich, die Sicherheit Ihres Ursprungsservers zu gewährleisten. Sie müssen sicherstellen, CloudFront dass dieser Zugriff auf den Server berechtigt ist und dass die Sicherheitseinstellungen Ihre Inhalte schützen.

3. Erstellen Sie Ihre CloudFront Distribution:

   - Ein detailliertes Verfahren zum Erstellen einer Distribution in der CloudFront Konsole finden Sie unterErstellen Sie eine CloudFront Distribution in der Konsole.
   - Informationen zum Erstellen einer Distribution mithilfe der CloudFront API finden Sie CreateDistributionin der Amazon CloudFront API-Referenz.

4. (Optional) Wenn Sie die CloudFront Konsole verwenden, um Ihre Distribution zu erstellen, erstellen Sie weitere Cache-Verhaltensweisen oder -Ursprünge für die Verteilung. Weitere Informationen zu Verhaltensweisen und Ursprüngen finden Sie unter So aktualisieren Sie eine Multi-Tenant-Distribution.
5. Testen Sie Ihre Verteilung. Weitere Informationen zum Testen finden Sie unter Testen einer Distribution.
6. Entwickeln Sie Ihre Website oder Anwendung so, dass sie mithilfe des von CloudFront nach der Erstellung Ihrer Verteilung in Schritt 3 zurückgegebenen Domänennamens auf Ihre Inhalte zugreift. Wenn beispielsweise d111111abcdef8.cloudfront.net als Domainnamen für Ihre

Distribution CloudFront zurückgegeben wird, lautet die URL für die Datei `image.jpg` in einem Amazon S3 S3-Bucket oder im Stammverzeichnis auf einem HTTP-Server. `https://d111111abcdef8.cloudfront.net/image.jpg`

Wenn Sie bei der Erstellung Ihrer Verteilung einen oder mehrere alternative Domänennamen (CNAMEs) angegeben haben, können Sie Ihren eigenen Domänennamen verwenden. In diesem Fall könnte die URL für `image.jpg` folgendermaßen lauten: `https://www.example.com/image.jpg`.

Beachten Sie Folgendes:

- Wenn Sie signierte URLs verwenden möchten, um den Zugriff auf Ihre Inhalte einzuschränken, informieren Sie sich unter Bereitstellen privater Inhalte mit signierten URLs und signierten Cookies.
- Wenn Sie komprimierte Inhalte bereitstellen möchten, informieren Sie sich unter Bereitstellen von komprimierten Dateien.
- Informationen zum CloudFront Anfrage- und Antwortverhalten für Amazon S3 und benutzerdefinierte Ursprünge finden Sie unterVerhalten von Anfragen und Antworten.

Themen

- Erstellen Sie eine CloudFront Distribution in der Konsole
- Werte, die in der Konsole CloudFront angezeigt werden
- Weitere Links
- Fügen Sie Ihrer CloudFront Standarddistribution eine Domain hinzu

Erstellen Sie eine CloudFront Distribution in der Konsole

Wenn Sie eine Verteilung erstellen, CloudFront konfiguriert es Ihre Verteilungseinstellungen für Sie, die auf Ihrem Inhaltsherkunftstyp basieren. Weitere Informationen zu vorkonfigurierten Einstellungen finden Sie unter Referenz für vorkonfigurierte Distributionseinstellungen.
Sie können auch Multi-Tenant-Distributionen mit Einstellungen erstellen, die für mehrere Distributionsmandanten wiederverwendet werden können. Weitere Informationen finden Sie unter Verständnis der Funktionsweise von Multi-Tenant-Distributionen. Alternativ können Sie Ihre eigenen Distributionseinstellungen manuell konfigurieren.

Multi-tenant

So erstellen Sie eine Multi-Tenant-Distribution

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home.
2. Wählen Sie im Navigationsbereich Distributionen und dann Distribution erstellen aus.
3. Wählen Sie Multi-tenant Architektur, Weiter.
4. Geben Sie einen Distributionsnamen für die Multi-Tenant-Distribution ein. Der Name wird als Wert für den `Name`-Schlüssel angezeigt. Sie können diesen Wert später ändern. Sie können bis zu 50 Tags für Ihre Multi-Tenant-Distribution hinzufügen. Weitere Informationen finden Sie unter Kennzeichnen Sie eine Ressource CloudFront.
5. (Optional) Wählen Sie für das Wildcard-Zertifikat das AWS Certificate Manager (ACM) - Zertifikat aus, das alle Subdomains unter der Stammdomain abdeckt, z. B. `*.example.com` Das Zertifikat muss in der Region USA Ost (Nord-Virginia) vorhanden sein.
6. Wählen Sie Weiter aus.
7. Wählen Sie auf der Seite „Herkunft angeben“ den Herkunftstyp aus, von dem Ihre CloudFront Inhalte abgerufen werden sollen. CloudFront verwendet die empfohlenen Einstellungen für diesen Herkunftstyp für Ihre Multi-Tenant-Distribution. Weitere Informationen zu den empfohlenen Einstellungen finden Sie unter Referenz für vorkonfigurierte Distributionseinstellungen.
8. Wählen Sie für Ursprung unter dem von Ihnen ausgewählten Ursprungstyp den zu verwendenden Ursprung aus oder geben Sie ihn ein.
9. Geben Sie als Ursprungspfad den Schrägstrich (`/`) gefolgt vom Ursprungspfad ein.
10. (Optional) Um einen Parameter hinzuzufügen, wählen Sie entweder für den Namen der Ursprungsdomain oder den Ursprungspfad die Option Parameter einfügen aus. Sie können bis zu zwei Parameter für jedes Feld eingeben.

    a. Wählen Sie Neuen Parameter erstellen aus.

    b. Geben Sie im Dialogfeld Neuen Parameter erstellen unter Parametername einen eindeutigen Namen für den Parameter und optional eine Beschreibung ein.

    c. Aktivieren Sie für Erforderlicher Parameter das Kontrollkästchen, damit dieser Parameterwert auf der Ebene des Distributionsmandanten erforderlich ist. Wenn dieser Wert nicht erforderlich ist, geben Sie einen Standardwert ein, den der Distributionsmandant übernimmt.

d. Wählen Sie Parameter erstellen aus. Dieser Parameter wird im entsprechenden Feld angezeigt.

11. Wählen Sie unter Optionen eine der folgenden Optionen aus:

    - Empfohlene Ursprungseinstellungen verwenden – Verwenden Sie die empfohlenen Cache- und Ursprungs-Standardeinstellungen für den ausgewählten Ursprungstyp.
    - Ursprungseinstellungen anpassen – Passen Sie die Cache- und Ursprungseinstellungen an. Wenn Sie diese Option wählen, geben Sie Ihre eigenen Werte an, die angezeigt werden.

12. Wählen Sie Weiter aus.
13. Wählen Sie auf der Seite Sicherheitsvorkehrungen aktivieren aus, ob der AWS WAF Sicherheitsschutz aktiviert werden soll. Sie können die Web-ACL später für bestimmte Distributionsmandanten anpassen. Weitere Informationen finden Sie unter Aktivieren AWS WAF für eine neue Distribution.
14. Wählen Sie Weiter und dann Distribution erstellen aus.
15. Auf der Seite Distributionen wird Ihre Multi-Tenant-Distribution in der Ressourcenliste angezeigt. Sie können das Dropdown-Menü Alle Distributionen auswählen, um nach Standarddistribution oder Multi-Tenant-Distribution zu filtern. Sie können auch die Spalte Typ auswählen, um nach Standard- oder Multi-Tenant-Distribution zu filtern.

CloudFront Erstellt standardmäßig eine Verbindungsgruppe für Sie. Die Verbindungsgruppe steuert, mit welcher Verbindung Zuschaueranfragen nach Inhalten verbunden CloudFront werden. Sie können einige Routing-Einstellungen in der Verbindungsgruppe anpassen. Weitere Informationen finden Sie unter Verständnis der Funktionsweise von Multi-Tenant-Distributionen.

Sie können zusätzliche Distributionsmandanten erstellen, indem Sie die Multi-Tenant-Distribution als Vorlage verwenden.

So erstellen Sie einen Distributionsmandanten

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home.
2. Führen Sie im Navigationsbereich eine der folgenden Aktionen aus:
    - Wählen Sie Distributionen und eine Multi-Tenant-Distribution aus und klicken Sie dann auf Mandant erstellen.
    - Wählen Sie Distributionsmandanten und anschließend Mandant erstellen aus.

3. Geben Sie unter Name des Distributionsmandanten den Namen ein. Der Name muss einzigartig sein AWS-Konto und kann nach der Erstellung nicht mehr geändert werden.
4. Wählen Sie für die Vorlagendistribution die ID einer Multi-Tenant-Distribution aus der Liste aus.
5. Fügen Sie unter Tags verwalten bis zu 50 Schlüssel-Wert-Paare für den Distributionsmandanten hinzu. Weitere Informationen finden Sie unter Kennzeichnen Sie eine Ressource CloudFront.
6. Wählen Sie Weiter aus.
7. Wählen Sie auf der Seite Domains hinzufügen unter Zertifikat aus, ob Sie ein benutzerdefiniertes TLS-Zertifikat für Ihren Distributionsmandanten wünschen. Das Zertifikat prüft, ob Sie berechtigt sind, den Domainnamen zu verwenden. Das Zertifikat muss in der Region USA Ost (Nord-Virginia) vorhanden sein.
8. Geben Sie bei Domains den Ihren Domainnamen ein.

> **Note**
>
> Wenn Sie einen Domainnamen eingegeben haben, der nicht durch ein Zertifikat abgedeckt ist, müssen Sie verifizieren, dass Sie Inhaber der Domain sind. Sie können jedoch erst den Distributionsmandanten erstellen und die Domaininhaberschaft später verifizieren. Weitere Informationen finden Sie unter Fordern Sie Zertifikate für Ihren CloudFront Distributionsmandanten an.

9. Wählen Sie Weiter aus.
10. Auf der Seite Parameter definieren werden die Parameter angezeigt, die Sie in der Multi-Tenant-Distribution angegeben haben. Geben Sie für die erforderlichen Parameter einen Wert neben dem Parameternamen ein und speichern Sie Ihre Änderungen.
11. Um einen weiteren Parameter hinzuzufügen, wählen Sie Parameter hinzufügen und geben Sie einen Namen und einen Wert ein.
12. Wählen Sie Weiter aus.
13. (Optional) Wenn Sie sich unter Sicherheitsanpassungen für Distributionseinstellungen überschreiben entscheiden, wählen Sie die für Ihren Anwendungsfall geeignete Option aus.
14. (Optional) Wenn Sie sich unter Anpassung geografischer Einschränkungen für Distributionseinstellungen überschreiben entscheiden, wählen Sie für Einschränkungstyp und Länder die für Ihren Anwendungsfall geeignete Option aus. Weitere Informationen finden Sie unter Einschränken der geografischen Distribution Ihrer Inhalte.

15. Wählen Sie Weiter aus.
16. Wählen Sie Distributionsmandant erstellen aus.

Sie finden alle Ihre Distributionsmandanten auf der Seite Distributionsmandanten. Sie können nach Folgendem filtern:

Zuordnung

- Verteilungs-ID
- Zertifikat-ID
- Verbindungsgruppen-ID
- Web-ACL-ID

Eigenschaften

- Name
- Domain

Sie können Ihre Distributionsmandanten bearbeiten, um bestimmte Einstellungen anzupassen. Weitere Informationen finden Sie unter Anpassungen von Distributionsmandanten.

Standard

So erstellen Sie eine Standarddistribution

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home.
2. Wählen Sie im Navigationsbereich Distributionen und dann Distribution erstellen aus.
3. Geben Sie einen Distributionsnamen für die Standarddistribution ein. Der Name wird als Wert für den `Name`-Schlüssel als Tag angezeigt. Sie können diesen Wert später ändern. Sie können bis zu 50 Tags für Ihre Standarddistribution hinzufügen. Weitere Informationen finden Sie unter Kennzeichnen Sie eine Ressource CloudFront.
4. Wählen Sie Einzelne Website oder App und dann Weiter aus.
5. (Optional) Geben Sie für das Domain-Setup eine Domain ein, die bereits bei Route 53 registriert ist AWS-Konto, in Ihre oder registrieren Sie eine neue Domain. Führen Sie die Einrichtungsschritte aus.

- Wenn Ihre Domain einen anderen DNS-Anbieter als Route 53 verwendet, können Sie die Domain trotzdem hinzufügen, müssen dies jedoch tun, nachdem Sie die Distribution erstellt haben. Überspringen Sie vorerst die Domaineinrichtung, um mit der Erstellung der Distribution fortzufahren. Sie müssen die Domain und das TLS-Zertifikat später manuell konfigurieren. Weitere Informationen finden Sie unter Fügen Sie Ihrer CloudFront Standarddistribution eine Domain hinzu.

6. Wählen Sie Weiter aus.
7. Wählen Sie auf der Seite „Herkunft angeben“ den Herkunftstyp aus, von dem Ihre Inhalte abgerufen CloudFront werden sollen. CloudFront verwendet die empfohlenen Einstellungen für diesen Herkunftstyp für Ihre Distribution. Weitere Informationen zu den empfohlenen Einstellungen finden Sie unter Referenz für vorkonfigurierte Distributionseinstellungen.
8. Wählen Sie unter Ursprung Ihren Ursprung aus oder geben Sie ihn ein.
9. Wählen Sie unter Einstellungen eine der folgenden Optionen aus:

   - Empfohlene Ursprungseinstellungen verwenden – Verwenden Sie die empfohlenen Cache- und Ursprungs-Standardeinstellungen für den ausgewählten Ursprungstyp.
   - Ursprungseinstellungen anpassen – Passen Sie die Cache- und Ursprungseinstellungen an. Wenn Sie diese Option wählen, geben Sie Ihre eigenen Werte an.

10. Wählen Sie Weiter aus.
11. Wählen Sie auf der Seite Sicherheitsvorkehrungen aktivieren aus, ob der AWS WAF Sicherheitsschutz aktiviert werden soll.
12. Wählen Sie Weiter aus.
13. (Optional) Wenn Sie Route 53 für Ihre Domain verwenden, wird die Seite mit dem TLS-Zertifikat angezeigt. Wenn CloudFront Sie in Ihrem kein vorhandenes AWS Certificate Manager (ACM-) Zertifikat für Ihre Domain finden können, können Sie wählen `us-east-1` AWS-Region, ob Sie ein Zertifikat automatisch oder manuell erstellen möchten. AWS-Konto Nachdem das Zertifikat erstellt wurde, wählen Sie Weiter aus.
14. Überprüfen Sie Ihre Distributionsdetails und wählen Sie Distribution erstellen aus.
15. Nachdem Sie Ihre Verteilung CloudFront erstellt haben, ändert sich der Wert der Spalte Status für Ihre Verteilung von Bereitstellen auf Datum und Uhrzeit der Bereitstellung der Verteilung.

Der Domainname, der Ihrer Distribution CloudFront zugewiesen wurde, wird in der Liste der Distributionen angezeigt. (Der Status wird auch auf der Registerkarte General für die ausgewählte Verteilung angezeigt.)

> **Tip**
>
> Sie können anstelle des Namens, der Ihnen von zugewiesen wurde, einen alternativen Domainnamen verwenden CloudFront, indem Sie die Schritte unter befolgen. Verwenden von benutzerdefinierten URLs durch Hinzufügen von alternativen Domainnamen (CNAMEs)

16. Stellen Sie nach der Bereitstellung Ihrer Distribution sicher, dass Sie mit Ihrer neuen CloudFront URL (d111111abcdef8.cloudfront.net) oder dem CNAME auf Ihre Inhalte zugreifen können. Weitere Informationen finden Sie unter Testen einer Distribution.
17. Achten Sie darauf, Ihre DNS-Einträge so zu aktualisieren, dass sie darauf hinweisen, wann Sie bereit sind, Traffic an Ihre Distribution zu senden CloudFront . Weitere Informationen finden Sie unter Verweisen Sie Domains auf CloudFront (Standardverteilung).

Werte, die in der Konsole CloudFront angezeigt werden

Wenn Sie eine neue Distribution erstellen oder eine bestehende Distribution aktualisieren, CloudFront werden die folgenden Informationen in der CloudFront Konsole angezeigt.

> **Note**
>
> Aktive vertrauenswürdige Unterzeichner, AWS-Konten die über ein aktives CloudFront key pair verfügen und zur Erstellung gültiger signierter URLs verwendet werden können, sind derzeit in der CloudFront Konsole nicht sichtbar.

Verteilungs-ID

Wenn Sie mithilfe der CloudFront API eine Aktion für eine Distribution ausführen, verwenden Sie die Verteilungs-ID, um anzugeben, welche Distribution verwendet werden soll, z. B. `EDFDVBD6EXAMPLE` Sie können die Verteilungs-ID einer Verteilung nicht ändern.

Bereitstellung und Status

Wenn Sie eine Distribution bereitstellen, wird der Status Wird bereitgestellt in der Spalte Letzte Änderung angezeigt. Warten Sie, bis die Bereitstellung der Distribution abgeschlossen ist, und stellen Sie sicher, dass in der Spalte Status Aktiviert angezeigt wird. Weitere Informationen finden Sie unter Status der Verteilung.

Letzte Änderung

Datum und Uhrzeit der letzten Änderung der Verteilung im ISO 8601-Format, z. B. T19:37:58Z 2012-05-19. Weitere Informationen finden Sie unter https://www.w3.org/TR/NOTE-datetime.

Domainname

Sie verwenden den Domänennamen der Verteilung in den Links zu Ihren Objekten. Wenn beispielsweise der Domainname der Distribution `d111111abcdef8.cloudfront.net` ist, würde der Link zu `/images/image.jpg` dann `https://d111111abcdef8.cloudfront.net/images/image.jpg` lauten. Sie können den CloudFront -Domänennamen für Ihre Verteilung nicht ändern. Weitere Informationen zu CloudFront URLs für Links zu Ihren Objekten finden Sie unter. Passen Sie das URL-Format für Dateien in an CloudFront

Wenn Sie einen oder mehrere alternative Domainnamen (CNAMes) angegeben haben, können Sie anstelle des Domainnamens Ihre eigenen Domainnamen für Links zu Ihren Objekten verwenden. CloudFront Weitere Informationen zu CNAMEs finden Sie unter Alternative Domänennamen (CNAMEs).

> **Note**
>
> CloudFront Domainnamen sind einzigartig. Der Domänenname Ihrer Verteilung ist zuvor niemals für eine Verteilung verwendet worden und wird zukünftig für keine andere Verteilung verwendet werden.

Weitere Links

Weitere Informationen zum Erstellen einer Distribution finden Sie unter den folgenden Links.

- Informationen zum Erstellen einer Distribution, die einen Amazon Simple Storage Service (Amazon S3)-Bucket-Ursprung mit Ursprungszugriffssteuerung (OAC) verwendet, finden Sie unter Beginnen Sie mit einer CloudFront Standarddistribution.

- Informationen zur Verwendung der CloudFront APIs zum Erstellen einer Distribution finden Sie CreateDistributionin der Amazon CloudFront API-Referenz.
- Informationen zur Aktualisierung einer Distribution (z. B. um Cacheverhalten zu Standarddistributionen hinzuzufügen oder Distributionsmandanten anzupassen) finden Sie unter Eine Verteilung aktualisieren.
- Informationen zu den aktuellen Höchstwerten für die Anzahl der Verteilungen, die Sie für jedes AWS -Konto erstellen können, oder zum Anfordern eines höheren Kontingents (früher als Limit bezeichnet) finden Sie unter Allgemeine Kontingente für Verteilungen.

Fügen Sie Ihrer CloudFront Standarddistribution eine Domain hinzu

Nachdem Sie eine neue CloudFront Standarddistribution erstellt haben, können Sie ihr eine Domain hinzufügen. Optional können Sie bei der Erstellung Ihrer Standarddistribution eine Domain mit Amazon Route 53 einrichten. Weitere Informationen finden Sie unter Erstellen Sie eine CloudFront Distribution in der Konsole.

Hinzufügen einer Domain zu Ihrer bestehenden Standarddistribution

So fügen Sie Ihrer Standarddistribution eine Domain hinzu

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home.
2. Wählen Sie im Navigationsbereich Distributionen und dann die Distributions-ID aus.
3. Wählen Sie unter Einstellungen, Alternative Domainnamen die Option Domain hinzufügen aus.
4. Geben Sie bis zu fünf zu bedienende Domains an.
5. Wählen Sie Weiter aus.
6. Wenn CloudFront Sie für ein TLS-Zertifikat kein vorhandenes AWS Certificate Manager (ACM) Zertifikat für Ihre Domain AWS-Konto in Ihrem finden `us-east-1` AWS-Region, können Sie eines erstellen.

   - Wenn Sie Amazon Route 53 (Route 53) verwenden, CloudFront wird automatisch ein Zertifikat für Sie erstellt.
7. Wenn Ihr Zertifikat bereitgestellt wird, müssen Sie Ihre DNS-Datensätze bei Ihrem DNS-Anbieter aktualisieren, um die Domaininhaberschaft nachzuweisen. Wählen Sie anschließend Zertifikat validieren aus. Weitere Informationen finden Sie unter Verweisen Sie Domains auf CloudFront (Standardverteilung).

- Wenn Sie Route 53 verwenden, CloudFront aktualisiert Ihre DNS-Einträge für Sie.

8. Wählen Sie Weiter aus.
9. Überprüfen Sie Ihre Änderungen und wählen Sie Domains hinzufügen aus.
10. Bevor Sie Traffic an Ihre Distribution senden, stellen Sie sicher, dass Sie Ihre DNS-Einträge so aktualisieren, dass sie darauf verweisen CloudFront. Weitere Informationen erhalten Sie, wenn Sie auf der Seite mit den Vertriebsdetails CloudFront im Abschnitt Einstellungen die Option Domains weiterleiten an auswählen.

    - Wenn Sie Route 53 verwenden, können Sie das DNS-Routing automatisch für Sie CloudFront einrichten lassen.

Verweisen Sie Domains auf CloudFront (Standardverteilung)

Aktualisieren Sie Ihre DNS-Einträge, um den Verkehr von jeder Domain zum CloudFront Hostnamen weiterzuleiten. Sie können mehrere Domainnamen haben, aber sie müssen alle zu diesem Hostnamen aufgelöst werden.

Um Domains zu verweisen CloudFront

1. Kopieren Sie den CloudFront Hostnamenwert, z. B. d111111abcdef8.cloudfront.net.
2. Aktualisieren Sie Ihre DNS-Einträge, um den Verkehr von jeder Domain zum Hostnamen weiterzuleiten. CloudFront

    1. Melden Sie sich bei der Domainvergabestelle oder der Managementkonsole für Ihren DNS-Anbieter an.
    2. Navigieren Sie zum DNS-Verwaltungsbereich für Ihre Domain.
        - Für Subdomains – Erstellen Sie einen CNAME-Datensatz. Beispiel:
            - Name – Ihre Subdomain (z. B. `www` oder `app`)
            - Value//Target — Ihr Hostname CloudFront
            - Datensatztyp – CNAME
            - TTL – 3600 (bzw. was für Ihren Anwendungsfall geeignet ist)
        - Für apex/root Domains — Dies erfordert eine eindeutige DNS-Konfiguration, da Standard-CNAME-Einträge nicht auf Root- oder Apex-Domainebene verwendet werden können. Da die meisten DNS-Anbieter ALIAS-Einträge nicht unterstützen, empfehlen wir, einen ALIAS-Eintrag in Route 53 zu erstellen. Beispiel:

- Name – Ihre Apex-Domain (z. B. `example.com`)
- Eintragstyp – A
- Alias – Ja
- Alias-Ziel — Ihr Hostname CloudFront
- Routing-Richtlinie – Einfach (bzw. was für Ihren Anwendungsfall geeignet ist)

3. Stellen Sie sicher, dass die DNS-Änderung übernommen wurde. (Dies geschieht normalerweise, wenn die TTL abgelaufen ist. Manchmal kann es 24–48 Stunden dauern.) Verwenden Sie ein Tool wie `dig` oder `nslookup`.

```
dig www.example.com
# Should eventually return a CNAME pointing to your CloudFront hostname
```

3. Kehren Sie zur CloudFront Konsole zurück und wählen Sie Submit. Wenn Ihre Domain aktiv ist, CloudFront aktualisiert sie den Domain-Status, um anzuzeigen, dass Ihre Domain bereit ist, Traffic bereitzustellen.

Weitere Informationen finden Sie in der Dokumentation zu Ihrem DNS-Anbieter.

- Cloudflare
- ClouDNS
- DNSimple
- Gandi.net
- GoDaddy
- Google Cloud DNS
- Nennen Sie cheap

Referenz für vorkonfigurierte Distributionseinstellungen

Wenn Sie Ihre CloudFront Verteilung erstellen, CloudFront werden die meisten Vertriebseinstellungen automatisch für Sie konfiguriert, basierend auf Ihrem Inhaltstyp. Optional können Sie Ihre Distributionseinstellungen manuell bearbeiten. Weitere Informationen finden Sie unter Referenz für alle Distributionseinstellungen.

In den folgenden Abschnitten werden die standardmäßigen Vorkonfigurationseinstellungen für Distributionen beschrieben und die Einstellungen, die Sie anpassen können.

Amazon-S3-Ursprung

Im Folgenden finden Sie die Origin-Einstellungen, die für Ihren Amazon S3 S3-Ursprung in einer Multi-Tenant-Distribution CloudFront vorkonfiguriert sind.

Ursprungseinstellungen (vorkonfiguriert)

- Origin Access Control (nur Konsole) — CloudFront richtet das für Sie ein. CloudFront versucht, die S3-Bucket-Richtlinie für Standardverteilungen und für Multi-Tenant-Distributionen ohne Verwendung von Parametern in der Ursprungsdomäne hinzuzufügen.
- Benutzerdefinierten Header hinzufügen – Keinen
- Origin Shield aktivieren – Nein
- Verbindungsversuche – 3

Im Folgenden finden Sie die Cache-Einstellungen, die für Ihren Amazon S3 S3-Ursprung in einer Multi-Tenant-Distribution CloudFront vorkonfiguriert sind.

Cache-Einstellungen (vorkonfiguriert)

- Objekte automatisch komprimieren – Ja
- Viewer-Protokollrichtlinie – Zu HTTPS weiterleiten
- Zulässige HTTP-Methode – `GET, HEAD`
- Viewer-Zugriff einschränken – Nein
- Cache-Richtlinie – `CachingOptimized`
- Ursprungsanforderungsrichtlinie – Keine
- Antwortheader-Richtlinie – keine
- Smooth Streaming – Nein
- Verschlüsselung auf Feldebene – Nein
- Zugriffsprotokolle in Echtzeit aktivieren — Nein
- Funktionen – Nein

Im Folgenden finden Sie die Einstellungen, die Sie für Ihren Amazon-S3-Ursprung in einer Multi-Tenant-Distribution anpassen können.

Anpassbare Einstellungen

- S3-Zugriff — CloudFront legt dies für Sie fest, basierend auf Ihren S3-Bucket-Einstellungen:
  - Wenn Ihr Bucket öffentlich ist – Es ist keine Richtlinie für die Ursprungszugriffssteuerung (OAC) erforderlich.
  - Wenn Ihr Bucket privat ist – Sie können eine zu verwendende OAC-Richtlinie auswählen oder erstellen.
- Origin Shield aktivieren – Nein
- Objekte automatisch komprimieren – Ja
  - Wenn Sie Ja wählen, wird die `CachingOptimized`-Caching-Richtlinie verwendet.
  - Wenn Sie Nein wählen, wird die `CachingOptimizedForUncompressedObjects`-Caching-Richtlinie verwendet.

API-Gateway-Ursprung

Im Folgenden finden Sie die Ursprungseinstellungen, die für Ihren API-Gateway-Ursprung in einer Multi-Tenant-Distribution CloudFront vorkonfiguriert sind.

Ursprungseinstellungen (vorkonfiguriert)

- Protokoll – Nur HTTPS
- HTTPS-Port — 443
- Minimaler Ursprung des SSL-Protokolls — TLSv1.2
- Ursprungspfad – Keiner
- Origin Access Control (nur Konsole) — CloudFront richtet das für dich ein
- Benutzerdefinierten Header hinzufügen – Keinen
- Origin Shield aktivieren – Nein
- Verbindungsversuche – 3
- Antwort-Timeout – 30
- Keep-alive Timeout — 5

Im Folgenden finden Sie die Cache-Einstellungen, die für Ihren API-Gateway-Ursprung in einer Multi-Tenant-Distribution CloudFront vorkonfiguriert sind.

Cache-Einstellungen (vorkonfiguriert)

- Objekte automatisch komprimieren – Ja
- Viewer-Protokollrichtlinie – Zu HTTPS weiterleiten
- Zulässige HTTP-Methode – `GET, HEAD, OPTIONS, PUT, POST, PATCH, DELETE`
- Cache-HTTP-Methoden – Nein
- gRPC-Anfragen zulassen über HTTP/2 — Nein
- Viewer-Zugriff einschränken – Nein
- Cache-Richtlinie – `CachingDisabled` (mögliche Werte: `UseOriginCacheControlHeaders`, `UseOriginCacheControlHeaders-QueryStrings`)
- Ursprungsanforderungsrichtlinie – `AllViewerExceptHostHeader` (mögliche Werte: `AllViewer`, `AllViewerandCloudFrontHeaders-2022-06`)
- Antwortheader-Richtlinie – keine
- Smooth Streaming – Nein
- Verschlüsselung auf Feldebene – Nein
- Zugriffsprotokolle in Echtzeit aktivieren — Nein
- Funktionen – Nein

Im Folgenden finden Sie die Einstellungen, die Sie für Ihren API-Gateway-Ursprung in einer Multi-Tenant-Distribution anpassen können.

Anpassbare Einstellungen

- Origin Shield aktivieren – (Standardeinstellung: Nein)
- Objekte automatisch komprimieren – (Standardeinstellung: Ja)

Benutzerdefinierter Ursprung und EC2-Instance

Im Folgenden finden Sie die Origin-Einstellungen, die für Ihren benutzerdefinierten Origin in einer Multi-Tenant-Distribution CloudFront vorkonfiguriert sind.

Ursprungseinstellungen (vorkonfiguriert)

- Protokoll – wie Viewer
- HTTP-Port – 80

- HTTPS-Port — 443
- Minimaler Ursprung des SSL-Protokolls — TLSv1.2
- Ursprungspfad – Keiner
- Benutzerdefinierten Header hinzufügen – Keinen
- Origin Shield aktivieren – Nein
- Verbindungsversuche – 3
- Antwort-Timeout – 30
- Keep-alive Zeitlimit — 5

Im Folgenden finden Sie die Cache-Einstellungen, die für Ihre benutzerdefinierte Origin- und EC2-Instance in einer Multi-Tenant-Distribution CloudFront vorkonfiguriert sind.

Cache-Einstellungen (vorkonfiguriert)

- Objekte automatisch komprimieren – Ja
- Viewer-Protokollrichtlinie – Zu HTTPS weiterleiten
- Zulässige HTTP-Methode – `GET, HEAD, OPTIONS, PUT, POST, PATCH, DELETE`
- Cache-HTTP-Methoden – Nein
- gRPC-Anfragen zulassen über HTTP/2 — Nein
- Viewer-Zugriff einschränken – Nein
- Cache-Richtlinie – `UseOriginCacheControlHeaders` (mögliche Werte: `UseOriginCacheControlHeaders-QueryStrings`, `CachingDisabled`, `CacheOptimized`, `CachingOptimizedForUncompressedObjects`)
- Ursprungsanforderungsrichtlinie – `AllViewer` (mögliche Werte: `AllViewerExceptHostHeader`, `AllViewerandCloudFrontHeaders-2022-06`)
- Antwortheader-Richtlinie – keine
- Smooth Streaming – Nein
- Verschlüsselung auf Feldebene – Nein
- Zugriffsprotokolle in Echtzeit aktivieren — Nein
- Funktionen – Nein

Im Folgenden finden Sie die Einstellungen, die Sie für Ihren benutzerdefinierten Ursprung und die EC2-Instance in einer Multi-Tenant-Distribution anpassen können.

Anpassbare Einstellungen

- Origin Shield aktivieren – (Standardeinstellung: Nein)
- Objekte automatisch komprimieren – (Standardeinstellung: Ja)
- Caching – (Standardeinstellung: `Cache by Default`)
  - Wenn `Cache by Default` ausgewählt ist, wird die `UseOriginCacheControlHeaders`-Cache-Richtlinie verwendet.
  - Wenn `Do Not Cache by Default` ausgewählt ist, wird die `CachingDisabled`-Cache-Richtlinie verwendet.
- Abfragezeichenfolge in Cache einbeziehen – (Standardeinstellung: Ja, wenn `Cache by Default` bereits ausgewählt ist)
  - Wenn `Do Not Cache by Default` bereits ausgewählt ist und Sie sich dann dafür entscheiden, die Abfragezeichenfolge in den Cache einzubeziehen, wird die Cache-Richtlinie `UseOriginCacheControlHeaders-QueryStrings` verwendet.

Elastic Load Balancing als Ursprung

Im Folgenden finden Sie die Origin-Einstellungen, die für Ihren Elastic Load Balancing Balancing-Ursprung in einer Multi-Tenant-Distribution CloudFront vorkonfiguriert sind.

Ursprungseinstellungen (vorkonfiguriert)

- Protokoll – Nur HTTPS
- HTTPS-Port — 443
- Minimaler Ursprung des SSL-Protokolls — TLSv1.2
- Ursprungspfad – Keiner
- Benutzerdefinierten Header hinzufügen – Keinen
- Origin Shield aktivieren – Nein
- Verbindungsversuche – 3
- Antwort-Timeout – 30
- Keep-alive Zeitlimit — 5

Im Folgenden finden Sie die Cache-Einstellungen, die für Ihren Elastic Load Balancing Balancing-Ursprung in einer Multi-Tenant-Distribution CloudFront vorkonfiguriert sind.

Cache-Einstellungen (vorkonfiguriert)

- Objekte automatisch komprimieren – Ja
- Viewer-Protokollrichtlinie – Zu HTTPS weiterleiten
- Zulässige HTTP-Methode – `GET, HEAD, OPTIONS, PUT, POST, PATCH, DELETE`
- Cache-HTTP-Methoden – Nein
- gRPC-Anfragen zulassen über HTTP/2 — Nein
- Viewer-Zugriff einschränken – Nein
- Caching – (Standardeinstellung: `Cache by Default`)
  - Wenn `Cache by Default` ausgewählt ist, wird die `UseOriginCacheControlHeaders`-Cache-Richtlinie verwendet.
  - Wenn `Do Not Cache by Default` ausgewählt ist, wird die `CachingDisabled`-Cache-Richtlinie verwendet.
- Abfragezeichenfolge in Cache einbeziehen – (Standardeinstellung: Ja, wenn `Cache by Default` bereits ausgewählt ist)
  - Wenn `Do Not Cache by Default` bereits ausgewählt ist und Sie sich dann dafür entscheiden, die Abfragezeichenfolge in den Cache einzubeziehen, wird die Cache-Richtlinie `UseOriginCacheControlHeaders-QueryStrings` verwendet.
- Ursprungsanforderungsrichtlinie – `All Viewer` (mögliche Werte: `AllViewerExceptHostHeader`, `AllViewerandCloudFrontHeaders-2022-06`)
- Antwortheader-Richtlinie – keine
- Smooth Streaming – Nein
- Verschlüsselung auf Feldebene – Nein
- Zugriffsprotokolle in Echtzeit aktivieren — Nein
- Funktionen – Nein

Im Folgenden finden Sie die Einstellungen, die Sie für Elastic Load Balancing als Ursprung in einer Multi-Tenant-Distribution anpassen können.

Anpassbare Einstellungen

- Origin Shield aktivieren – (Standardeinstellung: Nein)
- Objekte automatisch komprimieren – (Standardeinstellung: Ja)
- Caching – (Standardeinstellung: `Cache by Default`)

  - Wenn `Cache by Default` ausgewählt ist, wird die `UseOriginCacheControlHeaders`-Cache-Richtlinie verwendet.
  - Wenn `Do Not Cache by Default` ausgewählt ist, wird die `CachingDisabled`-Cache-Richtlinie verwendet.
- Abfragezeichenfolge in Cache einbeziehen – (Standardeinstellung: Ja, wenn `Cache by Default` bereits ausgewählt ist)
  - Wenn `Do Not Cache by Default` bereits ausgewählt ist und Sie sich dann dafür entscheiden, die Abfragezeichenfolge in den Cache einzubeziehen, wird die Cache-Richtlinie `UseOriginCacheControlHeaders-QueryStrings` verwendet.

MediaPackage v1-Ursprung

Im Folgenden finden Sie die Origin-Einstellungen, die für Ihren MediaPackage v1-Origin in einer Multi-Tenant-Distribution CloudFront vorkonfiguriert sind.

Ursprungseinstellungen (vorkonfiguriert)

- Protokoll – Nur HTTPS
- HTTPS-Port — 443
- Minimaler Ursprung des SSL-Protokolls — TLSv1.2
- Herkunftspfad — Sie geben diesen Pfad an, indem Sie Ihre MediaPackage URL eingeben.
- Benutzerdefinierten Header hinzufügen – Keinen
- Origin Shield aktivieren – Nein
- Verbindungsversuche – 3
- Antwort-Timeout – 30
- Keep-alive Auszeit — 5

Im Folgenden finden Sie die Cache-Einstellungen, die für Ihren MediaPackage v1-Ursprung in einer Multi-Tenant-Distribution CloudFront vorkonfiguriert sind.

Cache-Einstellungen (vorkonfiguriert)

- Objekte automatisch komprimieren – Ja
- Viewer-Protokollrichtlinie – Zu HTTPS weiterleiten

- Zulässige HTTP-Methode – `GET, HEAD`
- Cache-HTTP-Methoden – Nein
- gRPC-Anfragen zulassen über HTTP/2 — Nein
- Viewer-Zugriff einschränken – Nein
- Cache-Richtlinie – `Elemental-MediaPackage`
- Ursprungsanforderungsrichtlinie – Keine
- Antwortheader-Richtlinie – keine
- Smooth Streaming – Nein
- Verschlüsselung auf Feldebene – Nein
- Zugriffsprotokolle in Echtzeit aktivieren — Nein
- Funktionen – Nein

MediaPackage v2-Ursprung

Im Folgenden finden Sie die Origin-Einstellungen, die für Ihren MediaPackage v2-Ursprung in einer Multi-Tenant-Distribution CloudFront vorkonfiguriert sind.

Ursprungseinstellungen (vorkonfiguriert)

- Origin Access Control — CloudFront richtet das für dich ein und fügt die Richtlinie hinzu
- Protokoll – Nur HTTPS
- HTTPS-Port — 443
- Minimaler Ursprung des SSL-Protokolls — TLSv1.2
- Ursprungspfad – Keiner
- Benutzerdefinierten Header hinzufügen – Keinen
- Origin Shield aktivieren – Nein
- Verbindungsversuche – 3
- Antwort-Timeout – 30
- Keep-alive Zeitlimit — 5

Im Folgenden finden Sie die Cache-Einstellungen, die für Ihren MediaPackage v2-Ursprung in einer Multi-Tenant-Distribution CloudFront vorkonfiguriert sind.

Cache-Einstellungen (vorkonfiguriert)

- Objekte automatisch komprimieren – Ja
- Viewer-Protokollrichtlinie – Zu HTTPS weiterleiten
- Zulässige HTTP-Methode – `GET, HEAD`
- Cache-HTTP-Methoden – Nein
- gRPC-Anfragen zulassen über HTTP/2 — Nein
- Viewer-Zugriff einschränken – Nein
- Cache-Richtlinie – `Elemental-MediaPackage`
- Ursprungsanforderungsrichtlinie – Keine
- Antwortheader-Richtlinie – keine
- Smooth Streaming – Nein
- Verschlüsselung auf Feldebene – Nein
- Zugriffsprotokolle in Echtzeit aktivieren — Nein
- Funktionen – Nein

MediaTailor Herkunft

Im Folgenden finden Sie die Origin-Einstellungen, die für Ihren MediaTailor Ursprung in einer Multi-Tenant-Distribution CloudFront vorkonfiguriert sind.

Ursprungseinstellungen (vorkonfiguriert)

- Protokoll – Nur HTTPS
- HTTPS-Port — 443
- Minimaler Ursprung des SSL-Protokolls — TLSv1.2
- Herkunftspfad — Sie geben diesen Pfad an, indem Sie Ihre MediaPackage URL eingeben.
- Benutzerdefinierten Header hinzufügen – Keinen
- Origin Shield aktivieren – Nein
- Verbindungsversuche – 3
- Antwort-Timeout – 30
- Keep-alive Auszeit — 5

Im Folgenden finden Sie die Cache-Einstellungen, die für Ihren MediaTailor Ursprung in einer Multi-Tenant-Distribution CloudFront vorkonfiguriert sind.

Cache-Einstellungen (vorkonfiguriert)

- Objekte automatisch komprimieren – Ja
- Viewer-Protokollrichtlinie – Zu HTTPS weiterleiten
- Zulässige HTTP-Methode – `GET, HEAD`
- Cache-HTTP-Methoden – Nein
- gRPC-Anfragen zulassen über HTTP/2 — Nein
- Viewer-Zugriff einschränken – Nein
- Cache-Richtlinie – Keine
- Ursprungsanforderungsrichtlinie – `Elemental-MediaTailor-PersonalizedManifests`
- Antwortheader-Richtlinie – keine
- Smooth Streaming – Nein
- Verschlüsselung auf Feldebene – Nein
- Zugriffsprotokolle in Echtzeit aktivieren — Nein
- Funktionen – Nein

Referenz für alle Distributionseinstellungen

Sie können wählen, ob Sie Ihre CloudFront Vertriebseinstellungen manuell bearbeiten möchten, wenn Sie Ihre Distribution erstellen oder aktualisieren. Sie können die folgenden Einstellungen bearbeiten.

CloudFront Konfiguriert jedoch die meisten Vertriebseinstellungen für Sie, basierend auf der Herkunft Ihrer Inhalte. Weitere Informationen finden Sie unter Referenz für vorkonfigurierte Distributionseinstellungen.

Weitere Informationen zum Erstellen oder Aktualisieren einer Verteilung mithilfe der CloudFront-Konsole finden Sie unter the section called “Eine Distribution erstellen” oder the section called “Eine Verteilung aktualisieren”.

Themen

- Ursprungseinstellungen
- Einstellungen für das Cache-Verhalten
- Distribution Settings (Einstellungen für die Verteilung)
- Benutzerdefinierte Fehlerseiten und Zwischenspeicherung von Fehlern
- Geografische Einschränkungen

Ursprungseinstellungen

Wenn Sie die CloudFront Konsole verwenden, um eine Distribution zu erstellen oder zu aktualisieren, geben Sie Informationen zu einem oder mehreren Speicherorten, den sogenannten Origins, an denen Sie die Originalversionen Ihrer Webinhalte speichern. CloudFront ruft Ihre Webinhalte von Ihren Ursprüngen ab und stellt sie den Zuschauern mithilfe eines weltweiten Netzwerks von Edge-Servern zur Verfügung.

Informationen zur aktuell gültigen maximalen Anzahl von Ursprüngen, die Sie für eine Verteilung erstellen können, oder zum Anfordern eines höheren Kontingents finden Sie unter the section called “Allgemeine Kontingente für Verteilungen”.

Wenn Sie einen Ursprung löschen möchten, müssen Sie zunächst die Cache-Verhalten bearbeiten oder löschen, die mit diesem Ursprung verknüpft sind.

> **Important**
>
> Vergewissern Sie sich beim Löschen eines Ursprungs, dass die zuvor von diesem Ursprung bereitgestellten Dateien in einem anderen Ursprung verfügbar sind und dass Ihre Cache-Verhalten Anfragen für diese Dateien jetzt an den neuen Ursprung weiterleiten.

Beim Erstellen oder Aktualisieren einer Verteilung geben Sie die folgenden Werte für jeden Ursprung an.

Themen

- Ursprungsdomäne
- Protokoll (nur benutzerdefinierte Ursprünge)
- Ursprungspfad
- Name

- Ursprungszugriff (nur Amazon-S3-Ursprünge)
- Benutzerdefinierten Header hinzufügen
- Origin Shield aktivieren
- Verbindungsversuche
- Verbindungstimeout
- Antwort-Timeout
- Timeout für den Abschluss der Antwort
- Keep-alive Timeout (nur benutzerdefinierte und VPC-Ursprünge)
- Kontingent für Antwort- und Keep-Alive-Timeout

Ursprungsdomäne

Die Ursprungsdomain ist der DNS-Domainname der Ressource, von der Objekte für Ihren Ursprung abgerufen CloudFront werden, z. B. ein Amazon S3 S3-Bucket oder ein HTTP-Server. Beispiel:

- Amazon S3-Bucket – *amzn-s3-demo-bucket*.s3.*us-west-2*.amazonaws.com

> **Note**
>
> Wenn Sie den S3-Bucket kürzlich erstellt haben, gibt die CloudFront Verteilung möglicherweise `HTTP 307 Temporary Redirect` Antworten für bis zu 24 Stunden zurück. Es kann bis zu 24 Stunden dauern, bis der S3-Bucket-Name in alle AWS -Regionen weitergegeben wird. Wenn die Weitergabe abgeschlossen ist, stoppt die Verteilung das Senden dieser Umleitungsantworten automatisch. Sie müssen keine Maßnahmen ergreifen. Weitere Informationen finden Sie unter Warum erhalte ich eine HTTP 307 Temporary Redirect-Antwort von Amazon S3? und Temporäre Anforderungsumleitung.

- Amazon S3-Bucket, der als Website konfiguriert ist – *amzn-s3-demo-bucket*.s3-website.*us-west-2*.amazonaws.com
- MediaStore Behälter — *examplemediastore*.data.mediastore.*us-west-1*.amazonaws.com
- MediaPackage Endpunkt — *examplemediapackage*.mediapackage.*us-west-1*.amazonaws.com
- Amazon EC2-Instance – *ec2-203-0-113-25*.compute-1.amazonaws.com

- Elastic Load Balancing-Load Balancer – *example-load-balancer-1234567890*.*us-west-2*.elb.amazonaws.com
- Ihr eigener Webserver – www.example.com

Wählen Sie den Domänennamen im Feld Origin domain (Ursprungsdomäne) aus oder geben Sie den Namen ein. Ressourcen aus Opt-in-Regionen müssen manuell eingegeben werden. Bei dem Domänennamen wird die Groß- und Kleinschreibung nicht berücksichtigt. Ihre Ursprungsdomain muss einen öffentlich auflösbaren DNS-Namen haben, der Anforderungen von Clients über das Internet an Ziele weiterleitet.

Wenn Sie so konfigurieren, dass eine Verbindung CloudFront zu Ihrem Ursprung über HTTPS hergestellt wird, muss einer der Domainnamen im Zertifikat mit dem Domainnamen übereinstimmen, den Sie für Origin Domain Name angeben. Wenn kein Domainname übereinstimmt, wird der HTTP-Statuscode 502 (Bad Gateway) an den Betrachter CloudFront zurückgegeben. Weitere Informationen erhalten Sie unter Domainnamen in der CloudFront Distribution und im Zertifikat und SSL/TLS Fehler bei der Aushandlung zwischen CloudFront und einem benutzerdefinierten Ursprungsserver.

> **Note**
>
> Wenn Sie eine Ursprungsanforderungsrichtlinie verwenden, die den Viewer-Host-Header an den Ursprung weiterleitet, muss der Ursprung mit einem Zertifikat antworten, das dem Viewer-Host-Header entspricht. Weitere Informationen finden Sie unter CloudFront Anforderungsheader hinzufügen.

Wenn es sich bei Ihrem Ursprung um einen Amazon S3-Bucket handelt, beachten Sie Folgendes:

- Wenn der Bucket als Website konfiguriert ist, geben Sie den statischen Amazon-S3-Endpunkt für das Hosten von Websites für Ihren Bucket ein. Sie dürfen den Bucket-Namen nicht aus der Liste im Feld Origin domain (Ursprungsdomäne) auswählen. Der statische Endpunkt für das Hosten von Websites wird in der Amazon-S3-Konsole auf der Seite Properties (Eigenschaften) unter Static Website Hosting (Hosting der statischen Website) angezeigt. Weitere Informationen finden Sie unter the section called “Verwenden eines Amazon-S3-Buckets, der als Website-Endpunkt konfiguriert ist”.
- Wenn Sie für Ihren Bucket Amazon S3 Transfer Acceleration konfiguriert haben, geben Sie den s3-accelerate-Endpunkt für Origin domain (Ursprungsdomäne) nicht an.

- Wenn Sie einen Bucket von einem anderen AWS Konto verwenden und der Bucket nicht als Website konfiguriert ist, geben Sie den Namen im folgenden Format ein:

  bucket-name.s3.*region*.amazonaws.com

  Wenn sich Ihr Bucket in einer US-Region befindet und Amazon S3 Anforderungen an einen Standort in Nord-Virginia weiterleiten soll, verwenden Sie das folgende Format:

  bucket-name.s3.us-east-1.amazonaws.com

- Die Dateien müssen öffentlich lesbar sein, es sei denn, Sie sichern Ihre Inhalte in Amazon S3 mithilfe einer CloudFront Ursprungszugriffskontrolle. Weitere Informationen zur Zugriffskontrolle finden Sie unter the section called “Beschränken des Zugriffs auf einen Amazon-S3-Ursprung”.

> **Important**
>
> Wenn der Ursprung ein Amazon S3-Bucket ist, muss der Bucket-Name den Anforderungen für DNS-Namen entsprechen. Weitere Informationen finden Sie unter Bucket-Einschränkungen und -Limits im Benutzerhandbuch zu Amazon Simple Storage Service.

Wenn Sie den Wert der Origin-Domain für einen Ursprung ändern, beginnt CloudFront sofort, die Änderung an den CloudFront Edge-Standorten zu replizieren. Leitet weiterhin Anfragen an den vorherigen Ursprung CloudFront weiter, bis die Verteilungskonfiguration an einem bestimmten Edge-Standort aktualisiert ist. Sobald die Verteilungskonfiguration an diesem Edge-Standort aktualisiert wurde, CloudFront beginnt die Weiterleitung von Anfragen an den neuen Ursprung.

Wenn Sie den Ursprung ändern, CloudFront müssen die Edge-Caches nicht erneut mit Objekten aus dem neuen Ursprung gefüllt werden. Solange die Anzeigeanfragen in Ihrer Anwendung nicht geändert wurden, stellt CloudFront weiterhin Objekte bereit, die sich bereits in einem Edge-Cache befinden, bis die TTL für jedes Objekt abläuft oder bis selten angefragte Objekte entfernt wurden.

Protokoll (nur benutzerdefinierte Ursprünge)

> **Note**
>
> Dies gilt nur für benutzerdefinierte Ursprünge.

Die Protokollrichtlinie, die Sie beim Abrufen CloudFront von Objekten von Ihrem Ursprung verwenden möchten.

Wählen Sie einen der folgenden Werte aus:

- Nur HTTP: CloudFront verwendet nur HTTP für den Zugriff auf den Ursprung.

> **Important**
>
> HTTP only (Nur HTTP) ist die Standardeinstellung, wenn der Ursprung ein statischer Amazon S3-Endpunkt für das Hosten von Websites ist, da Amazon S3 keine HTTPS-Verbindungen für statische Endpunkte für das Hosten von Websites unterstützt. Die CloudFront Konsole unterstützt das Ändern dieser Einstellung für statische Amazon S3 S3-Website-Hosting-Endpunkte nicht.

- Nur HTTPS: CloudFront verwendet nur HTTPS für den Zugriff auf den Ursprung.
- Match Viewer: CloudFront Kommuniziert je nach Protokoll der Viewer-Anfrage über HTTP oder HTTPS mit deinem Ursprung. CloudFront speichert das Objekt nur einmal im Cache, auch wenn Zuschauer Anfragen sowohl mit HTTP- als auch mit HTTPS-Protokollen stellen.

> **Important**
>
> Bei HTTPS-Viewer-Anfragen, CloudFront die an diesen Ursprung weiterleiten, muss einer der Domainnamen im SSL/TLS Zertifikat auf Ihrem Ursprungsserver mit dem Domainnamen übereinstimmen, den Sie für die Origin-Domain angeben. Andernfalls CloudFront reagiert auf die Viewer-Anfragen mit dem HTTP-Statuscode 502 (Bad Gateway), anstatt das angeforderte Objekt zurückzugeben. Weitere Informationen finden Sie unter the section called “Anforderungen für die Verwendung von SSL/TLS Zertifikaten mit CloudFront”.

Themen

- HTTP-Port
- HTTPS-Port
- Mindest-SSL-Protokoll für Ursprung

HTTP-Port

> **Note**
>
> Dies gilt nur für benutzerdefinierte Ursprünge.

(Optional) Sie können den HTTP-Port angeben, den der benutzerdefinierte Ursprung überwacht. Zu den gültigen Werten gehören die Ports 80, 443 und 1024 bis 65535. Der Standardwert ist Port 80.

> **Important**
>
> Port 80 ist die Standardeinstellung, wenn der Ursprung ein statischer Amazon S3-Endpunkt für das Hosten von Websites ist, da Amazon S3 den Port 80 nur für statische Endpunkte für das Hosten von Websites unterstützt. Die CloudFront Konsole unterstützt das Ändern dieser Einstellung für statische Amazon S3 S3-Website-Hosting-Endpunkte nicht.

HTTPS-Port

> **Note**
>
> Dies gilt nur für benutzerdefinierte Ursprünge.

(Optional) Sie können den HTTPS-Port angeben, den der benutzerdefinierte Ursprung überwacht. Zu den gültigen Werten gehören die Ports 80, 443 und 1024 bis 65535. Der Standardwert ist Port 443. Wenn Protocol (Protokoll) auf HTTP only (Nur HTTP) festgelegt wird, können Sie keinen Wert für HTTPS port (HTTPS-Port) angeben.

Mindest-SSL-Protokoll für Ursprung

> **Note**
>
> Dies gilt nur für benutzerdefinierte Ursprünge.

Wählen Sie das TLS/SSL Mindestprotokoll aus, das verwendet CloudFront werden kann, wenn es eine HTTPS-Verbindung zu Ihrem Ursprung herstellt. Da niedrigere TLS-Protokollversionen weniger

sicher sind, sollten Sie das neueste TLS-Protokoll auswählen, das Ihr Ursprung unterstützt. Wenn Protocol (Protokoll) auf HTTP only (Nur HTTP) festgelegt wird, können Sie keinen Wert für Minimum origin SSL protocol (SSL-Mindestursprungsprotokoll) angeben.

Wenn Sie die CloudFront API verwenden, um das CloudFront zu verwendende TLS/SSL Protokoll festzulegen, können Sie kein Mindestprotokoll festlegen. Stattdessen geben Sie alle TLS/SSL Protokolle an, die Sie mit Ihrem Origin verwenden CloudFront können. Weitere Informationen finden Sie OriginSslProtocolsin der Amazon CloudFront API-Referenz.

Ursprungspfad

Wenn Sie Ihre Inhalte aus einem Verzeichnis in Ihrem Ursprungsverzeichnis anfordern möchten CloudFront , geben Sie den Verzeichnispfad ein, der mit einem Schrägstrich (/) beginnt. CloudFront hängt den Verzeichnispfad an den Wert der Origin-Domain an, zum Beispiel. **`cf-origin.example.com/production/images`** Fügen Sie keinen Schrägstrich (/) am Pfadende hinzu.

Angenommen, Sie haben die folgenden Werte für Ihre Verteilung angegeben:

- Origin domain (Ursprungsdomäne) – Ein Amazon-S3-Bucket mit dem Namen **`amzn-s3-demo-bucket`**
- Origin path (Ursprungspfad) – **`/production`**
- Alternate domain names (Alternative Domänennamen) (CNAME) – **`example.com`**

Wenn ein Benutzer einen Browser `example.com/index.html` aufruft, CloudFront sendet eine Anfrage an Amazon S3 für`amzn-s3-demo-bucket/production/index.html`.

Wenn ein Benutzer einen Browser `example.com/acme/index.html` aufruft, CloudFront sendet eine Anfrage an Amazon S3 für`amzn-s3-demo-bucket/production/acme/index.html`.

Name

Eine Zeichenfolge, die diesen Ursprung eindeutig in dieser Verteilung identifiziert. Sie können zusätzlich zum Standard-Cache-Verhalten auch Cache-Verhalten erstellen. In diesem Fall verwenden Sie den Namen, den Sie hier angeben, um den Ursprung zu identifizieren, an den Sie eine Anforderung weiterleiten möchten CloudFront , wenn die Anforderung dem Pfadmuster für dieses Cache-Verhalten entspricht.

Ursprungszugriff (nur Amazon-S3-Ursprünge)

> **Note**
>
> Dies gilt nur für Amazon S3-Bucket-Ursprünge (d. h. Ursprünge, die nicht den statischen S3-Website-Endpunkt verwenden).

Wählen Sie Origin-Zugriffskontrolleinstellungen (empfohlen), wenn Sie den Zugriff auf einen Amazon S3 S3-Bucket-Ursprung auf bestimmte CloudFront Distributionen beschränken möchten.

Wählen Sie Public (Öffentlich) aus, wenn der Amazon-S3-Bucket-Ursprung öffentlich zugänglich ist.

Weitere Informationen finden Sie unter the section called “Beschränken des Zugriffs auf einen Amazon-S3-Ursprung”.

Informationen darüber, wie Sie von Benutzern verlangen können, auf Objekte in einem benutzerdefinierten Ursprung zuzugreifen, indem sie nur CloudFront URLs verwenden, finden Sie unterthe section called “Beschränken des Zugriffs auf Dateien auf benutzerdefinierten Ursprungsservern”.

Benutzerdefinierten Header hinzufügen

Wenn Sie benutzerdefinierte Header hinzufügen CloudFront möchten, wann immer eine Anfrage an Ihren Ursprung gesendet wird, geben Sie den Header-Namen und seinen Wert an. Weitere Informationen finden Sie unter the section called “Hinzufügen benutzerdefinierter Header zu Ursprungsanforderungen”.

Informationen zur derzeit maximalen Anzahl von benutzerdefinierten Headern, die Sie hinzufügen können, zur maximalen Länge eines benutzerdefinierten Header-Namens und -Werts und zur maximalen Gesamtlänge aller Header-Namen und -Werte finden Sie unter Kontingente.

Origin Shield aktivieren

Wähle Ja, um CloudFront Origin Shield zu aktivieren. Weitere Informationen über Origin Shield erhalten Sie unter the section called “Verwenden von Origin Shield”.

Verbindungsversuche

Du kannst festlegen, wie oft CloudFront versucht wird, eine Verbindung zum Origin herzustellen. Sie können 1, 2 oder 3 als Anzahl der Versuche angeben. Die Standardanzahl (wenn Sie nichts anderes angeben) ist 3.

Verwenden Sie diese Einstellung zusammen mit dem Verbindungs-Timeout, um anzugeben, wie lange es CloudFront dauert, bis versucht wird, eine Verbindung zum sekundären Ursprung herzustellen, oder eine Fehlerantwort an den Viewer zurückgegeben wird. CloudFront Wartet standardmäßig bis zu 30 Sekunden (3 Versuche à 10 Sekunden), bevor versucht wird, eine Verbindung zum sekundären Ursprung herzustellen, oder eine Fehlerantwort zurückgegeben wird. Sie können diese Zeit reduzieren, indem Sie weniger Versuche, ein kürzeres Verbindungs-Timeout, oder beides angeben.

Wenn die angegebene Anzahl von Verbindungsversuchen fehlschlägt, CloudFront führt eine der folgenden Aktionen aus:

- Wenn der Ursprung Teil einer Ursprungsgruppe ist, wird CloudFront versucht, eine Verbindung zum sekundären Ursprung herzustellen. Wenn die angegebene Anzahl von Verbindungsversuchen mit dem sekundären Ursprung fehlschlägt, wird eine Fehlerantwort an den Viewer CloudFront zurückgegeben.
- Wenn der Ursprung nicht Teil einer Ursprungsgruppe ist, wird eine Fehlerantwort an den Betrachter CloudFront zurückgegeben.

Für einen benutzerdefinierten Ursprung (einschließlich eines Amazon S3 S3-Buckets, der mit statischem Website-Hosting konfiguriert ist) gibt diese Einstellung auch an, wie oft CloudFront versucht wird, eine Antwort vom Ursprung zu erhalten. Weitere Informationen finden Sie unter the section called “Antwort-Timeout”.

Verbindungstimeout

Das Verbindungs-Timeout ist die Anzahl der Sekunden, die CloudFront gewartet werden, wenn versucht wird, eine Verbindung zum Ursprung herzustellen. Sie können eine Anzahl von Sekunden zwischen 1 und 10 (inklusive) angeben. Das Standardtimeout (wenn Sie nichts anderes angeben) beträgt 10 Sekunden.

Verwenden Sie diese Einstellung zusammen mit Verbindungsversuchen, um anzugeben, wie lange CloudFront es dauert, bis versucht wird, eine Verbindung zum sekundären Ursprung herzustellen, oder bis eine Fehlerantwort an den Viewer zurückgegeben wird. CloudFront Wartet standardmäßig

bis zu 30 Sekunden (3 Versuche à 10 Sekunden), bevor versucht wird, eine Verbindung zum sekundären Ursprung herzustellen, oder eine Fehlerantwort zurückgegeben wird. Sie können diese Zeit reduzieren, indem Sie weniger Versuche, ein kürzeres Verbindungs-Timeout, oder beides angeben.

Wenn innerhalb der angegebenen Anzahl von Sekunden CloudFront keine Verbindung zum Ursprung hergestellt wird, CloudFront wird eine der folgenden Aktionen ausgeführt:

- Wenn die angegebene Anzahl von Verbindungsversuchen mehr als 1 beträgt, wird erneut CloudFront versucht, eine Verbindung herzustellen. CloudFront versucht bis zu dreimal, je nach dem Wert der Verbindungsversuche.
- Wenn alle Verbindungsversuche fehlschlagen und der Ursprung Teil einer Ursprungsgruppe ist, versucht CloudFront , eine Verbindung mit dem sekundären Ursprung herzustellen. Wenn die angegebene Anzahl von Verbindungsversuchen mit dem sekundären Ursprung fehlschlägt, wird eine Fehlerantwort an den Viewer CloudFront zurückgegeben.
- Wenn alle Verbindungsversuche fehlschlagen und der Ursprung nicht Teil einer Ursprungsgruppe ist, wird eine Fehlerantwort an den Viewer CloudFront zurückgegeben.

Antwort-Timeout

Das Ursprungs-Reaktions-Timeout, das auch als Ursprungs-Lese-Timeout oder Ursprungs-Anforderungs-Timeout bezeichnet wird, gilt für folgende Werte:

- Wie lange (in Sekunden) auf eine Antwort gewartet wird, nachdem eine Anfrage an den Ursprung weitergeleitet CloudFront wurde.
- Wie lange (in Sekunden) nach dem Empfang eines Antwortpakets vom Ursprung und vor dem Empfang des nächsten Pakets CloudFront gewartet wird.

> **Tip**
>
> Wenn Sie den Timeout-Wert erhöhen möchten, da die Viewer HTTP 504-Statuscodefehler erhalten, suchen Sie nach anderen Möglichkeiten zur Vermeidung dieser Fehler, bevor Sie den Timeout-Wert ändern. Vorschläge zur Fehlerbehebung finden Sie unter the section called “HTTP 504-Statuscode (Gateway Timeout)”.

CloudFront Das Verhalten hängt von der HTTP-Methode in der Viewer-Anfrage ab:

- `GET`und `HEAD` Anfragen — Wenn der Ursprung nicht oder nicht innerhalb der Dauer des Antwort-Timeouts reagiert, wird die CloudFront Verbindung unterbrochen. CloudFront versucht erneut, eine Verbindung entsprechend dem Wert von the section called “Verbindungsversuche” herzustellen.
- `DELETE`,`OPTIONS`, `PATCH`PUT, und `POST` Anfragen — Wenn der Absender für die Dauer des Lese-Timeouts nicht reagiert, CloudFront bricht er die Verbindung ab und versucht nicht erneut, den Ursprung zu kontaktieren. Der Client kann die Anfrage erneut senden, falls erforderlich.

Timeout für den Abschluss der Antwort

> **Note**
>
> Das Timeout für den Abschluss der Antwort unterstützt das Feature zur kontinuierlichen Bereitstellung nicht.

Die Zeit (in Sekunden), in der eine Anfrage von CloudFront der Quelle geöffnet bleiben und auf eine Antwort warten kann. Wenn bis zu diesem Zeitpunkt keine vollständige Antwort vom Ursprung eingegangen ist, wird die Verbindung CloudFront beendet.

Im Gegensatz zum Antwort-Timeout, bei dem es sich um die Wartezeit für einzelne Antwortpakete handelt, ist der Timeout für den Antwortabschluss die maximal zulässige CloudFront Wartezeit, bis die Antwort abgeschlossen ist. Sie können diese Einstellung verwenden, um sicherzustellen, dass CloudFront nicht unbegrenzt auf einen langsamen oder nicht reagierenden Ursprung gewartet wird, auch wenn andere Timeout-Einstellungen eine längere Wartezeit zulassen.

Dieses maximale Timeout beinhaltet das, was Sie für andere Timeout-Einstellungen angegeben haben, und die Anzahl der Verbindungsversuche bei jedem erneuten Versuch. Sie können diese Einstellungen zusammen verwenden, um festzulegen, wie lange CloudFront auf die vollständige Anfrage gewartet wird und wann die Anfrage beendet werden soll, unabhängig davon, ob sie abgeschlossen ist oder nicht.

Angenommen, Sie legen die folgenden Einstellungen fest:

- Verbindungsversuche: 3
- Verbindungs-Timeout: 10 Sekunden
- Antwort-Timeout: 30 Sekunden
- Timeout für den Abschluss der Antwort: 60 Sekunden

Das bedeutet, dass versucht CloudFront wird, eine Verbindung zum Ursprung herzustellen (bis zu 3 Versuche insgesamt), wobei bei jedem Verbindungsversuch eine Zeitüberschreitung von 10 Sekunden eintritt. Sobald die Verbindung hergestellt ist, CloudFront wird bis zu 30 Sekunden gewartet, bis der Ursprung auf die Anfrage antwortet, bis er das letzte Paket der Antwort erhält.

Unabhängig von der Anzahl der Verbindungsversuche oder dem Antwort-Timeout CloudFront wird die Verbindung beendet, wenn die vollständige Antwort vom Ursprung länger als 60 Sekunden dauert. CloudFront gibt dem Betrachter dann eine the section called “HTTP 504-Statuscode (Gateway Timeout)” Fehlerantwort oder eine benutzerdefinierte Fehlerantwort zurück, falls Sie eine angegeben haben.

Hinweise

- Wenn Sie das Timeout für den Abschluss der Antwort festlegen, muss der Wert gleich oder größer als der Wert für das Antwort-Timeout (Ursprungs-Lese-Timeout) sein.
- Wenn Sie keinen Wert für das Timeout für den Abschluss der Antwort festlegen, CloudFront wird kein Höchstwert erzwungen.

Keep-alive Timeout (nur benutzerdefinierte und VPC-Ursprünge)

Das Keep-Alive-Timeout gibt an, wie lange (in Sekunden) CloudFront versucht wird, eine Verbindung zu Ihrem benutzerdefinierten Ursprung aufrechtzuerhalten, nachdem dieser das letzte Paket einer Antwort erhalten hat. Durch eine persistente Verbindung wird die Zeit gespart, die erforderlich ist, um die TCP-Verbindung erneut herzustellen und einen weiteren TLS-Handshake für nachfolgende Anfragen durchzuführen. Durch das Erhöhen des Keepalive-Timeouts können Sie die Metrik „Anfrage-pro-Verbindung” für Verteilungen verbessern.

Note

Damit der Keep-alive Timeout-Wert wirksam wird, muss Ihr Origin so konfiguriert sein, dass persistente Verbindungen zulässig sind.

Kontingent für Antwort- und Keep-Alive-Timeout

- Der Standardwert für das Antwort-Timeout beträgt 30 Sekunden.

- Der Standardwert für das Keep-Alive-Timeout beträgt 5 Sekunden.

Wenn Sie eine Erhöhung des Timeouts für Ihre anfordern AWS-Konto, aktualisieren Sie Ihre Distributionsquellen so, dass sie die gewünschten Werte für das Antwort-Timeout und das Keep-Alive-Timeout haben. Eine Kontingenterhöhung für Ihr Konto führt nicht automatisch zu einer Aktualisierung Ihrer Ursprünge. Wenn Sie beispielsweise eine Lambda@Edge-Funktion verwenden, um ein Keep-Alive-Timeout von 90 Sekunden festzulegen, muss Ihr Ursprung bereits über ein Keep-Alive-Timeout von 90 Sekunden oder mehr verfügen. Andernfalls kann Ihre Lambda@Edge-Funktion möglicherweise nicht ausgeführt werden.

Weitere Informationen zu Distributionskontingenten und dazu, wie Sie eine Erhöhung anfordern, finden Sie unter Allgemeine Kontingente für Verteilungen.

Einstellungen für das Cache-Verhalten

Durch die Einstellung des Cache-Verhaltens können Sie eine Vielzahl von CloudFront Funktionen für ein bestimmtes URL-Pfadmuster für Dateien auf Ihrer Website konfigurieren. Beispielsweise kann ein Cache-Verhalten für alle `.jpg`-Dateien im `images`-Verzeichnis auf einem Webserver gelten, den Sie als Ursprungs-Server für CloudFront verwenden. Zu den Funktionen, die Sie für jedes Cache-Verhalten konfigurieren können, gehören folgende:

- Das Pfadmuster
- Wenn Sie mehrere Ursprünge für Ihre CloudFront Distribution konfiguriert haben, den Ursprung, an den Sie Ihre Anfragen weiterleiten CloudFront möchten
- Die Angabe, ob Abfragezeichenfolgen an Ihren Ursprung weiterleitet werden sollen
- Die Angabe, ob für den Zugriff auf die angegebenen Dateien signierte URLs erforderlich sind
- Die Angabe, ob Benutzer HTTPS für den Zugriff auf diese Dateien verwenden müssen
- Die Mindestdauer, für die diese Dateien im CloudFront Cache verbleiben, unabhängig vom Wert der `Cache-Control` Header, die Ihr Ursprung den Dateien hinzufügt

Wenn Sie eine neue Verteilung erstellen, geben Sie Einstellungen für das Standard-Cache-Verhalten an, das alle Anfragen automatisch an den Ursprung weiterleitet, den Sie beim Erstellen der Verteilung angeben. Nachdem Sie eine Verteilung erstellt haben, können Sie zusätzliche Cache-Verhaltensweisen erstellen, die definieren, wie sie CloudFront reagiert, wenn sie eine Anforderung für Objekte empfängt, die einem Pfadmuster entsprechen, `*.jpg` z. B. Wenn Sie zusätzliche Cache-Verhalten erstellen, wird das Standard-Cache-Verhalten immer als Letztes verarbeitet. Andere

Cache-Verhaltensweisen werden in der Reihenfolge verarbeitet, in der sie in der CloudFront Konsole aufgeführt sind, oder, wenn Sie die CloudFront API verwenden, in der Reihenfolge, in der sie im `DistributionConfig` Element für die Verteilung aufgeführt sind. Weitere Informationen finden Sie unter Pfadmuster.

Wenn Sie ein Cache-Verhalten erstellen, geben Sie den einen Ursprung an, von dem Sie Objekte abrufen CloudFront möchten. Wenn Sie also Objekte aus all Ihren Ursprüngen verteilen CloudFront möchten, müssen Sie mindestens so viele Cache-Verhalten (einschließlich des Standard-Cache-Verhaltens) wie Ursprünge haben. Wenn Sie beispielsweise zwei Ursprünge und nur das Standard-Cache-Verhalten haben, bewirkt CloudFront das Standard-Cache-Verhalten, dass Objekte von einem der Ursprünge abgerufen werden, der andere Ursprung jedoch nie verwendet wird.

Informationen zur aktuell gültigen maximalen Anzahl von Cache-Verhaltensweisen, die Sie einer Verteilung hinzufügen können, oder zum Anfordern eines höheren Kontingents (früher als Limit bezeichnet) finden Sie unter Allgemeine Kontingente für Verteilungen.

Themen

- Pfadmuster
- Ursprung oder Ursprungsgruppe
- Viewer-Protokollrichtlinien
- Zulässige HTTP-Methoden
- Field-level Verschlüsselungskonfiguration
- Zwischengespeicherte HTTP-Methoden
- gRPC-Anfragen zulassen über HTTP/2
- Basierend auf den ausgewählten Anforderungsheadern
- Zulassungslisten-Header
- Zwischenspeicherung von Objekten
- Mindest-TTL
- Höchst-TTL
- Standard-TTL
- Cookies weiterleiten
- Zulassungslisten-Cookies
- Weiterleitung und Zwischenspeicherung von Abfragezeichenfolgen
- Zulassungsliste für Abfragezeichenfolgen

- Smooth Streaming
- Viewerzugriff einschränken (Signierte URLs oder signierte Cookies verwenden)
- Vertrauenswürdiger Signaturgeber
- AWS-Konto Ziffern
- Objekte automatisch komprimieren
- CloudFront Ereignis
- ARN der Lambda-Funktion
- Include body (Text einschließen)

Pfadmuster

Ein Pfadmuster (z. B. `images/*.jpg`) gibt an, für welche Anfragen dieses Cache-Verhalten gelten soll. Wenn eine CloudFront Endbenutzeranfrage eingeht, wird der angeforderte Pfad mit Pfadmustern in der Reihenfolge verglichen, in der das Cache-Verhalten in der Distribution aufgeführt ist. Die erste Übereinstimmung bestimmt, welches Cache-Verhalten auf diese Anfrage angewendet wird. Nehmen wir beispielsweise an, dass Sie drei Cache-Verhalten mit den folgenden drei Pfadmustern haben, in dieser Reihenfolge:

- `images/*.jpg`
- `images/*`
- `*.gif`

> **Note**
>
> Sie können optional einen Schrägstrich (/) am Anfang des Pfadmusters angeben, z. B. `/images/*.jpg` CloudFront Das Verhalten ist mit oder ohne das führende/identisch. Wenn Sie/nicht am Anfang des Pfads angeben, wird dieses Zeichen automatisch impliziert; CloudFront behandelt den Pfad auf dieselbe Weise, mit oder ohne das führende /. CloudFront Behandelt beispielsweise `/*product.jpg` dasselbe wie `*product.jpg`

Eine Anfrage für die Datei `images/sample.gif` entspricht nicht dem ersten Pfadmuster, sodass die zugehörigen Cache-Verhalten nicht auf die Anfrage angewendet werden. Die Datei entspricht dem zweiten Pfadmuster, sodass die mit dem zweiten Pfadmuster verknüpften Cache-Verhalten angewendet werden, auch wenn die Anfrage ebenfalls dem dritten Pfadmuster entspricht.

> **Note**
>
> Wenn Sie eine neue Verteilung erstellen, wird der Wert von Path Pattern für das Standard-Cache-Verhalten auf * (alle Dateien) gesetzt und kann nicht geändert werden. Dieser Wert bewirkt CloudFront , dass alle Anfragen für Ihre Objekte an den Ursprung weitergeleitet werden, den Sie im Ursprungsdomäne Feld angegeben haben. Wenn die Anforderung für ein Objekt nicht mit dem Pfadmuster für eines der anderen Cache-Verhalten übereinstimmt, wird das Verhalten CloudFront angewendet, das Sie im Standard-Cache-Verhalten angeben.

> **Important**
>
> Gehen Sie beim Definieren von Pfadmustern und ihrer Reihenfolge sorgfältig vor. Andernfalls gewähren Sie Benutzern möglicherweise unerwünscht Zugriff auf Ihre Inhalte. Nehmen wir beispielsweise an, eine Anfrage stimmt mit dem Pfadmuster für zwei Cache-Verhalten überein. Das erste Cache-Verhalten erfordert keine signierten URLs und das zweite Cache-Verhalten erfordert signierte URLs. Benutzer können auf die Objekte zugreifen, ohne eine signierte URL zu verwenden, da das Cache-Verhalten CloudFront verarbeitet wird, das mit der ersten Übereinstimmung verknüpft ist.

Wenn Sie mit einem MediaPackage Kanal arbeiten, müssen Sie spezifische Pfadmuster für das Cache-Verhalten angeben, das Sie für den Endpunkttyp für Ihren Ursprung definieren. Für einen DASH-Endpunkt geben Sie beispielsweise `*.mpd` in Path Pattern (Pfadmuster) ein. Weitere Informationen und spezielle Anweisungen finden Sie unter Servieren Sie Live-Videos, formatiert mit AWS Elemental MediaPackage.

Der von Ihnen angegebene Pfad gilt für Anfragen für alle Dateien im angegebenen Verzeichnis und in Unterverzeichnissen unterhalb des angegebenen Verzeichnisses. CloudFront berücksichtigt bei der Auswertung des Pfadmusters keine Abfragezeichenfolgen oder Cookies. Wenn ein `images`-Verzeichnis beispielsweise `product1`- und `product2`-Unterverzeichnisse enthält, gilt das Pfadmuster `images/*.jpg` für Anfragen für alle JPG-Dateien in den Verzeichnissen `images`, `images/product1` und `images/product2`. Wenn Sie auf die Dateien im Verzeichnis `images/product1` ein anderes Cache-Verhalten als auf die Dateien in den Verzeichnissen `images` und `images/product2` anwenden möchten, erstellen Sie ein separates Cache-Verhalten für `images/product1` und verschieben Sie dieses Cache-Verhalten an eine Stelle über (vor) dem Cache-Verhalten für das Verzeichnis `images`.

Sie können die folgenden Platzhalterzeichen in Ihrem Pfadmuster verwenden:

- `*` entspricht 0 oder mehr Zeichen.
- `?` entspricht genau 1 Zeichen.

Die folgenden Beispiele zeigen, wie die Platzhalterzeichen funktionieren:

Pfadmuster	Dateien, die dem Pfadmuster entsprechen
`*.jpg`	Alle .jpg-Dateien.
`images/*.jpg`	Alle .jpg-Dateien im `images`-Verzeichnis und in den Unterverzeichnissen unter dem `images`-Verzeichnis.
`a*.jpg`	• Alle .jpg-Dateien, deren Dateiname mit a beginnt, z. B. `apple.jpg` und `appalachian_trail_2012_05_21.jpg` . • Alle JPG-Dateien, deren Dateipfad mit a beginnt, z. B. `abra/cadabra/magic.jpg` .
`a??.jpg`	Alle .jpg-Dateien, deren Dateiname mit a beginnt, gefolgt von genau zwei anderen Zeichen, z. B. `ant.jpg` und `abe.jpg`.
`*.doc*`	Alle Dateien, deren Dateinamenerweiterung mit `.doc` beginnt, z. B. `.doc`-, `.docx`- und `.docm`-Dateien. Sie können das Pfadmuster `*.doc?` in diesem Fall nicht verwenden, weil dieses Pfadmuster nicht für Anfragen für `.doc`-Dateien gelten würde. Das Platzhalterzeichen `?` ersetzt genau ein Zeichen.

Die maximale Länge eines Pfadmusters beträgt 255 Zeichen. Der Wert kann folgende Zeichen enthalten:

- A-Z, a-z

Bei den Pfadmustern muss die Groß- und Kleinschreibung beachtet werden, so dass das Pfadmuster `*.jpg` nicht für die Datei `LOGO.JPG` gilt.

- 0-9
- _ - . * $ / ~ " ' @ : +
- & (übergeben und zurückgegeben als `&`)

Pfadnormalisierung

CloudFront normalisiert URI-Pfade, die mit RFC 3986 konform sind, und ordnet dem Pfad dann das richtige Cache-Verhalten zu. CloudFront Sendet den unformatierten URI-Pfad an den Ursprung, nachdem das Cache-Verhalten abgeglichen wurde. Bei fehlender Übereinstimmung werden Anforderungen stattdessen Ihrem Standard-Cacheverhalten zugeordnet.

Einige Zeichen werden normalisiert und aus dem Pfad entfernt, wie z. B. mehrere Schrägstriche (`//`) oder Punkte (`..`). Dadurch kann die URL, die CloudFront verwendet wird, so geändert werden, dass sie dem beabsichtigten Cache-Verhalten entspricht.

Example Beispiel

Sie geben die Pfade `/a/b*` und `/a*` für Ihr Cacheverhalten an.

- Ein Viewer sendet den Pfad `/a/b?c=1`, der dem Cacheverhalten `/a/b*` zugeordnet wird.
- Ein Viewer sendet den Pfad `/a/b/..?c=1`, der dem Cacheverhalten `/a*` zugeordnet wird.

Um die Normalisierung der Pfade zu umgehen, können Sie Ihre Anforderungspfade oder das Pfadmuster für das Cacheverhalten aktualisieren.

Ursprung oder Ursprungsgruppe

Diese Einstellung gilt nur, wenn Sie das Cacheverhalten einer vorhandenen Distribution erstellen oder aktualisieren.

Geben Sie den Wert eines vorhandenen Ursprungs oder einer Ursprungsgruppe ein. Dadurch wird der Ursprung oder die Ursprungsgruppe identifiziert, an die Sie Anfragen weiterleiten CloudFront möchten, wenn eine Anfrage (z. B.https://example.com/logo.jpg) dem Pfadmuster für ein Cache-Verhalten (z. B. *.jpg) oder für das Standard-Cache-Verhalten (*) entspricht.

Viewer-Protokollrichtlinien

Wählen Sie die Protokollrichtlinie aus, die Zuschauer für den Zugriff auf Ihre Inhalte an CloudFront Edge-Standorten verwenden sollen:

- HTTP und HTTPS: Viewer können beide Protokolle verwenden.
- Redirect HTTP to HTTPS: Viewer können zwar beide Protokolle verwenden, doch HTTP-Anfragen werden automatisch umgeleitet und als HTTPS-Anfragen gesendet.
- HTTPS Only: Viewer können nur auf Ihre Inhalte zugreifen, wenn sie HTTPS verwenden

Weitere Informationen finden Sie unter Erfordern Sie HTTPS für die Kommunikation zwischen Zuschauern und CloudFront.

Zulässige HTTP-Methoden

Geben Sie die HTTP-Methoden an, die Sie verarbeiten und CloudFront an Ihren Ursprung weiterleiten möchten:

- GET, HEAD: Sie können es CloudFront nur verwenden, um Objekte von Ihrem Ursprung abzurufen oder um Objekt-Header abzurufen.
- GET, HEAD, OPTIONS: Sie können CloudFront nur für den Objektabruf aus dem Ursprung, den Abruf von Objekt-Headern oder den Abruf einer Liste mit Optionen, die vom Ursprungs-Server unterstützt werden, verwenden.
- GET, HEAD, OPTIONS, PUT, POST, PATCH, DELETE: Sie können CloudFront sie verwenden, um Objekte abzurufen, hinzuzufügen, zu aktualisieren und zu löschen sowie Objekt-Header abzurufen. Darüber hinaus können Sie andere POST-Vorgänge wie das Senden von Daten aus einem Webformular ausführen.

> **Note**
>
> Wenn Sie gRPC in Ihrer Workload nutzen, müssen Sie GET, HEAD, OPTIONS, PUT, POST, PATCH, DELETE auswählen. gRPC-Workloads erfordern die `POST`-Methode. Weitere Informationen finden Sie unter gRPC mit CloudFront Distributionen verwenden. CloudFront speichert Antworten auf `HEAD` Anfragen `GET` und, optional, `OPTIONS` Anfragen im Cache. Antworten auf `OPTIONS` Anfragen werden getrennt von Antworten auf `GET` und `HEAD` Anfragen zwischengespeichert (die `OPTIONS` Methode ist im Cache-Schlüssel für

OPTIONS Anfragen enthalten). CloudFront speichert keine Antworten auf Anfragen, die andere Methoden verwenden.

Important

Wenn Sie GET, HEAD, OPTIONS oder GET, HEAD, OPTIONS, PUT, POST, PATCH, DELETE auswählen, müssen Sie möglicherweise den Zugriff auf den Amazon S3-Bucket oder den benutzerdefinierten Ursprung einschränken, um zu verhindern, dass Benutzer Operationen ausführen, die sie nicht ausführen sollen. Die folgenden Beispiele erläutern, wie Sie den Zugriff beschränken:

- Wenn Sie Amazon S3 als Quelle für Ihren Vertrieb verwenden: Erstellen Sie eine CloudFront Ursprungszugriffskontrolle, um den Zugriff auf Ihre Amazon S3 S3-Inhalte einzuschränken, und erteilen Sie der Ursprungszugriffskontrolle Berechtigungen. Wenn Sie beispielsweise so konfigurieren CloudFront , dass diese Methoden nur akzeptiert und weitergeleitet werden, weil Sie sie verwenden möchtenPUT, müssen Sie dennoch die Amazon S3 S3-Bucket-Richtlinien konfigurieren, um DELETE Anfragen angemessen zu behandeln. Weitere Informationen finden Sie unter Beschränken des Zugriffs auf einen Amazon-S3-Ursprung.
- Wenn Sie einen benutzerdefinierten Ursprungsserver verwenden: Konfigurieren Sie Ihren Ursprungsserver so, dass alle Methoden verarbeitet werden. Wenn Sie beispielsweise so konfigurieren CloudFront , dass diese Methoden nur akzeptiert und weitergeleitet werden, weil Sie sie verwenden möchtenPOST, müssen Sie Ihren Ursprungsserver trotzdem so konfigurieren, dass er DELETE Anfragen entsprechend verarbeitet.

Field-level Verschlüsselungskonfiguration

Wenn Sie die Verschlüsselung auf Feldebene für bestimmte Datenfelder erzwingen möchten, wählen Sie in den Dropdown-Listen eine Verschlüsselungskonfiguration auf Feldebene.

Weitere Informationen finden Sie unter Vertrauliche Daten durch Verschlüsselung auf Feldebene schützen.

Zwischengespeicherte HTTP-Methoden

Geben Sie an CloudFront , ob Sie die Antwort von Ihrem Ursprung zwischenspeichern möchten, wenn ein Zuschauer eine `OPTIONS` Anfrage einreicht. CloudFront speichert die Antwort auf `GET` und `HEAD` Anfragen immer im Cache.

gRPC-Anfragen zulassen über HTTP/2

Geben Sie an, ob Ihre Distribution gRPC-Anforderungen zulassen soll. Um gRPC zu aktivieren, wählen Sie die folgenden Einstellungen:

- Wählen Sie für Zulässige HTTP-Methoden die Methoden GET, HEAD, OPTIONS, PUT, POST, PATCH, DELETE aus. gRPC erfordert die `POST`-Methode.
- Markieren Sie das gRPC-Kontrollkästchen, das nach Auswahl der `POST`-Methode angezeigt wird.
- Wählen Sie bei Unterstützte HTTP-Versionen die Option HTTP/2.

Weitere Informationen finden Sie unter gRPC mit CloudFront Distributionen verwenden.

Basierend auf den ausgewählten Anforderungsheadern

Geben Sie an, ob Sie CloudFront Objekte auf der Grundlage der Werte der angegebenen Header zwischenspeichern möchten:

- Keine (verbessert das Caching) — Ihre Objekte werden CloudFront nicht auf der Grundlage von Header-Werten zwischengespeichert.
- Allowlist — CloudFront speichert Ihre Objekte nur auf der Grundlage der Werte der angegebenen Header im Cache. Verwenden Sie Allowlist Headers, um die Header auszuwählen, auf denen das Caching basieren soll. CloudFront
- Alle — Die Objekte, die mit diesem Cache-Verhalten verknüpft sind, werden CloudFront nicht zwischengespeichert. CloudFront Sendet stattdessen jede Anfrage an den Ursprung. (Nicht empfohlen für Amazon S3-Ursprünge.)

Leitet unabhängig von der ausgewählten Option bestimmte Header CloudFront an Ihren Ursprung weiter und ergreift auf der Grundlage der von Ihnen weitergeleiteten Header bestimmte Aktionen. Weitere Hinweise zur CloudFront Handhabung der Header-Weiterleitung finden Sie unter. Header und CloudFront Verhalten von HTTP-Anfragen (benutzerdefiniert und Amazon S3-Ursprünge)

Weitere Hinweise zur Konfiguration des Zwischenspeichers mithilfe CloudFront von Anforderungsheadern finden Sie unter. Zwischenspeichern von Inhalten auf der Grundlage von Anforderungsheadern

Zulassungslisten-Header

Diese Einstellungen gelten nur, wenn Sie Zulassungsliste für Cache basierend auf ausgewählten Anforderungs-Headern auswählen.

Geben Sie die Header an, die Sie beim CloudFront Zwischenspeichern Ihrer Objekte berücksichtigen möchten. Wählen Sie die Header aus der Liste der verfügbaren Header aus und klicken Sie auf Add. Um einen benutzerdefinierten Header weiterzuleiten, geben Sie den Namen des Headers in das Feld ein und wählen Sie Add Custom.

Informationen zur aktuell gültigen maximalen Anzahl von Headern, die Sie für die einzelnen Cache-Verhaltensweisen auf die Zulassungsliste setzen können, oder zum Anfordern eines höheren Kontingents (früher als Limit bezeichnet) finden Sie unter Kontingente für Header.

Zwischenspeicherung von Objekten

Wenn Ihr Original-Server Ihren Objekten einen `Cache-Control` Header hinzufügt, um zu kontrollieren, wie lange die Objekte im CloudFront Cache bleiben, und wenn Sie den `Cache-Control` Wert nicht ändern möchten, wählen Sie Origin-Cache-Header verwenden.

Um eine Mindest- und Höchstdauer anzugeben, für die Ihre Objekte unabhängig von den **`Cache-Control`** Headern im CloudFront Cache verbleiben, und eine Standardzeit, für die Ihre Objekte im CloudFront Cache verbleiben, wenn der **`Cache-Control`** Header eines Objekts fehlt, wählen Sie Anpassen. Geben Sie dann in den Feldern Minimum TTL, Default TTL und Maximum TTL den entsprechenden Wert ein.

Weitere Informationen finden Sie unter Verwalten der Dauer, die Inhalte im Cache bleiben (Ablauf).

Mindest-TTL

Geben Sie die Mindestdauer in Sekunden an, für die Objekte im CloudFront Cache verbleiben sollen, bevor eine weitere Anfrage an den Ursprung CloudFront gesendet wird, um festzustellen, ob das Objekt aktualisiert wurde.

> **Warning**
>
> Wenn Ihre Mindest-TTL größer als 0 ist, CloudFront wird Inhalt mindestens für die in der Mindest-TTL der Cache-Richtlinie angegebene Dauer zwischengespeichert, auch wenn die `private` Direktiven `Cache-Control: no-cache` `no-store`, oder in den Origin-Headern vorhanden sind.

Weitere Informationen finden Sie unter Verwalten der Dauer, die Inhalte im Cache bleiben (Ablauf).

Höchst-TTL

Geben Sie die maximale Zeit in Sekunden an, für die Objekte in CloudFront Caches verbleiben sollen, bevor Ihr Ursprung CloudFront abgefragt wird, um festzustellen, ob das Objekt aktualisiert wurde. Der Wert, den Sie für Maximum TTL angeben, gilt nur, wenn Ihr Ursprung HTTP-Header, wie beispielsweise `Cache-Control max-age`, `Cache-Control s-maxage` oder `Expires` zu Objekten hinzufügt. Weitere Informationen finden Sie unter Verwalten der Dauer, die Inhalte im Cache bleiben (Ablauf).

Um einen Wert für Maximum TTL anzugeben, müssen Sie die Option Customize für die Object Caching-Einstellung auswählen.

Der Standardwert für Maximum TTL beträgt 31 536 000 Sekunden (ein Jahr). Wenn Sie den Wert von Minimum TTL oder Default TTL in mehr als 31 536 000 Sekunden ändern, ändert sich der Standardwert von Maximum TTL in den Wert von Default TTL.

Standard-TTL

Geben Sie die Standarddauer in Sekunden an, CloudFront für die Objekte in CloudFront Caches verbleiben sollen, bevor eine weitere Anfrage an Ihren Ursprung weitergeleitet wird, um festzustellen, ob das Objekt aktualisiert wurde. Der Wert, den Sie für Standard-TTL angeben, gilt nur, wenn Ihr Ursprung keine HTTP-Header wie `Cache-Control max-age`, `Cache-Control s-maxage` oder `Expires` zu Objekten hinzufügt. Weitere Informationen finden Sie unter Verwalten der Dauer, die Inhalte im Cache bleiben (Ablauf).

Um einen Wert für Default TTL anzugeben, müssen Sie die Option Customize für die Object Caching-Einstellung auswählen.

Der Standardwert für Default TTL beträgt 86 400 Sekunden (ein Tag). Wenn Sie den Wert von Minimum TTL in mehr als 86 400 Sekunden ändern, ändert sich der Standardwert von Default TTL in den Wert von Minimum TTL.

Cookies weiterleiten

> **Note**
>
> Bei Amazon-S3-Ursprüngen gilt diese Option nur für Buckets, die als Website-Endpunkte konfiguriert sind.

Geben Sie an CloudFront , ob Sie Cookies an Ihren Ursprungsserver weiterleiten möchten und wenn ja, welche. Wenn nur ausgewählte Cookies (eine Zulassungsliste von Cookies) weitergeleitet werden sollen, geben Sie die Cookie-Namen im Feld Zulassungslisten-Cookies ein. Wenn Sie All auswählen, leitet CloudFront alle Cookies weiter, unabhängig davon, wie viele Ihre Anwendung verwendet.

Amazon S3 verarbeitet keine Cookies. Die Weiterleitung von Cookies an den Ursprung reduziert die Möglichkeit zum Caching. Wählen Sie für Cache-Verhalten, die Anforderungen an einem Amazon S3-Ursprung weiterleiten, None (Keine) in Forward Cookies (Cookies weiterleiten) aus.

Weitere Informationen zum Weiterleiten von Cookies an den Ursprung finden Sie unter Zwischenspeichern von Inhalten auf der Grundlage von Cookies.

Zulassungslisten-Cookies

> **Note**
>
> Bei Amazon S3-Ursprüngen gilt diese Option nur für Buckets, die als Website-Endpunkte konfiguriert sind.

Wenn Sie in der Liste Forward-Cookies die Option Allowlist ausgewählt haben, geben Sie im Feld Allowlist-Cookies die Namen der Cookies ein, die Sie für dieses Cache-Verhalten an Ihren Ursprungsserver weiterleiten möchten CloudFront. Geben Sie die einzelnen Cookie-Namen jeweils in einer neuen Zeile ein.

Sie können die folgenden Platzhalter verwenden, wenn Sie Cookie-Namen angeben:

- * steht für 0 oder mehr Zeichen in dem Cookie-Namen
- ? steht für genau 1 Zeichen in dem Cookie-Namen

Nehmen wir beispielsweise an, dass Viewer-Anforderungen für ein Objekt ein Cookie mit dem Namen

`userid_`*`member-number`*

Wofür jeder Ihrer Benutzer einen eindeutigen Wert hat. *`member-number`* Sie CloudFront möchten für jedes Mitglied eine separate Version des Objekts zwischenspeichern. Sie könnten dies erreichen, indem Sie alle Cookies an Ihren Ursprung weiterleiten, aber die Anfragen von Zuschauern enthalten einige Cookies, die Sie nicht zwischenspeichern CloudFront möchten. Alternativ könnten Sie den folgenden Wert als Cookie-Namen angeben, wodurch CloudFront alle Cookies, die mit beginnen, an den Ursprung weitergeleitet werden`userid_`:

`userid_*`

Informationen zur aktuell gültigen maximalen Anzahl von Cookie-Namen, die Sie für die einzelnen Cache-Verhaltensweisen auf die Zulassungsliste setzen können, oder zum Anfordern eines höheren Kontingents (früher als Limit bezeichnet) finden Sie unter Kontingente für Cookies (Legacy-Cache-Einstellungen).

Weiterleitung und Zwischenspeicherung von Abfragezeichenfolgen

CloudFront kann verschiedene Versionen Ihres Inhalts auf der Grundlage der Werte von Abfragezeichenfolgenparametern zwischenspeichern. Wählen Sie eine der folgenden Optionen:

None (Improves Caching)

> Wählen Sie diese Option aus, wenn Ihr Ursprung dieselbe Version eines Objekts unabhängig von den Werten der Abfragezeichenfolgeparameter zurückgibt. Dadurch steigt die Wahrscheinlichkeit, dass eine Anfrage aus dem Cache bedient werden CloudFront kann, wodurch die Leistung verbessert und die Belastung Ihres Quellservers verringert wird.

Alle weiterleiten, basierend auf Zulassungliste zwischenspeichern

> Wählen Sie diese Option aus, wenn Ihr Ursprungs-Server verschiedene Versionen Ihrer Objekte auf der Grundlage von einem oder mehreren Abfragezeichenfolgeparametern zurückgibt. Geben Sie dann die Parameter CloudFront an, die Sie als Grundlage für das Caching im Zulassungsliste für Abfragezeichenfolgen Feld verwenden möchten.

Forward all, cache based on all

Wählen Sie diese Option aus, wenn Ihr Ursprungsserver verschiedene Versionen Ihrer Objekte für alle Abfragezeichenfolgeparameter zurückgibt.

Weitere Informationen zur Zwischenspeicherung auf der Grundlage von Abfragezeichenfolgeparametern, einschließlich Informationen zur Verbesserung der Leistung, finden Sie unter Zwischenspeichern von Inhalten auf der Grundlage von Abfragezeichenfolgeparametern.

Zulassungsliste für Abfragezeichenfolgen

Diese Einstellung gilt nur, wenn Sie Alle weiterleiten, basierend auf Zulassungsliste zwischenspeichern für Weiterleitung und Zwischenspeicherung von Abfragezeichenfolgen auswählen. Sie können die Parameter der Abfragezeichenfolge angeben, die Sie als Grundlage für das Caching verwenden möchten CloudFront .

Smooth Streaming

Wählen Sie Yes, wenn Sie Mediendateien im Microsoft Smooth Streaming-Format verteilen möchten und keinen IIS-Server verwenden.

Wählen Sie No, wenn Sie einen Microsoft IIS-Server als Ursprung verwenden möchten, um Mediendateien im Microsoft Smooth Streaming-Format zu verteilen, oder wenn Sie keine Smooth Streaming-Mediendateien verteilen.

> **Note**
>
> Wenn Sie Yes angeben, können Sie weiterhin andere Inhalte mit diesem Cache-Verhalten verteilen, wenn diese Inhalte mit dem Wert von Path Pattern übereinstimmen.

Weitere Informationen finden Sie unter Konfigurieren von Video-on-Demand für Microsoft Smooth Streaming.

Viewerzugriff einschränken (Signierte URLs oder signierte Cookies verwenden)

Wenn Sie möchten, dass Anfragen für Objekte, die dem `PathPattern` für dieses Cache-Verhalten entsprechen, öffentliche URLs verwenden, wählen Sie No aus.

Wenn Sie möchten, dass Anfragen für Objekte, die dem `PathPattern` für dieses Cache-Verhalten entsprechen, signierte URLs verwenden, wählen Sie Yes aus. Geben Sie dann die AWS Konten an, die Sie zum Erstellen signierter URLs verwenden möchten. Diese Konten werden als vertrauenswürdige Unterzeichner bezeichnet.

Weitere Informationen zu vertrauenswürdigen Ausstellern finden Sie unter Festlegen der Signaturgeber, die signierte URLs und signierte Cookies erstellen können.

Vertrauenswürdiger Signaturgeber

Diese Einstellung gilt nur, wenn Sie Ja für Viewer-Zugriff einschränken (signierte URLs oder signierte Cookies verwenden) auswählen.

Wählen Sie aus, welche AWS Konten Sie als vertrauenswürdige Unterzeichner für dieses Cache-Verhalten verwenden möchten:

- Selbst: Verwende das Konto, mit dem du derzeit AWS-Managementkonsole als vertrauenswürdiger Unterzeichner angemeldet bist. Wenn Sie derzeit als IAM-Benutzer angemeldet sind, wird das zugehörige AWS Konto als vertrauenswürdiger Unterzeichner hinzugefügt.
- Wenn Sie das Kontrollkästchen Specify Accounts aktiviert haben, geben Sie die Konto-IDs der vertrauenswürdigen Signaturgeber im Feld AWS Account Numbers ein.

Um signierte URLs zu erstellen, muss ein AWS Konto über mindestens ein aktives CloudFront key pair verfügen.

> **Important**
>
> Wenn Sie eine Verteilung aktualisieren, die Sie bereits zum Verteilen von Inhalten verwenden, fügen Sie vertrauenswürdige Aussteller erst hinzu, wenn Sie bereit sind, signierte URLs für Ihre Objekte zu generieren. Wenn Sie vertrauenswürdige Aussteller zu einer Verteilung hinzugefügt haben, müssen Benutzer für den Zugriff auf Objekte, die dem `PathPattern` für dieses Cache-Verhalten entsprechen, signierte URLs verwenden.

AWS-Konto Ziffern

Diese Einstellung gilt nur, wenn Sie Konten angeben für Vertrauenswürdige Signaturgeber auswählen.

Wenn Sie signierte URLs AWS-Konten zusätzlich zum oder anstelle des aktuellen Kontos erstellen möchten, geben Sie eine AWS-Konto Zahl pro Zeile in dieses Feld ein. Beachten Sie Folgendes:

- Die angegebenen Konten müssen über mindestens ein aktives CloudFront-Schlüsselpaar verfügen. Weitere Informationen finden Sie unter Erstellen von Schlüsselpaaren für Ihre Signaturgeber.
- Sie können keine CloudFront Schlüsselpaare für IAM-Benutzer erstellen, sodass Sie IAM-Benutzer nicht als vertrauenswürdige Unterzeichner verwenden können.
- Informationen zum Abrufen der AWS-Konto Nummer für ein Konto finden Sie im Management-Referenzhandbuch unter AWS-Konto Identifikatoren anzeigen.AWS-Konto
- Wenn Sie die Kontonummer für das Girokonto eingeben, CloudFront wird automatisch das Kontrollkästchen Selbst aktiviert und die Kontonummer wird aus der AWS Kontonummernliste entfernt.

Objekte automatisch komprimieren

Wenn Sie Dateien bestimmter Typen automatisch komprimieren CloudFront möchten, wenn Zuschauer komprimierte Inhalte unterstützen, wählen Sie Ja. Wenn CloudFront Ihre Inhalte komprimiert, werden Downloads schneller ausgeführt, da die Dateien kleiner sind, und Ihre Webseiten werden schneller für Ihre Benutzer wiedergegeben. Weitere Informationen finden Sie unter Bereitstellen von komprimierten Dateien.

CloudFront Ereignis

Diese Einstellung gilt für Lambda-Funktionszuordnungen.

Sie können wählen, ob eine Lambda-Funktion ausgeführt werden soll, wenn eines oder mehrere der folgenden CloudFront Ereignisse eintreten:

- Wann CloudFront erhält er eine Anfrage von einem Zuschauer (Zuschaueranfrage)
- Bevor CloudFront eine Anfrage an den Ursprung weitergeleitet wird (ursprüngliche Anfrage)
- Wann CloudFront erhält er eine Antwort vom Ursprung (ursprüngliche Antwort)
- Before CloudFront gibt die Antwort an den Zuschauer zurück (Antwort des Betrachters)

Weitere Informationen finden Sie unter Auswählen des Ereignisses zur Auslösung der Funktion.

ARN der Lambda-Funktion

Diese Einstellung gilt für Lambda-Funktionszuordnungen.

Geben Sie den Amazon-Ressourcennamen (ARN) der Lambda-Funktion an, für die Sie einen Auslöser hinzufügen möchten. Informationen zum Abrufen des ARN für eine Funktion finden Sie in Schritt 1 des Verfahrens Hinzufügen von Triggern mithilfe der CloudFront Konsole.

Include body (Text einschließen)

Diese Einstellung gilt für Lambda-Funktionszuordnungen.

Weitere Informationen finden Sie unter Text einschließen.

Distribution Settings (Einstellungen für die Verteilung)

Die folgenden Werte gelten für die gesamte Verteilung.

Themen

- Preisklasse
- AWS WAF Web-ACL
- Alternative Domänennamen (CNAMEs)
- SSL-Zertifikat
- Benutzerdefinierte SSL-Clientunterstützung
- Sicherheitsrichtlinie (SSL/TLS Mindestversion)
- Unterstützte HTTP-Versionen
- Standardstammobjekt
- Standardprotokollierung
- Verbindungsprotokolle.
- Protokollpräfix
- Protokollierung von Cookies
- Aktivieren von IPv6 (Viewer-Anforderungen)
- Gegenseitige Authentifizierung
- Aktivieren von IPv6 für benutzerdefinierte Ursprünge (Ursprungsanforderungen)

- Comment
- Status der Verteilung

Preisklasse

Wählen Sie die Preisklasse, die dem Höchstpreis entspricht, den Sie für die CloudFront Dienstleistung zahlen möchten. Standardmäßig werden Ihre Objekte von Randstandorten in allen CloudFront Regionen aus CloudFront bedient.

Weitere Informationen zu Preisklassen und dazu, wie sich Ihre Wahl der Preisklasse auf die CloudFront Leistung Ihres Vertriebs auswirkt, finden Sie unter CloudFront Preisgestaltung.

AWS WAF Web-ACL

Sie können Ihre CloudFront Distribution mit einer Firewall für Webanwendungen schützen AWS WAF, mit der Sie Ihre Webanwendungen und APIs so schützen können, dass Anfragen blockiert werden, bevor sie Ihre Server erreichen. Dies ist möglichAktivieren AWS WAF für Distributionen, wenn Sie eine CloudFront Distribution erstellen oder bearbeiten.

Optional können Sie später in der AWS WAF Konsole unter zusätzliche Sicherheitsvorkehrungen für andere anwendungsspezifische Bedrohungen konfigurieren. https://console.aws.amazon.com/wafv2/

Weitere Informationen AWS WAF dazu finden Sie im AWS WAF Entwicklerhandbuch.

Alternative Domänennamen (CNAMEs)

Optional. Geben Sie einen oder mehrere Domainnamen an, die Sie für URLs für Ihre Objekte verwenden möchten, und nicht den Domainnamen, den Sie bei der Erstellung Ihrer Distribution CloudFront zugewiesen haben. Sie müssen Eigentümer des Domainnamens sein oder über die Autorisierung verfügen, ihn zu verwenden. Dies überprüfen Sie, indem Sie ein SSL/TLS Zertifikat hinzufügen.

Wenn beispielsweise die URL für das Objekt:

```
/images/image.jpg
```

wie folgt angezeigt werden soll:

```
https://www.example.com/images/image.jpg
```

und nicht wie folgt:

`https://d111111abcdef8.cloudfront.net/images/image.jpg`

Fügen Sie einen CNAME für `www.example.com` hinzu.

> **Important**
>
> Wenn Sie einen CNAME für `www.example.com` zu Ihrer Verteilung hinzufügen, müssen Sie auch die folgenden Schritte ausführen:
>
> - Erstellen (oder aktualisieren) Sie einen CNAME-Datensatz für Ihren DNS-Service, um Abfragen für `www.example.com` an `d111111abcdef8.cloudfront.net` weiterzuleiten.
> - Fügen Sie ein Zertifikat CloudFront von einer vertrauenswürdigen Zertifizierungsstelle (CA) hinzu, das den Domainnamen (CNAME) abdeckt, den Sie Ihrer Distribution hinzufügen, um Ihre Autorisierung zur Verwendung des Domainnamens zu überprüfen.
>
> Sie müssen die Berechtigung besitzen, beim DNS-Dienstanbieter für die Domäne einen CNAME-Datensatz zu erstellen. Das bedeutet in der Regel, dass Sie der Besitzer der Domäne sind oder Anwendungen für den Besitzer der Domäne entwickeln.

Informationen zur aktuell gültigen maximalen Anzahl von alternativen Domainnamen, die Sie einer Distribution hinzufügen können, oder zum Anfordern eines höheren Kontingents (früher als Limit bezeichnet) finden Sie unter Allgemeine Kontingente für Verteilungen.

Weitere Informationen zu alternativen Domänennamen finden Sie unter Verwenden von benutzerdefinierten URLs durch Hinzufügen von alternativen Domainnamen (CNAMEs). Weitere Hinweise zu CloudFront URLs finden Sie unterPassen Sie das URL-Format für Dateien in an CloudFront.

SSL-Zertifikat

Wenn Sie einen alternativen Domänennamen angegeben haben, der für Ihre Verteilung verwendet werden soll, wählen Sie Custom SSL Certificate (Benutzerdefiniertes SSL-Zertifikat) und anschließend ein Zertifikat aus, das diesen Domänennamen abdeckt, um Ihre Berechtigung zur Verwendung des alternativen Domänennamens zu bestätigen. Wenn Sie möchten, dass Viewer HTTPS für den Zugriff auf Ihre Objekte verwenden, wählen Sie die entsprechende Einstellung aus.

- CloudFront Standardzertifikat (*.cloudfront.net) — Wählen Sie diese Option, wenn Sie den CloudFront Domainnamen in den URLs für Ihre Objekte verwenden möchten, z. B. `https://d111111abcdef8.cloudfront.net/image1.jpg`
- Benutzerdefiniertes SSL-Zertifikate – Wählen Sie diese Option aus, wenn Sie in den URLs für Ihre Objekte Ihren eigenen Domänennamen als alternativen Domänennamen verwenden möchten, z. B. `https://example.com/image1.jpg`. Wählen Sie anschließend ein Zertifikat aus, das den alternativen Domänennamen abdeckt. Die Liste der Zertifikate kann folgende Arten von Zertifikaten enthalten:
  - Zertifikate bereitgestellt von AWS Certificate Manager
  - Zertifikate, die Sie von einer externen Zertifizierungsstelle erworben und zu ACM hochgeladen haben
  - Zertifikate, die Sie von einer externen Zertifizierungsstelle erworben und zum IAM-Zertifikatspeicher hochgeladen haben

  Bei Auswahl dieser Einstellung sollten Sie nur einen alternativen Domänennamen in Ihren Objekt-URLs () verwen (https://example.com/logo.jpg). Wenn Sie Ihren CloudFront Distributionsdomänennamen (https://d111111abcdef8.cloudfront.net/logo.jpg) verwenden und ein Client einen älteren Viewer verwendet, der SNI nicht unterstützt, hängt die Reaktion des Viewers davon ab, welchen Wert Sie für Unterstützte Clients wählen:
  - Alle Clients: Der Viewer zeigt eine Warnung an, da der CloudFront Domainname nicht mit dem Domainnamen in Ihrem SSL/TLS Zertifikat übereinstimmt.
  - Nur Clients, die Server Name Indication (SNI) Support: CloudFront unterbricht die Verbindung mit dem Viewer, ohne das Objekt zurückzugeben.

Benutzerdefinierte SSL-Clientunterstützung

Gilt nur, wenn Sie Benutzerdefiniertes SSL-Zertifikat (example.com) unter SSL-Zertifikat auswählen. Wenn Sie einen oder mehrere alternative Domainnamen und ein benutzerdefiniertes SSL-Zertifikat für die Verteilung angegeben haben, wählen Sie aus, wie Sie HTTPS-Anfragen bearbeiten CloudFront möchten:

- Clients that Support Server Name Indication (SNI) - (Recommended) (Clients, die SNI (Server Name Indication) unterstützen – (Empfohlen)): Mit dieser Einstellung können nahezu alle modernen Webbrowser und Clients eine Verbindung zur Verteilung herstellen, da sie SNI unterstützen. Einige Viewer verwenden jedoch möglicherweise ältere Webbrowser oder Clients, die SNI nicht unterstützen, was bedeutet, dass sie keine Verbindung zur Verteilung herstellen können.

Um diese Einstellung mithilfe der CloudFront API anzuwenden, geben Sie dies `sni-only` in dem `SSLSupportMethod` Feld an. In CloudFormation wird das Feld `SslSupportMethod` genannt (Beachten Sie die geänderte Groß-/Kleinschreibung).

- Unterstützung für Legacy-Clients: Mit dieser Einstellung können ältere Webbrowser und Clients, die SNI nicht unterstützen, eine Verbindung zur Distribution herstellen. Für diese Einstellung fallen jedoch zusätzliche monatliche Gebühren an. Den genauen Preis finden Sie auf der CloudFront Amazon-Preisseite und suchen Sie auf der Seite nach Dedicated IP Custom SSL.

  Um diese Einstellung mithilfe der CloudFront API anzuwenden, geben Sie dies `vip` in das `SSLSupportMethod` Feld ein. In ist CloudFormation das Feld benannt `SslSupportMethod` (beachten Sie die unterschiedliche Groß-/Kleinschreibung).

Weitere Informationen finden Sie unter Wählen Sie aus, wie CloudFront HTTPS-Anfragen verarbeitet werden.

Sicherheitsrichtlinie (SSL/TLS Mindestversion)

Geben Sie die Sicherheitsrichtlinie CloudFront an, die Sie für HTTPS-Verbindungen mit Zuschauern (Clients) verwenden möchten. Eine Sicherheitsrichtlinie bestimmt zwei Einstellungen:

- Das SSL/TLS Mindestprotokoll, das für die Kommunikation mit Zuschauern CloudFront verwendet wird.
- Die Chiffren, mit denen der Inhalt verschlüsselt werden CloudFront kann, der an die Zuschauer zurückgegeben wird.

Weitere Informationen zu den Sicherheitsrichtlinien einschließlich der jeweils enthaltenen Protokolle und Verschlüsselungen finden Sie unter Unterstützte Protokolle und Chiffren zwischen Viewern und CloudFront.

Die verfügbaren Sicherheitsrichtlinien hängen von den Werten ab, die Sie für SSL-Zertifikate und benutzerdefinierte SSL-Clientunterstützung (bekannt als `CloudFrontDefaultCertificate` und `SSLSupportMethod` in der CloudFront API) angeben:

- Wenn das SSL-Zertifikat das CloudFront Standardzertifikat (*.cloudfront.net) ist (wenn `CloudFrontDefaultCertificate` es `true` in der API enthalten ist), wird die Sicherheitsrichtlinie CloudFront automatisch auf TLSv1 festgelegt.

- Wenn SSL-Zertifikat auf Benutzerdefiniertes SSL-Zertifikat (example.com) und Benutzerdefinierte SSL-Client-Unterstützung auf Clients, die Server Name Indication (SNI) unterstützen – (empfohlen) eingestellt ist (wenn `CloudFrontDefaultCertificate` in der API auf `false` und `SSLSupportMethod` auf `sni-only` festgelegt ist), können Sie aus den folgenden Sicherheitsrichtlinien auswählen:
  - TLSv1.3_2025
  - TLSv1.2_2025
  - TLSv1.2_2021
  - TLSv1.2_2019
  - TLSv1.2_2018
  - TLSv1.1_2016
  - TLSv1_2016
  - TLSv1
- Wenn SSL-Zertifikat auf Benutzerdefiniertes SSL-Zertifikat (example.com) und Benutzerdefinierte SSL-Client-Unterstützung auf Unterstützung für Legacy-Clients eingestellt ist (wenn `CloudFrontDefaultCertificate` in der API auf `false` und `SSLSupportMethod` auf `vip` festgelegt ist), können Sie aus den folgenden Sicherheitsrichtlinien auswählen:
  - TLSv1
  - SSLv3

  In dieser Konfiguration sind die Sicherheitsrichtlinien TLSv1.3 _2025, TLSv1.2 _2025, TLSv1.2 _2021, TLSv1.2 _2019, TLSv1.2 _2018, TLSv1.1 _2016 und TLSv1_2016 nicht in der Konsole oder API verfügbar. CloudFront Wenn Sie eine dieser Sicherheitsrichtlinien verwenden möchten, stehen Ihnen folgende Optionen zur Verfügung:
  - Prüfen Sie, ob Ihre Verteilung Unterstützung für Legacy-Clients mit dedizierten IP-Adressen benötigt. Wenn Ihre Viewer Server Name Indication (SNI) unterstützen, empfehlen wir, die Einstellung Benutzerdefinierte SSL-Client-Unterstützung Ihrer Distribution auf Clients, die Server Name Indication (SNI) unterstützen zu aktualisieren (`SSLSupportMethod` in der API auf `sni-only` einzustellen). Auf diese Weise können Sie alle verfügbaren TLS-Sicherheitsrichtlinien verwenden und Ihre Gebühren können gesenkt werden. CloudFront
  - Wenn Sie Legacy Clients Support mit dedizierten IP-Adressen beibehalten müssen, können Sie eine der anderen TLS-Sicherheitsrichtlinien (TLSv1.3_2025, _2025, TLSv1.2 _2021, TLSv1.2 TLSv1.2 _2019, TLSv1.2 _2018, _2016 oder TLSv1.1 TLSv1_2016) anfordern, indem Sie im Support Center einen Fall erstellen.AWS

> **Note**
>
> Bevor Sie sich an den AWS Support wenden, um diese Änderung zu beantragen, sollten Sie Folgendes beachten:
>
> - Wenn Sie eine dieser Sicherheitsrichtlinien (TLSv1.3_2025, _2025, TLSv1.2 _2021, TLSv1.2 TLSv1.2 _2019, TLSv1.2 _2018, _2016 oder TLSv1.1 TLSv1_2016) zu einer Legacy Clients Support-Distribution hinzufügen, wird die Sicherheitsrichtlinie auf alle Nicht-SNI-Viewer-Anfragen für alle Legacy Clients Support-Distributionen in Ihrem Konto angewendet. AWS Wenn Viewer jedoch SNI-Anforderungen an eine Verteilung mit Unterstützung für Legacy-Clients senden, gilt die Sicherheitsrichtlinie dieser Verteilung. Um sicherzustellen, dass Ihre gewünschte Sicherheitsrichtlinie auf alle Zuschaueranfragen angewendet wird, die an alle Legacy Clients Support-Distributionen in Ihrem AWS Konto gesendet werden, fügen Sie die gewünschte Sicherheitsrichtlinie jeder Distribution einzeln hinzu.
> - Per Definition unterstützt die neue Sicherheitsrichtlinie nicht dieselben Verschlüsselungen und Protokolle wie die alte. Wenn Sie sich beispielsweise dafür entscheiden, die Sicherheitsrichtlinie einer Distribution von TLSv1 auf TLSv1.1 _2016 zu aktualisieren, unterstützt diese Distribution die Verschlüsselung nicht mehr. DES-CBC3-SHA Weitere Informationen zu den Verschlüsselungen und Protokollen, die von den einzelnen Sicherheitsrichtlinien unterstützt werden, finden Sie unter Unterstützte Protokolle und Chiffren zwischen Viewern und CloudFront.

Unterstützte HTTP-Versionen

Wählen Sie die HTTP-Versionen aus, die Ihre Distribution unterstützen soll, wenn Zuschauer mit ihnen kommunizieren CloudFront.

Damit Zuschauer CloudFront sie verwenden können HTTP/2, müssen Zuschauer Server Name Indication (SNI) TLSv1.2 oder höher unterstützen.

Für Zuschauer und CloudFront zur Nutzung HTTP/3 müssen Zuschauer Server Name Indication (SNI) unterstützen TLSv1.3 . CloudFront unterstützt die HTTP/3 Verbindungsmigration, sodass der Zuschauer zwischen Netzwerken wechseln kann, ohne die Verbindung zu verlieren. Weitere Angaben zur Verbindungsmigration finden Sie in den Informationen zur Verbindungsmigration in RFC 9000.

> **Note**
>
> Weitere Hinweise zu unterstützten TLSv1.3 Verschlüsselungen finden Sie unter. Unterstützte Protokolle und Chiffren zwischen Viewern und CloudFront

> **Note**
>
> Wenn Sie Amazon Route 53 verwenden, können Sie anhand von HTTPS-Einträgen Protokollaushandlungen als Teil der DNS-Suche ermöglichen, sofern der Client dies unterstützt. Weitere Informationen finden Sie unter Create alias resource record set.

Standardstammobjekt

Optional. Das Objekt, das Sie von Ihrem Ursprungsserver anfordern CloudFront möchten (z. B.`index.html`), wenn ein Betrachter die Stamm-URL Ihrer Distribution (`https://www.example.com/`) anstelle eines Objekts in Ihrer Distribution (`https://www.example.com/product-description.html`) anfordert. Durch die Festlegung eines Angeben eines Standardstammobjekt wird vermieden, dass die Inhalte Ihrer Verteilung preisgegeben werden.

Die maximale Länge des Namens beträgt 255 Zeichen. Der Name kann folgende Zeichen enthalten:

- A-Z, a-z
- 0-9
- _ - . * $ / ~ " '
- & (übergeben und zurückgegeben als `&`)

Geben Sie bei der Angabe des Standardstammobjekts nur den Objektnamen ein, z. B. `index.html`. Fügen Sie keinen / vor dem Objektnamen hinzu.

Weitere Informationen finden Sie unter Angeben eines Standardstammobjekts.

Standardprotokollierung

Geben Sie an CloudFront , ob Sie Informationen zu jeder Anforderung für ein Objekt protokollieren und die Protokolldateien speichern möchten. Sie können die Protokollierung jederzeit aktivieren

oder deaktivieren. Es fallen keine zusätzlichen Kosten für die Aktivierung der Protokollierung an. Sie müssen jedoch mit Gebühren für das Speichern von Daten und den Zugriff darauf rechnen. Sie können die Protokolle jederzeit löschen.

CloudFront unterstützt die folgenden Standard-Protokollierungsoptionen:

- Standardprotokollierung (v2) — Sie können Protokolle an Lieferziele wie Amazon CloudWatch Logs, Amazon Data Firehose und Amazon Simple Storage Service (Amazon S3) senden.
- Standardprotokollierung (Legacy) – Sie können Protokolle nur an einen Amazon-S3-Bucket senden.

Verbindungsprotokolle.

Wenn Sie die gegenseitige Authentifizierung für Ihre Distribution aktivieren, werden Verbindungsprotokolle CloudFront bereitgestellt, die Attribute der an Ihre Distributionen gesendeten Anfragen aufzeichnen. Verbindungsprotokolle enthalten Informationen wie die Client-IP-Adresse und den Port, Informationen zum Client-Zertifikat, die Verbindungsergebnisse und die verwendeten TLS-Chiffren. Diese Verbindungsprotokolle können dann verwendet werden, um Anforderungsmuster und andere Trends zu überprüfen.

Weitere Informationen zu Verbindungsprotokollen finden Sie unterBeobachtbarkeit mithilfe von Verbindungsprotokollen.

Protokollpräfix

(Optional) Wenn Sie die Standardprotokollierung (Legacy) aktivieren, geben Sie gegebenenfalls die Zeichenfolge CloudFront an, der Sie den Namen der Zugriffsprotokolldateien für diese Verteilung voranstellen möchten, z. B.`exampleprefix/`. Der abschließende Schrägstrich (/) ist optional, jedoch empfohlen, um das Durchsuchen Ihrer Protokolldateien zu vereinfachen. Weitere Informationen finden Sie unter Konfigurieren der Standardprotokollierung (Legacy).

Protokollierung von Cookies

Wenn Sie Cookies in CloudFront die Zugriffsprotokolle aufnehmen möchten, wählen Sie On. Wenn Sie Cookies in die Protokolle aufnehmen möchten, werden alle Cookies unabhängig davon CloudFront protokolliert, wie Sie das Cache-Verhalten für diese Verteilung konfigurieren: alle Cookies weiterleiten, keine Cookies weiterleiten oder eine bestimmte Liste von Cookies an den Ursprung weiterleiten.

Amazon S3 verarbeitet keine Cookies. Wenn Ihre Verteilung keinen Amazon EC2- oder einen anderen benutzerdefinierten Ursprung enthält, sollten Sie daher Off (Aus) als Wert in Cookie Logging (Cookie-Protokollierung) auswählen.

Weitere Informationen zu Cookies finden Sie unter Zwischenspeichern von Inhalten auf der Grundlage von Cookies.

Aktivieren von IPv6 (Viewer-Anforderungen)

Wenn Sie CloudFront auf Zuschaueranfragen von IPv4- und IPv6-IP-Adressen antworten möchten, wählen Sie IPv6 aktivieren. Weitere Informationen finden Sie unter IPv6 für CloudFront Distributionen aktivieren.

Gegenseitige Authentifizierung

Optional. Sie können sich dafür entscheiden, die gegenseitige Authentifizierung für Ihre Distribution zu aktivieren. CloudFront Weitere Informationen finden Sie unter Gegenseitige TLS-Authentifizierung mit CloudFront (Viewer mTLS).

Aktivieren von IPv6 für benutzerdefinierte Ursprünge (Ursprungsanforderungen)

Wenn Sie einen benutzerdefinierten Ursprung verwenden (mit Ausnahme von Amazon S3- und VPC-Ursprüngen), können Sie die Ursprungseinstellungen für Ihre Distribution anpassen, um auszuwählen, wie über IPv4- oder IPv6-Adressen CloudFront eine Verbindung zu Ihrem Ursprung hergestellt wird. Weitere Informationen finden Sie unter IPv6 für CloudFront Distributionen aktivieren.

Comment

Optional. Wenn Sie eine Verteilung erstellen, können Sie einen Kommentar mit einer Länge von bis zu 128 Zeichen einfügen. Sie können den Kommentar jederzeit aktualisieren.

Status der Verteilung

Gibt an, ob die Distribution nach der Bereitstellung aktiviert oder deaktiviert werden soll:

- Aktiviert bedeutet, dass Sie sofort Links mit dem Domainnamen der Distribution verwenden können und dass Benutzer die Inhalte darüber abrufen können, sobald die Distribution vollständig bereitsteht. Wenn eine Verteilung aktiviert ist, akzeptiert und verarbeitet CloudFront alle Anfragen zu diesen Inhalten von Endbenutzern, die den Domänennamen verwenden, welcher der Verteilung zugewiesen wurde.

Wenn Sie eine CloudFront Verteilung erstellen, ändern oder löschen, dauert es einige Zeit, bis Ihre Änderungen in der CloudFront Datenbank übernommen werden. Bei einer unmittelbaren Anfrage für Informationen zu einer Verteilung wird die Änderung möglicherweise nicht angezeigt. Die Weiterleitung wird in der Regel innerhalb weniger Minuten abgeschlossen. Dieser Zeitraum kann sich bei einer hohen Zeitauslastung oder Netzwerkpartition jedoch verlängern.

- Disabled bedeutet, dass die Verteilung möglicherweise bereitgestellt und betriebsbereit ist, aber Benutzer sie noch nicht verwenden können. Immer wenn eine Verteilung deaktiviert ist, akzeptiert CloudFront sie keine Anfragen von Endbenutzern, die den mit dieser Verteilung verknüpften Domainnamen verwenden. Die Verteilung kann erst verwendet werden, wenn Sie den Status von Deaktiviert zu Aktiviert ändern (indem Sie die Konfiguration der Verteilung aktualisieren).

Sie können in Bezug auf den Status einer Verteilung so oft zwischen Deaktiviert und Aktiviert wechseln, wie Sie möchten. Führen Sie die Schritte zum Aktualisieren der Konfiguration einer Verteilung aus. Weitere Informationen finden Sie unter Eine Verteilung aktualisieren.

Benutzerdefinierte Fehlerseiten und Zwischenspeicherung von Fehlern

Sie können ein Objekt an den Viewer CloudFront zurücksenden lassen (z. B. eine HTML-Datei), wenn Ihr Amazon S3- oder benutzerdefinierter Origin einen HTTP-Statuscode 4xx oder 5xx-an zurückgibt. CloudFront Sie können auch angeben, wie lange eine Fehlerantwort von Ihrem Absender oder einer benutzerdefinierten Fehlerseite in CloudFront Edge-Caches zwischengespeichert wird. Weitere Informationen finden Sie unter Erstellen einer benutzerdefinierten Fehlerseite für bestimmte HTTP-Statuscodes.

> **Note**
>
> Die folgenden Werte sind im Assistenten zum Erstellen von Verteilungen nicht enthalten, deshalb können Sie benutzerdefinierte Fehlerseiten nur konfigurieren, wenn Sie eine Verteilung aktualisieren.

Themen

- HTTP-Fehlercode
- Pfad zur Antwortseite
- HTTP-Antwortcode
- Mindest-TTL für die Zwischenspeicherung von Fehlern (Sekunden)

HTTP-Fehlercode

Der HTTP-Statuscode, für den Sie eine benutzerdefinierte Fehlerseite zurückgeben CloudFront möchten. Sie können so konfigurieren CloudFront , dass benutzerdefinierte Fehlerseiten für keinen, einige oder alle CloudFront zwischengespeicherten HTTP-Statuscodes zurückgegeben werden.

Pfad zur Antwortseite

Der Pfad zu der benutzerdefinierten Fehlerseite (z. B. `/4xx-errors/403-forbidden.html`), die CloudFront an Betrachter zurückgeben soll, wenn Ihr Ursprung den HTTP-Statuscode zurückgibt, den Sie für Error Code angegeben haben (z. B. 403). Wenn Sie Ihre Objekte und Ihre benutzerdefinierten Fehlerseiten an verschiedenen Orten speichern möchten, muss Ihre Verteilung ein Cache-Verhalten mit den folgenden Eigenschaften enthalten:

- Der Wert von Path Pattern stimmt mit dem Pfad zu Ihren benutzerdefinierten Fehlermeldungen überein. Angenommen, Sie haben benutzerdefinierte Fehlerseiten für 4xx-Fehler in einem Amazon S3-Bucket in einem Verzeichnis mit dem Namen gespeicher `/4xx-errors`. Ihre Verteilung muss ein Cache-Verhalten beinhalten, für welches das Pfadmuster Anfragen für Ihre benutzerdefinierten Fehlerseiten an diesen Ort weiterleitet, z. B. /4xx-errors/*.
- Der Wert von Origin legt den Wert von Origin ID für den Ursprung fest, der Ihre benutzerdefinierten Fehlerseiten enthält.

HTTP-Antwortcode

Der HTTP-Statuscode, den Sie zusammen mit der benutzerdefinierten Fehlerseite an den Viewer zurückgeben möchten CloudFront .

Mindest-TTL für die Zwischenspeicherung von Fehlern (Sekunden)

Die Mindestdauer, für die Sie Fehlerantworten von Ihrem Ursprungsserver zwischenspeichern möchten CloudFront .

Geografische Einschränkungen

Wenn Sie verhindern möchten, dass Benutzer in ausgewählten Ländern auf Ihre Inhalte zugreifen, können Sie Ihre CloudFront Distribution mit einer Zulassungsliste oder einer Sperrliste konfigurieren. Für die Konfiguration von geografischen Einschränkungen fallen keine zusätzlichen Kosten an. Weitere Informationen finden Sie unter Einschränken der geografischen Distribution Ihrer Inhalte.

Testen einer Distribution

Nachdem Sie Ihre Distribution erstellt haben, CloudFront weiß er, wo sich Ihr Ursprungsserver befindet, und Sie kennen den Domainnamen, der mit der Distribution verknüpft ist. Um Ihre Distribution zu testen, gehen Sie folgendermaßen vor:

1. Warten Sie, bis Ihre Distribution bereitgestellt ist.
   - Sehen Sie sich die Details Ihrer Distribution in der Konsole an. Wenn die Bereitstellung Ihrer Distribution abgeschlossen ist, ändert sich das Feld Letzte Änderung von Wird bereitgestellt in ein Datum und eine Uhrzeit.
2. Verwenden Sie das folgende Verfahren, um Links zu Ihren Objekten mit dem CloudFront Domainnamen zu erstellen.
3. Testen Sie die Links. CloudFront stellt die Objekte Ihrer Webseite oder Anwendung zur Verfügung.

Erstellen von Links zu Ihren Objekten

Gehen Sie wie folgt vor, um Testlinks für die Objekte in Ihrer CloudFront Webdistribution zu erstellen.

So erstellen Sie Links zu Objekten in einer Web-Verteilung

1. Kopieren Sie den folgenden HTML-Code in eine neue Datei, *domain-name* ersetzen Sie ihn durch den Domainnamen Ihrer Distribution und *object-name* ersetzen Sie ihn durch den Namen Ihres Objekts.

   ```
   <html>
   <head>
    <title>My CloudFront Test</title>
   </head>
   <body>
    <p>My text content goes here.</p>
    <p><img src="https://domain-name/object-name" alt="my test image"></p>
   </body>
   </html>
   ```

   Wenn Ihr Domänenname beispielsweise `d111111abcdef8.cloudfront.net` ist und Ihr Objekt `image.jpg`, würde die URL für den Link folgendermaßen lauten:

   `https://d111111abcdef8.cloudfront.net/image.jpg`.

Wenn sich das Objekt in einem Ordner auf Ihrem Ursprungsserver befindet, muss der Ordner ebenfalls in der URL enthalten sein. Wenn sich image.jpg beispielsweise im Ordner „images” auf Ihrem Ursprungsserver befindet, würde die URL folgendermaßen lauten:

```
https://d111111abcdef8.cloudfront.net/images/image.jpg
```

2. Speichern Sie den HTML-Code unter einem Dateinamen mit der Erweiterung .html.
3. Öffnen Sie Ihre Webseite in einem Browser, um sicherzustellen, dass Sie Ihr Objekt sehen können.

Der Browser gibt Ihre Seite mit der eingebetteten Bilddatei zurück, die von der Edge-Position aus bereitgestellt wurde, CloudFront die für das Objekt als geeignet erachtet wurde.

Eine Verteilung aktualisieren

In der CloudFront Konsole können Sie die CloudFront Distributionen sehen, die mit Ihrer Distribution verknüpft sind AWS-Konto, die Einstellungen für eine Distribution einsehen und die meisten Einstellungen aktualisieren. Beachten Sie, dass die von Ihnen vorgenommenen Einstellungsänderungen erst wirksam werden, nachdem die Verteilung an die AWS -Edge-Standorte weitergegeben wurde.

Aktualisieren einer Distribution in der Konsole

Die folgenden Verfahren zeigen Ihnen, wie Sie eine CloudFront Distribution in der Konsole aktualisieren.

Multi-tenant

So aktualisieren Sie eine Multi-Tenant-Distribution

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home.
2. Suchen Sie nach der ID der Multi-Tenant-Distribution und wählen Sie sie aus.
3. Wählen Sie die Registerkarte der Einstellungen aus, die Sie aktualisieren möchten.
4. Nehmen Sie die Aktualisierungen vor und wählen Sie zum Speichern Änderungen speichern aus. Weitere Informationen zu den aktualisierbaren Einstellungen finden Sie unter Referenz für vorkonfigurierte Distributionseinstellungen.

Sie können eine Distribution auch mithilfe der CloudFront API aktualisieren:

- Informationen zum Aktualisieren einer Distribution finden Sie UpdateDistributionin der Amazon CloudFront API-Referenz.

> **Important**
>
> Beachten Sie, dass bei der Aktualisierung Ihrer Distribution eine Reihe von zusätzlichen Feldern erforderlich sind, die bei der Erstellung der Distribution nicht erforderlich waren. Um sicherzustellen, dass alle erforderlichen Felder enthalten sind, wenn Sie die CloudFront API zum Aktualisieren einer Distribution verwenden, folgen Sie den UpdateDistributionin der Amazon CloudFront API-Referenz beschriebenen Schritten.

Um die Multi-Tenant-Distribution für einen Distributionsmandanten zu ändern, müssen Sie den Distributionsmandanten aktualisieren. Sie müssen ihn auch aktualisieren, um dessen Domain, Zertifikat, Anpassungen oder Parameterwerte zu ändern. Weitere Informationen zur Aktualisierung des Zertifikats für den Distributionsmandanten finden Sie unter Hinzufügen einer Domain und eines Zertifikats (Distributionsmandant).

So aktualisieren Sie einen Distributionsmandanten

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home.
2. Wählen Sie unter SaaS die Option Distributionsmandanten aus.
3. Suchen Sie nach dem Distributionsmandanten. Verwenden Sie das Dropdown-Menü in der Suchleiste, um nach Domain, Name, Distributions-ID, Zertifikat-ID, Verbindungsgruppen-ID oder Web-ACL-ID zu filtern.
4. Wählen Sie den Namen des Distributionsmandanten aus.
5. Um die allgemeinen Details zu aktualisieren, wählen Sie Bearbeiten aus, nehmen die entsprechenden Änderungen vor und wählen anschließend Distributionsmandant aktualisieren aus.
6. Für alle anderen Einstellungen, die Sie aktualisieren möchten, wählen Sie die entsprechende Registerkarte aus, nehmen die Änderungen vor und speichern sie. Weitere Informationen zu den anpassbaren Einstellungen von Distributionsmandanten finden Sie unter Anpassungen von Distributionsmandanten.

Standard

So aktualisieren Sie eine Standarddistribution

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home.
2. Wählen Sie die ID einer Verteilung aus. Die Liste enthält alle Distributionen, die dem AWS Konto zugeordnet sind, mit dem Sie sich bei der CloudFront Konsole angemeldet haben.
3. Um die allgemeinen Einstellungen zu aktualisieren, wählen Sie Edit (Bearbeiten) aus. Andernfalls wählen Sie die Registerkarte der Einstellungen aus, die Sie aktualisieren wollen.
4. Nehmen Sie die gewünschten Änderungen vor und wählen Sie dann Änderungen speichern aus. Informationen zu den einzelnen Feldern finden Sie unter den folgenden Themen:
   - General settings (Allgemeine Einstellungen: Distribution Settings (Einstellungen für die Verteilung)
   - Origin settings (Ursprungseinstellungen: Ursprungseinstellungen
   - Cache behavior settings (Einstellungen für das Cache-Verhalten: Einstellungen für das Cache-Verhalten
5. Wenn Sie einen Ursprung in Ihrer Verteilung löschen möchten, gehen Sie wie folgt vor:
   a. Wählen Sie Behaviors (Verhalten) und stellen Sie sicher, dass Sie alle dem Ursprung zugeordneten Standard-Cache-Verhalten auf einen anderen Ursprung verschoben haben.
   b. Wählen Sie Origins (Ursprünge) und wählen Sie dann einen Ursprung aus.
   c. Wählen Sie Delete (Löschen).

Sie können eine Distribution auch mithilfe der CloudFront API aktualisieren:

- Informationen zum Aktualisieren einer Distribution finden Sie UpdateDistributionin der Amazon CloudFront API-Referenz.

> **Important**
>
> Wenn Sie eine Verteilung aktualisieren, sollten Sie daran denken, dass eine Reihe von zusätzlichen Felder erforderlich sind, die bei der Erstellung der Verteilung nicht erforderlich sind. Um sicherzustellen, dass alle erforderlichen Felder enthalten sind, wenn

Sie die CloudFront API zum Aktualisieren einer Distribution verwenden, folgen Sie den UpdateDistributionin der Amazon CloudFront API-Referenz beschriebenen Schritten.

Wenn Sie Änderungen an Ihrer Distributionskonfiguration speichern, CloudFront beginnt die Übertragung der Änderungen auf alle Edge-Standorte. Aufeinanderfolgende Konfigurationsänderungen werden in ihrer jeweiligen Reihenfolge verbreitet. Solange die Konfiguration an einem Edge-Standort aktualisiert wird, stellt CloudFront Ihre Inhalte von diesem Standort aus auf Basis der vorherigen Konfiguration bereit. Wenn die Konfiguration an einem Edge-Standort aktualisiert wurde, beginnt CloudFront sofort damit, Ihre Inhalte von diesem Standort aus auf Basis der neuen Konfiguration bereitzustellen.

Ihre Änderungen werden nicht gleichzeitig auf jeden Edge-Standort übertragen. Während CloudFront Ihre Änderungen übertragen werden, können wir anhand der vorherigen Konfiguration oder der neuen Konfiguration nicht feststellen, ob ein bestimmter Edge-Standort Ihre Inhalte bereitstellt.

Note

In seltenen Fällen, wenn eine Host- oder Netzwerkverbindung unterbrochen wird, kann es vorkommen, dass ein Teil des Datenverkehrs des Distributionsmandanten für einen kurzen Zeitraum über ältere Konfigurationen bedient wird, bis Ihre Änderungen im Netzwerk ankommen.

Ob Ihre Änderungen übernommen wurden, können Sie in den Details Ihrer Distribution in der Konsole einsehen. Das Feld Letzte Änderung ändert sich nach Abschluss der Bereitstellung von Wird bereitgestellt in ein Datum und eine Uhrzeit.

Kennzeichnen Sie eine Ressource CloudFront

Tags sind Wörter oder Ausdrücke, die Sie verwenden können, um Ihre AWS Ressourcen zu identifizieren und zu organisieren. Sie können jeder Ressource mehrere Tags hinzufügen, und jedes Tag enthält einen Schlüssel und einen Wert, den Sie definieren. Der Schlüssel könnte beispielsweise „domain“ heißen und der Wert „example.com“. Sie können Ihre Ressourcen anhand der von Ihnen hinzugefügten Tags suchen und filtern. Weitere Informationen zu Anwendungsfällen für Tags finden Sie unter Allgemeine Tagging-Strategien im Tagging AWS Resources and Tag Editor Guide.

Sie können Tags CloudFront beispielsweise mit den folgenden Beispielen verwenden:

- Erzwingen Sie tagbasierte Berechtigungen für CloudFront Distributionen. Weitere Informationen finden Sie unter ABAC mit CloudFront.
- Verfolgen Sie Fakturierungsinformationen in verschiedenen Kategorien. Wenn Sie Tags auf CloudFront Distributionen oder andere AWS Ressourcen (wie Amazon EC2-Instances oder Amazon S3 S3-Buckets) anwenden und die Tags aktivieren, wird ein Kostenzuordnungsbericht als kommagetrennter Wert (CSV-Datei) AWS generiert, in dem Ihre Nutzung und die Kosten nach Ihren aktiven Tags aggregiert werden.

  Sie können Tags anwenden, die geschäftliche Kategorien (wie Kostenstellen, Anwendungsnamen oder Eigentümer) darstellen, um die Kosten für mehrere Services zu organisieren. Weitere Informationen zur Verwendung von Tags für die Kostenzuordnung finden Sie unter Verwenden von Kostenzuordnungs-Tags im AWS Billing -Benutzerhandbuch.

Hinweise

- Sie können Verteilungen mit Tags markieren, nicht jedoch Ursprungszugriffsidentitäten oder Invalidierungen.
- Tag-Editor und Ressourcengruppen werden derzeit nicht unterstützt. CloudFront
- Informationen zur zurzeit maximalen Anzahl von Tags, die Sie einer Verteilung hinzufügen können, finden Sie unter Allgemeine Kontingente.

Inhalt

- Tag-Einschränkungen
- Markieren von Ressourcen für die Fakturierung
- Hinzufügen, Bearbeiten und Löschen von Tags
- Programmgesteuertes Markieren

Tag-Einschränkungen

Die folgenden grundlegenden Einschränkungen gelten für Tags (Markierungen):

- Informationen zur maximale Anzahl von Tags pro Distribution finden Sie unter Allgemeine Kontingente.
- Maximale Schlüssellänge – 128 Unicode-Zeichen.

- Maximale Wertlänge – 256 Unicode-Zeichen.
- Gültige Werte für Schlüssel und Wert — a-z A-Z, 0-9, Leerzeichen und die folgenden Zeichen: _.:/= + - und @
- Bei Tag-Schlüsseln und -Werten muss die Groß-/Kleinschreibung beachtet werden
- Verwenden Sie `aws:` nicht als Präfix für Schlüssel. Dieses Präfix ist zur Verwendung in AWS reserviert.

Markieren von Ressourcen für die Fakturierung

Sie können Tags verwenden, um Ihre AWS Rechnung so zu organisieren, dass sie Ihrer eigenen Kostenstruktur entspricht. Dazu müssen Sie zunächst Tags für die Kostenzuweisung in der Billing and Cost Management-Konsole aktivieren. Weitere Informationen zum Einrichten eines Kostenzuordnungsberichts mit Stichwörtern finden Sie unter Monatlicher Kostenzuordnungsbericht im AWS Billing User Guide. Um die Kosten kombinierter Ressourcen anzuzeigen, können Sie Ihre Fakturierungsinformationen nach Ressourcen mit gleichen Tag (Markierung)-Schlüsselwerten strukturieren. Beispielsweise können Sie mehrere Ressourcen mit einem bestimmten Anwendungsnamen markieren und dann Ihre Fakturierungsinformationen so organisieren, dass Sie die Gesamtkosten dieser Anwendung über mehrere Services hinweg sehen können. Weitere Informationen finden Sie unter Verwenden von Kostenzuordnungs-Tags im AWS Billing User Guide.

> **Note**
>
> Wenn Sie die Berichterstellung gerade erst aktiviert haben, werden die Daten für den aktuellen Monat nach 24 Stunden bereitgestellt.

Hinzufügen, Bearbeiten und Löschen von Tags

Sie können die CloudFront Konsole verwenden, um Tags für Ihre CloudFront Ressourcen zu verwalten.

Um Tags für eine Ressource hinzuzufügen, zu bearbeiten oder zu löschen

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home.
2. Navigieren Sie zu Ihrer Ressource:

- Für eine Distribution: Wählen Sie die ID für die Distribution aus, die Sie aktualisieren möchten.
- Für eine Funktion: Wählen Sie Funktionen und dann den Namen der Funktion, die Sie aktualisieren möchten.
- Für eine KeyValueStore: Wähle Funktionen, wähle die KeyValueStore Registerkarte und dann den Namen der Funktion KeyValueStore , die du aktualisieren möchtest.

3. Wählen Sie die Registerkarte Tags aus.
4. Wählen Sie Tags verwalten aus.
5. Auf der Seite Tags verwalten haben Sie folgende Möglichkeiten:
   - Um ein Tag hinzuzufügen, geben Sie einen Schlüssel und optional einen Wert für das Tag ein. Wählen Sie Neues Tag hinzufügen aus, um weitere Tags hinzuzufügen.
   - Um ein Tag zu bearbeiten, ändern Sie den Schlüssel des Tags oder seinen Wert oder beides. Sie können den Wert für ein Tag löschen, aber der Schlüssel ist erforderlich.
   - Zum Entfernen eines Tags wählen Sie Remove (Entfernen).
6. Wählen Sie Änderungen speichern aus.

Programmgesteuertes Markieren

Sie können auch die CloudFront API AWS Command Line Interface (AWS CLI), die AWS SDKs und das AWS Tools for Windows PowerShell Anwenden von Tags verwenden. Weitere Informationen finden Sie unter den folgenden Themen:

- CloudFront API-Operationen:
  - ListTagsForResource
  - TagResource
  - UntagResource
- AWS CLI — Siehe Cloudfront in der AWS CLI Befehlsreferenz
- AWS SDKs — Die entsprechende SDK-Dokumentation finden Sie auf der AWS Dokumentationsseite
- Tools für Windows PowerShell — siehe Amazon CloudFront in der AWS -Tools für PowerShell Cmdlet-Referenz

Löschen einer -Verteilung

Mit dem folgenden Verfahren wird eine Distribution mithilfe der CloudFront Konsole gelöscht. Informationen zum Löschen mit der CloudFront API finden Sie DeleteDistributionin der Amazon CloudFront API-Referenz.

Wichtige Informationen zum Löschen einer Distribution, bei der eine OAC an einen S3-Bucket angehängt ist, finden Sie unter Löschen einer Distribution mit einer OAC, die an einen S3-Bucket angehängt ist.

⚠ Warning

- Bevor Sie eine Distribution löschen können, müssen Sie sie deaktivieren. Hierzu benötigen Sie die Berechtigung zum Aktualisieren der Distribution. Nach dem Löschen kann die Distribution nicht wiederhergestellt werden.
- Wenn Sie eine Distribution deaktivieren, der ein alternativer Domainname zugeordnet ist, wird kein Traffic CloudFront mehr für diesen Domainnamen akzeptiert (z. B. www.example.com), auch wenn eine andere Distribution einen alternativen Domainnamen mit einem Platzhalter (*) hat, der derselben Domain entspricht (z. B. *.example.com).
- Sie können eine Distribution, die einen Pauschaltarif abonniert hat, nicht löschen. CloudFront Sie erhalten die folgende Fehlermeldung: „Sie können diese Distribution nicht löschen, solange sie einen Preisplan abonniert hat.“ Sie müssen zuerst den Preisplan kündigen und dann nach dem aktuellen Abrechnungszeitraum die Verteilung löschen.

Multi-tenant

Bevor Sie eine Multi-Tenant-Distribution löschen können, müssen Sie zunächst alle zugehörigen Distributionsmandanten löschen.

So löschen Sie eine Multi-Tenant-Distribution

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home.
2. Wählen Sie im rechten Bereich der CloudFront Konsole den Namen der Mehrmandantenverteilung aus, die Sie löschen möchten.
3. Wählen Sie für Mandanten alle zugehörigen Distributionsmandanten aus und löschen Sie sie.

4. Um die Distribution zu deaktivieren, wählen Sie Deaktivieren und anschließend zur Bestätigung Distribution deaktivieren aus.
5. Warten Sie, bis der neue Zeitstempel in der Spalte Letzte Änderung angezeigt wird.
   - Es kann einige Minuten dauern, CloudFront bis Ihre Änderung auf alle Edge-Standorte übertragen ist.
6. Wählen Sie Löschen und anschließend Distribution löschen aus.

So löschen Sie einen Distributionsmandanten

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home.
2. Wählen Sie unter SaaS die Option Distributionsmandanten aus.
3. Suchen Sie nach dem Distributionsmandanten. Verwenden Sie das Dropdown-Menü in der Suchleiste, um nach Domain, Name, Distributions-ID, Zertifikat-ID, Verbindungsgruppen-ID oder Web-ACL-ID zu filtern.
4. Wählen Sie den zu löschenden Distributionsmandanten aus.
5. Wählen Sie Mandant löschen und anschließend Distributionsmandant löschen aus.

Standard

So löschen Sie eine Standarddistribution

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home.
2. Suchen Sie im rechten Bereich der CloudFront Konsole nach der Distribution, die Sie löschen möchten.
   - Wenn in der Spalte Status für die Distribution bereits Deaktiviert angezeigt wird, fahren Sie mit Schritt 6 fort.
   - Wenn der Status der Distribution Aktiviert ist, in der Spalte Letzte Änderung jedoch immer noch Wird bereitgestellt angezeigt wird, warten Sie, bis die Bereitstellung abgeschlossen ist, bevor Sie mit Schritt 3 fortfahren.
3. Aktivieren Sie im rechten Bereich der CloudFront Konsole das Kontrollkästchen für die Distribution, die Sie löschen möchten.

4. Klicken Sie auf Disable (Deaktivieren), um die Verteilung zu deaktivieren, und wählen Sie Yes, Disable (Ja, deaktivieren) aus, um den Vorgang zu bestätigen. Wählen Sie anschließend Close (Schließen) aus.

   - Der Wert in der Spalte Status wird sofort in Deaktiviert geändert.

5. Warten Sie, bis der neue Zeitstempel in der Spalte Letzte Änderung angezeigt wird.

   - Es kann einige Minuten dauern CloudFront , bis Ihre Änderung auf alle Edge-Standorte übertragen ist.

6. Wählen Sie das Kontrollkästchen für die Distribution, die Sie löschen möchten.
7. Wählen Sie Löschen, Löschen aus.

   - Wenn die Option Löschen nicht verfügbar ist, bedeutet dies, dass Ihre Änderung immer noch auf die Edge-Standorte übertragen CloudFront wird. Warten Sie, bis der neue Zeitstempel in der Spalte Letzte Änderung angezeigt wird, und wiederholen Sie dann die Schritte 6–7.

Verwenden Sie bei Verteilungen verschiedene Ursprünge CloudFront

Wenn Sie eine Distribution erstellen, geben Sie den Ursprung an, von dem Anfragen für die Dateien CloudFront gesendet werden. Sie können mehrere verschiedene Arten von Ursprüngen mit verwenden CloudFront. Sie können beispielsweise einen Amazon S3 S3-Bucket, einen MediaStore Container, einen MediaPackage Channel, einen Application Load Balancer oder eine AWS Lambda Funktions-URL verwenden. Wenn Sie Ihre CloudFront Distribution erstellen, konfiguriert die meisten Distributionseinstellungen CloudFront automatisch für Sie, basierend auf Ihrem Inhaltstyp. Weitere Informationen finden Sie unter Referenz für vorkonfigurierte Distributionseinstellungen.

Wenn Sie einen Application Load Balancer, Network Load Balancer oder eine EC2-Instance in einem privaten Subnetz haben, können Sie sie als VPC-Ursprung verwenden. Mit VPC-Ursprüngen kann auf Ihre Anwendungen nur in einem privaten Subnetz mit einer CloudFront Verteilung zugegriffen werden, wodurch verhindert wird, dass Ihre Anwendung im öffentlichen Internet zugänglich ist. Weitere Informationen finden Sie unter the section called “Zugriffsbeschränkung mit VPC-Ursprüngen”.

> **Note**
>
> Sie können Edge-Funktionen verwenden, um dynamisch den geeigneten Ursprung für jede Anforderung auszuwählen. Mithilfe von CloudFront Functions oder Lambda @Edge können Sie Anfragen anhand von Faktoren wie dem geografischen Standort des Betrachters, den Anforderungsheadern oder Abfragezeichenfolgenparametern an verschiedene Ursprünge weiterleiten. Weitere Informationen finden Sie unter Anpassen am Edge mit Funktionen.

Themen

- Verwenden eines Amazon-S3-Buckets
- Verwenden Sie einen MediaStore Container oder einen MediaPackage Kanal
- Verwenden eines Application Load Balancer
- Verwenden eines Network Load Balancer
- Verwenden einer Lambda-Funktions-URL
- Verwenden Sie eine AWS Lambda Web-Funktion
- Verwenden von Amazon EC2 (oder eines anderen benutzerdefinierten Ursprungs)
- Verwenden Sie CloudFront Ursprungsgruppen
- Verwenden von Amazon API Gateway

Verwenden eines Amazon-S3-Buckets

In den folgenden Themen werden die verschiedenen Möglichkeiten beschrieben, wie Sie einen Amazon S3 S3-Bucket als Ursprung für eine CloudFront Distribution verwenden können.

Themen

- Verwenden eines standardmäßigen Amazon-S3-Buckets
- Verwenden von Amazon S3 Object Lambda
- Verwenden von Amazon-S3-Zugangspunkten
- Verwenden eines Amazon-S3-Buckets, der als Website-Endpunkt konfiguriert ist
- CloudFront Zu einem vorhandenen Amazon S3 S3-Bucket hinzufügen
- Verschieben Sie einen Amazon S3 S3-Bucket in einen anderen AWS-Region
- Verwenden Sie einen Amazon S3 Multi-Region Access Point

Verwenden eines standardmäßigen Amazon-S3-Buckets

Wenn Sie Amazon S3 als Quelle für Ihre Distribution verwenden, platzieren Sie die Objekte, die Sie liefern CloudFront möchten, in einem Amazon S3 S3-Bucket. Sie können alle von Amazon S3 unterstützten Methoden zum Bereitstellen Ihrer Objekte in Amazon S3 verwenden. Sie können beispielsweise die Amazon-S3-Konsole oder -API oder ein Drittanbieter-Tool verwenden. Sie können in Ihrem Bucket wie für jeden anderen standardmäßigen Amazon-S3-Bucket eine Hierarchie zum Speichern der Objekte erstellen.

Wenn Sie einen vorhandenen Amazon S3 S3-Bucket als Ihren CloudFront Ursprungsserver verwenden, ändert sich der Bucket in keiner Weise. Sie können ihn weiterhin wie gewohnt verwenden, um Amazon S3-Objekte zum Amazon S3-Standardpreis zu speichern und darauf zuzugreifen. Es fallen die regulären Amazon S3-Gebühren für die Speicherung der Objekte im Bucket an. Weitere Informationen zu den zu CloudFront verwendenden Gebühren finden Sie unter CloudFrontAmazon-Preise. Weitere Informationen zur Verwendung CloudFront mit einem vorhandenen S3-Bucket finden Sie unterthe section called “CloudFront Zu einem vorhandenen Amazon S3 S3-Bucket hinzufügen”.

> **Important**
>
> Damit Ihr Bucket verwendet werden kann CloudFront, muss der Name den DNS-Benennungsanforderungen entsprechen. Weitere Informationen finden Sie unter Benennungsregeln für Buckets im Benutzerhandbuch zu Amazon Simple Storage Service.

Wenn Sie einen Amazon S3 S3-Bucket als Ursprung für angeben CloudFront, empfehlen wir Ihnen, das folgende Format zu verwenden:

```
bucket-name.s3.region.amazonaws.com
```

Wenn Sie den Bucket-Namen in diesem Format angeben, können Sie die folgenden CloudFront -Funktionen verwenden:

- Konfigurieren Sie CloudFront für die Kommunikation mit Ihrem Amazon S3 S3-Bucket mithilfe von SSL/TLS. Weitere Informationen finden Sie unter the section called “Verwenden Sie HTTPS mit CloudFront”.
- Verwenden Sie eine Ursprungszugriffskontrolle, um zu verlangen, dass Zuschauer über CloudFront URLs auf Ihre Inhalte zugreifen, nicht über Amazon S3 S3-URLs. Weitere Informationen finden Sie unter the section called “Beschränken des Zugriffs auf einen Amazon-S3-Ursprung”.

- Aktualisieren Sie den Inhalt Ihres Buckets, indem Sie PUT Anfragen an senden POST CloudFront. Weitere Informationen finden Sie unter the section called “HTTP-Methoden” im Thema the section called “Wie CloudFront verarbeitet und leitet Anfragen an Ihren Amazon S3-Ursprung weiter”.

Verwenden Sie für die Angabe des Buckets nicht die folgenden Formate:

- Den Amazon-S3-Pfadstil: `s3.amazonaws.com/bucket-name`
- Den Amazon-S3-CNAME

> **Note**
>
> CloudFront unterstützt S3-Ursprünge unter Verwendung beliebiger Speicherklassen, einschließlich S3Intelligent-Tiering. Wenn CloudFront Objekte von einem S3-Ursprung angefordert werden, werden die Objekte unabhängig von der Speicherebene abgerufen, in der sie sich derzeit befinden. Die Verwendung CloudFront mit S3 Intelligent-Tiering hat keinen Einfluss auf die Leistung oder Funktionalität Ihrer Distribution. Weitere Informationen finden Sie unter Verwaltung der Speicherkosten mit Amazon S3 Intelligent-Tiering im Amazon Simple Storage Service-Benutzerhandbuch.

Verwenden von Amazon S3 Object Lambda

Wenn Sie einen Object-Lambda-Zugangspunkt erstellen, generiert Amazon S3 automatisch ein eindeutiges Alias für Ihren Object-Lambda-Zugangspunkt. Sie können diesen Alias anstelle eines Amazon S3 S3-Bucket-Namens als Ursprung für Ihre CloudFront Distribution verwenden.

Wenn Sie einen Object Lambda Access Point-Alias als Ursprung für verwenden CloudFront, empfehlen wir Ihnen, das folgende Format zu verwenden:

`alias.s3.region.amazonaws.com`

Weitere Informationen zur Suche des *alias* finden Sie unter Verwenden eines Alias im Bucket-Stil für Ihren Object-Lambda-Zugangspunkt für S3-Buckets im Amazon-S3-Benutzerhandbuch.

> **Important**
>
> Wenn Sie einen Object Lambda Access Point als Ursprung für verwenden CloudFront, müssen Sie die Origin-Zugriffskontrolle verwenden.

Ein Beispiel für einen Anwendungsfall finden Sie unter Verwenden von Amazon S3 Object Lambda mit Amazon CloudFront zur Anpassung von Inhalten für Endbenutzer.

CloudFront behandelt einen Object Lambda Access Point-Ursprung genauso wie einen standardmäßigen Amazon S3 S3-Bucket-Ursprung.

Wenn Sie Amazon S3 Object Lambda als Ursprung für Ihre Distribution verwenden, müssen Sie die folgenden vier Berechtigungen konfigurieren:

Object Lambda Access Point

So fügen Sie Berechtigungen für den Objekt-Lambda-Zugangspunkt hinzu

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im Navigationsbereich Object-Lambda-Zugangspunkte aus.
3. Wählen Sie den Object-Lambda-Zugangspunkt aus, den Sie verwenden möchten.
4. Wählen Sie die Registerkarte Berechtigungen.
5. Wählen Sie im Abschnitt Object-Lambda-Zugangspunktrichtlinie die Option Bearbeiten aus.
6. Fügen Sie die folgende Richtlinie in das Feld Richtlinie ein.

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "cloudfront.amazonaws.com"
            },
            "Action": "s3-object-lambda:Get*",
            "Resource": "arn:aws:s3-object-lambda:us-
east-1:123456789012:accesspoint/Object-Lambda-Access-Point-name",
            "Condition": {
                "StringEquals": {
                    "aws:SourceArn":
 "arn:aws:cloudfront::123456789012:distribution/CloudFront-distribution-
ID"
                }
```

```
            }
        }
    ]
}
```

7. Wählen Sie Änderungen speichern aus.

Amazon S3 Access Point

So fügen Sie Berechtigungen für den Amazon-S3-Zugangspunkt hinzu

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im Navigationsbereich Zugangspunkte aus.
3. Wählen Sie den Amazon S3 Access Point aus, den Sie verwenden möchten.
4. Wählen Sie die Registerkarte Berechtigungen.
5. Wählen Sie im Abschnitt Zugangspunktrichtlinie die Option Bearbeiten aus.
6. Fügen Sie die folgende Richtlinie in das Feld Richtlinie ein.

   JSON

```
{
    "Version":"2012-10-17",
    "Id": "default",
    "Statement": [
        {
            "Sid": "s3objlambda",
            "Effect": "Allow",
            "Principal": {
                "Service": "cloudfront.amazonaws.com"
            },
            "Action": "s3:*",
            "Resource": [
                "arn:aws:s3:us-east-1:123456789012:accesspoint/Access-
Point-name",
                "arn:aws:s3:us-east-1:123456789012:accesspoint/Access-
Point-name/object/*"
            ],
            "Condition": {
                "ForAnyValue:StringEquals": {
```

```
                    "aws:CalledVia": "s3-object-lambda.amazonaws.com"
                }
            }
        }
    ]
}
```

7. Wählen Sie Speichern.

Amazon S3 bucket

So fügen Sie Berechtigungen zum Amazon-S3-Bucket hinzu

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im Navigationsbereich die Option Buckets aus.
3. Wählen Sie den Amazon-S3-Bucket aus, den Sie verwenden möchten.
4. Wählen Sie die Registerkarte Berechtigungen.
5. Wählen Sie im Abschnitt Bucket-Richtlinie die Option Bearbeiten aus.
6. Fügen Sie die folgende Richtlinie in das Feld Richtlinie ein.

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "*"
            },
            "Action": "*",
            "Resource": [
                "arn:aws:s3:::bucket-name",
                "arn:aws:s3:::bucket-name/*"
            ],
            "Condition": {
                "StringEquals": {
                    "s3:DataAccessPointAccount": "AWS-account-ID"
                }
```

```
            }
        }
    ]
}
```

7. Wählen Sie Änderungen speichern aus.

AWS Lambda function

So fügen Sie Berechtigungen zur Lambda-Funktion hinzu

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die AWS Lambda Konsole unter https://console.aws.amazon.com/lambda/.
2. Wählen Sie im Navigationsbereich Funktionen aus.
3. Wählen Sie die AWS Lambda Funktion aus, die Sie verwenden möchten.
4. Wählen Sie die Registerkarte Konfiguration und dann Berechtigungen aus.
5. Wählen Sie im Abschnitt mit den Resource-based Richtlinienerklärungen die Option Berechtigungen hinzufügen aus.
6. Wählen Sie AWS-Konto.
7. Geben Sie einen Namen für die Anweisungs-ID ein.
8. Geben Sie `cloudfront.amazonaws.com` als Prinzipal ein.
9. Wählen Sie `lambda:InvokeFunction` im Drop-down-Menü Aktion aus:
10. Wählen Sie Speichern.

Verwenden von Amazon-S3-Zugangspunkten

Wenn Sie einen S3-Zugangspunkt verwenden, generiert Amazon S3 automatisch ein eindeutiges Alias für Sie. Sie können diesen Alias anstelle eines Amazon S3 S3-Bucket-Namens als Ursprung für Ihre CloudFront Distribution verwenden.

Wenn Sie einen Amazon S3 Access Point-Alias als Ursprung für verwenden CloudFront, empfehlen wir Ihnen, das folgende Format zu verwenden:

alias`.s3.`*region*`.amazonaws.com`

Weitere Informationen zur Suche des *alias* finden Sie unter Verwenden eines Alias im Bucket-Stil für Ihren S3-Bucket-Zugangspunkt im Amazon-S3-Benutzerhandbuch.

> **Important**
>
> Wenn Sie einen Amazon S3 Access Point als Ausgangspunkt für verwenden CloudFront, müssen Sie die Origin-Zugriffskontrolle verwenden.

CloudFront behandelt einen Amazon S3 Access Point-Ursprung genauso wie einen standardmäßigen Amazon S3 S3-Bucket-Ursprung.

Wenn Sie Amazon S3 Object Lambda als Ursprung für Ihre Distribution verwenden, müssen Sie die folgenden beiden Berechtigungen konfigurieren:

Amazon S3 Access Point

So fügen Sie Berechtigungen für den Amazon-S3-Zugangspunkt hinzu

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im Navigationsbereich Zugangspunkte aus.
3. Wählen Sie den Amazon S3 Access Point aus, den Sie verwenden möchten.
4. Wählen Sie die Registerkarte Berechtigungen.
5. Wählen Sie im Abschnitt Zugangspunktrichtlinie die Option Bearbeiten aus.
6. Fügen Sie die folgende Richtlinie in das Feld Richtlinie ein.

   JSON

```
{
    "Version":"2012-10-17",
    "Id": "default",
    "Statement": [
        {
            "Sid": "s3objlambda",
            "Effect": "Allow",
            "Principal": {"Service": "cloudfront.amazonaws.com"},
            "Action": "s3:*",
            "Resource": [
                "arn:aws:s3:us-east-1:123456789012:accesspoint/Access-
Point-name",
```

```
                "arn:aws:s3:us-east-1:123456789012:accesspoint/Access-
Point-name/object/*"
            ],
            "Condition": {
                "StringEquals": {"aws:SourceArn":
 "arn:aws:cloudfront::123456789012:distribution/CloudFront-distribution-
ID"}
            }
        }
    ]
}
```

7. Wählen Sie Speichern.

Amazon S3 bucket

So fügen Sie Berechtigungen zum Amazon-S3-Bucket hinzu

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im Navigationsbereich die Option Buckets aus.
3. Wählen Sie den Amazon-S3-Bucket aus, den Sie verwenden möchten.
4. Wählen Sie die Registerkarte Berechtigungen.
5. Wählen Sie im Abschnitt Bucket-Richtlinie die Option Bearbeiten aus.
6. Fügen Sie die folgende Richtlinie in das Feld Richtlinie ein.

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "*"
            },
            "Action": "*",
            "Resource": [
                "arn:aws:s3:::bucket-name",
                "arn:aws:s3:::bucket-name/*"
```

```
            ],
            "Condition": {
                "StringEquals": {
                    "s3:DataAccessPointAccount": "AWS-account-ID"
                }
            }
        }
    ]
}
```

7. Wählen Sie Änderungen speichern aus.

Verwenden eines Amazon-S3-Buckets, der als Website-Endpunkt konfiguriert ist

Sie können einen Amazon S3 S3-Bucket, der als Website-Endpunkt konfiguriert ist, als benutzerdefinierten Ursprung mit verwendenCloudFront. Wenn Sie Ihre CloudFront Distribution konfigurieren, geben Sie für den Ursprung den statischen Amazon S3 S3-Website-Hosting-Endpunkt für Ihren Bucket ein. Dieser Wert wird in der Amazon-S3-Konsole auf der Registerkarte Properties (Eigenschaften) im Bereich Static Website Hosting (Hosten statischer Websites) angezeigt. Beispiel:

`http://bucket-name.s3-website-region.amazonaws.com`

Weitere Informationen zur Angabe von Amazon-S3-Endpunkten für statische Websites finden Sie unter Website-Endpunkte im Benutzerhandbuch zu Amazon Simple Storage Service.

Wenn Sie den Bucket-Namen in diesem Format als Ursprung angeben, können Sie Amazon S3-Umleitungen und benutzerdefinierte Amazon S3-Fehlerdokumente verwenden. Weitere Informationen finden Sie unter Konfiguration eines benutzerdefinierten Fehlerdokuments und Konfiguration einer Umleitung im Amazon Simple Storage Service-Benutzerhandbuch. (bietet CloudFront auch benutzerdefinierte Fehlerseiten. Weitere Informationen finden Sie unterthe section called “Erstellen einer benutzerdefinierten Fehlerseite für bestimmte HTTP-Statuscodes”.)

Wenn Sie einen Amazon S3 S3-Bucket als Ihren CloudFront Ursprungsserver verwenden, ändert sich der Bucket in keiner Weise. Sie können ihn weiter wie normal zu den regulären Amazon S3-Gebühren verwenden. Weitere Informationen zu den zu CloudFront verwendenden Gebühren finden Sie unter CloudFront Amazon-Preise.

> **Note**
>
> Wenn Sie die CloudFront API verwenden, um Ihre Distribution mit einem Amazon S3 S3-Bucket zu erstellen, der als Website-Endpunkt konfiguriert ist, müssen Sie ihn mithilfe von

konfigurieren`CustomOriginConfig`, obwohl die Website in einem Amazon S3 S3-Bucket gehostet wird. Weitere Informationen zum Erstellen von Verteilungen mithilfe der CloudFront API finden Sie CreateDistributionin der Amazon CloudFront API-Referenz.

CloudFront Zu einem vorhandenen Amazon S3 S3-Bucket hinzufügen

Wenn Sie Ihre Objekte in einem Amazon S3 S3-Bucket speichern, können Sie entweder festlegen, dass Benutzer Ihre Objekte direkt von S3 abrufen, oder Sie können so konfigurieren, CloudFront dass Ihre Objekte von S3 abgerufen und dann an Ihre Benutzer verteilt werden. Die Verwendung CloudFront kann kostengünstiger sein, wenn Ihre Benutzer häufig auf Ihre Objekte zugreifen, da bei höherer Nutzung der Preis für die CloudFront Datenübertragung niedriger ist als der Preis für die Amazon S3 S3-Datenübertragung. Darüber hinaus sind Downloads mit Amazon S3 allein schneller CloudFront als mit Amazon S3 allein, da Ihre Objekte näher an Ihren Benutzern gespeichert werden.

> **Note**
>
> Wenn Sie CloudFront die ursprungsübergreifenden Amazon S3-Einstellungen für die gemeinsame Nutzung von Ressourcen respektieren möchten, konfigurieren Sie die Konfiguration so, CloudFront dass der `Origin` Header an Amazon S3 weitergeleitet wird. Weitere Informationen finden Sie unter the section called “Zwischenspeichern von Inhalten auf der Grundlage von Anforderungsheadern”.

Wenn Sie derzeit Inhalte direkt aus Ihrem Amazon S3 S3-Bucket unter Verwendung Ihres eigenen Domainnamens (z. B. example.com) anstelle des Domainnamens Ihres Amazon S3-Buckets (z. B. amzn-s3-demo-bucket. s3.us-west-2.amazonaws.com) verteilen, können Sie ohne Unterbrechung hinzufügen CloudFront, indem Sie das folgende Verfahren verwenden.

Um hinzuzufügen CloudFront , wenn Sie Ihre Inhalte bereits über Amazon S3 verteilen

1. Erstellen Sie eine CloudFront Distribution. Weitere Informationen finden Sie unter the section called “Eine Distribution erstellen”.

   Wenn Sie die Distribution erstellen, geben Sie den Namen Ihres Amazon S3-Buckets als Ursprungsserver an.

> **Important**
>
> Damit Ihr Bucket verwendet werden kann CloudFront, muss der Name den DNS-Benennungsanforderungen entsprechen. Weitere Informationen finden Sie unter Benennungsregeln für Buckets im Benutzerhandbuch zu Amazon Simple Storage Service.

Wenn Sie einen CNAME mit Amazon S3 verwenden, geben Sie den CNAME für Ihre Verteilung ebenfalls an.

2. Erstellen Sie eine Testwebseite mit Links, die auf öffentlich verfügbare Objekte in Ihrem Amazon S3-Bucket verweisen, und testen Sie diese Links. Verwenden Sie bei diesem ersten Test den CloudFront -Domain-Namen Ihrer Verteilung in den Objekt-URLs, z. B. `https://d111111abcdef8.cloudfront.net/images/image.jpg`.

   Weitere Hinweise zum Format von CloudFront URLs finden Sie unterthe section called “Anpassen von Datei-URLs”.

3. Wenn Sie Amazon-S3-CNAMEs verwenden, verweist Ihre Anwendung mit dem Domainnamen (z. B. example.com) auf die Objekte in Ihrem Amazon-S3-Bucket, anstatt den Namen des Buckets (z. B. amzn-s3-demo-bucket.s3.amazonaws.com) zu verwenden. Wenn Sie weiterhin Ihren Domainnamen verwenden möchten, um auf Objekte zu verweisen, anstatt den CloudFront Domainnamen für Ihre Distribution zu verwenden (z. B. d111111abcdef8.cloudfront.net), müssen Sie Ihre Einstellungen bei Ihrem DNS-Dienstanbieter aktualisieren.

   Damit Amazon S3-CNAMEs korrekt funktionieren, muss Ihr DNS-Serviceanbieter über einen CNAME-Ressourcendatensatz für Ihre Domäne verfügen, der Abfragen für die Domäne an Ihren Amazon S3-Bucket weiterleitet. Wenn ein Benutzer z. B. dieses Objekt anfordert:

   `https://example.com/images/image.jpg`

   Die Anfrage wird automatisch umgeleitet und dem Benutzer wird dieses Objekt angezeigt:

   `https://amzn-s3-demo-bucket.s3.amazonaws.com/images/image.jpg`

   Um Anfragen an Ihre CloudFront Distribution und nicht an Ihren Amazon S3 S3-Bucket weiterzuleiten, müssen Sie die von Ihrem DNS-Dienstanbieter bereitgestellte Methode verwenden, um den CNAME-Ressourceneintrag für Ihre Domain zu aktualisieren. Dieser

aktualisierte CNAME-Eintrag leitet DNS-Anfragen von Ihrer Domain an den CloudFront Domainnamen für Ihre Distribution weiter. Weitere Informationen finden Sie in der Dokumentation Ihres DNS-Serviceanbieters.

> **Note**
>
> Wenn Sie Route 53 als DNS-Service verwenden, können Sie entweder einen CNAME-Ressourcendatensatz oder einen Alias-Ressourcendatensatz verwenden. Informationen zum Bearbeiten von Ressourceneintragssätzen finden Sie unter Bearbeiten von Datensätzen. Informationen zu Alias-Ressourceneintragssätzen finden Sie unter Wählen zwischen Alias- und Nicht-Alias-Datensätzen. Beide Themen finden Sie im Entwicklerhandbuch für Amazon Route 53.

Weitere Hinweise zur Verwendung von CNAMEs mit finden Sie CloudFront unter. the section called “Verwenden benutzerdefinierter URLs”

Wenn Sie den CNAME-Ressourcendatensatz aktualisiert haben, kann es bis zu 72 Stunden dauern, bis die Änderungen im gesamten DNS-System übernommen werden, auch wenn das in der Regel schneller geschieht. Während dieser Zeit werden einige Anfragen für Ihre Inhalte weiterhin an Ihren Amazon S3 S3-Bucket und andere an Ihren Amazon S3-Bucket weitergeleitet. CloudFront

Verschieben Sie einen Amazon S3 S3-Bucket in einen anderen AWS-Region

Wenn Sie Amazon S3 als Ursprung für eine CloudFront Distribution verwenden und den Bucket in einen anderen verschieben AWS-Region, CloudFront kann es bis zu einer Stunde dauern, bis die Datensätze aktualisiert sind, sodass die neue Region verwendet wird, wenn beide der folgenden Bedingungen zutreffen:

- Sie verwenden eine CloudFront Origin Access Identity (OAI), um den Zugriff auf den Bucket einzuschränken.
- Sie verschieben den Bucket in eine Amazon S3-Region, die Signature Version 4 für die Authentifizierung erfordert.

Wenn Sie OAIs verwenden, CloudFront verwendet die Region (neben anderen Werten), um die Signatur zu berechnen, die verwendet wird, um Objekte aus Ihrem Bucket anzufordern.

Weitere Informationen über OAIs finden Sie unter the section called “Verwenden einer Ursprungszugriffsidentität (Legacy, nicht empfohlen)”. Eine Liste der AWS-Regionen unterstützten Signature Version 2 finden Sie unter Signaturprozess für Signature Version 2 in der Allgemeine Amazon Web Services-Referenz.

Um eine schnellere Aktualisierung der CloudFront Datensätze zu erzwingen, können Sie Ihre CloudFront Distribution aktualisieren, indem Sie beispielsweise das Feld Beschreibung auf der Registerkarte Allgemein in der CloudFront Konsole aktualisieren. Wenn Sie eine Distribution aktualisieren, überprüft es CloudFront sofort die Region, in der sich Ihr Bucket befindet. Das Übertragen der Änderung auf alle Edge-Standorte sollte nur wenige Minuten in Anspruch nehmen.

Verwenden Sie einen Amazon S3 Multi-Region Access Point

Sie können auch einen Amazon S3 Multi-Region Access Point als CloudFront Ausgangspunkt verwenden. Amazon S3 Multi-Region Access Points bieten einen globalen Endpunkt, der Anfragen basierend auf der Netzwerklatenz automatisch an den nächstgelegenen Amazon S3 S3-Bucket weiterleitet. Wenn Sie einen Amazon S3 Multi-Region Access Point als Ursprung verwenden, geben Sie den Multi-Region Access Point-Hostnamen als Ursprungs-Domainnamen an:

```
multi-region-access-point-alias.accesspoint.s3-global.amazonaws.com
```

Informationen zur Beschränkung des Zugriffs auf den S3 Multi-Region Access Point, sodass er nur über Ihre CloudFront Distribution zugänglich ist, finden Sie unterthe section called “Beschränken Sie den Zugriff auf einen Amazon S3 Multi-Region Access Point-Ursprung”.

Verwenden Sie einen MediaStore Container oder einen MediaPackage Kanal

Um Videos zu streamen CloudFront, können Sie einen Amazon S3 S3-Bucket einrichten, der als MediaStore Container konfiguriert ist, oder einen Kanal und Endpunkte mit MediaPackage erstellen. Anschließend erstellen und konfigurieren Sie eine Distribution, um das Video CloudFront zu streamen.

Weitere Informationen und Schritt-für-Schritt-Anweisungen finden Sie in den folgenden Themen:

- the section called “Stellen Sie das Video bereit, indem Sie AWS Elemental MediaStore als Ursprung”
- the section called “Servieren Sie Live-Videos, formatiert mit AWS Elemental MediaPackage”

Verwenden eines Application Load Balancer

Sie können CloudFront damit den Datenverkehr sowohl an interne als auch an mit dem Internet verbundene Application Load Balancer weiterleiten.

Wenn es sich bei Ihrem Ursprung um einen oder mehrere HTTP(S)-Server (Webserver) handelt, die auf einer oder mehreren Amazon-EC2-Instances gehostet werden, können Sie einen mit dem Internet verbundenen Application Load Balancer verwenden, um den Datenverkehr an die Instances zu verteilen. Ein mit dem Internet verbundener Load Balancer hat einen öffentlich auflösbaren DNS-Namen und leitet Anforderungen von Clients über das Internet an Ziele weiter.

Weitere Informationen zur Verwendung eines mit dem Internet verbundenen Application Load Balancer als Ausgangsbasis und dazu CloudFront, wie Sie sicherstellen können, dass Zuschauer nur über den Load Balancer CloudFront und nicht direkt auf Ihre Webserver zugreifen können, finden Sie unter. the section called “Beschränken des Zugriffs auf Application Load Balancer”

Alternativ können Sie VPC-Ursprünge verwenden, um Inhalte aus Anwendungen bereitzustellen, die mit einem internen Application Load Balancer in Ihren privaten Virtual Private Cloud (VPC)-Subnetzen gehostet werden. VPC-Ursprünge verhindern, dass Ihre Anwendung im öffentlichen Internet zugänglich ist. Weitere Informationen finden Sie unter Zugriffsbeschränkung mit VPC-Ursprüngen.

Verwenden eines Network Load Balancer

Sie können sowohl interne als auch mit dem Internet verbundene Network Load Balancer mit Amazon verwenden. CloudFront Sie können interne Network Load Balancer in privaten Subnetzen verwenden, CloudFront indem Sie VPC-Ursprünge verwenden. CloudFront VPC-Ursprünge ermöglichen es Ihnen, Inhalte von Anwendungen bereitzustellen, die in privaten VPC-Subnetzen gehostet werden, ohne sie dem öffentlichen Internet zugänglich zu machen. Weitere Informationen finden Sie unter Zugriffsbeschränkung mit VPC-Ursprüngen.

Alternativ können Sie es auch CloudFront für die Übertragung von Datenverkehr von mit dem Internet verbundenen Network Load Balancers verwenden. Ein mit dem Internet verbundener Load Balancer hat einen öffentlich auflösbaren DNS-Namen und kann Anfragen sowohl von Clients im Internet als auch von Distributionen empfangen. CloudFront

Verwenden einer Lambda-Funktions-URL

Eine Lambda-Funktions-URL ist ein dedizierter HTTPS-Endpunkt für eine Lambda-Funktion. Sie können eine Lambda-Funktions-URL verwenden, um eine Serverless-Webanwendung vollständig

in Lambda zu entwickeln. Sie können die Lambda-Webanwendung direkt über die Funktions-URL aufrufen, ohne dass eine Integration in API Gateway oder einen Application Load Balancer erforderlich ist.

Wenn Sie eine serverlose Webanwendung mithilfe von Lambda-Funktionen mit Funktions-URLs erstellen, können Sie sie hinzufügen, um die folgenden Vorteile CloudFront zu erhalten:

- Beschleunigen Ihrer Anwendung, indem Sie Inhalte näher an den Viewern zwischenspeichern
- Verwenden eines benutzerdefinierten Domänennamens für Ihre Webanwendung
- Verschiedene URL-Pfade mithilfe von CloudFront Cache-Verhalten an verschiedene Lambda-Funktionen weiterleiten
- Blockieren Sie bestimmte Anfragen mithilfe CloudFront geografischer Einschränkungen oder AWS WAF (oder beidem)
- Verwenden Sie AWS WAF with CloudFront , um Ihre Anwendung vor bösartigen Bots zu schützen, häufige Anwendungs-Exploits zu verhindern und den Schutz vor DDoS-Angriffen zu verbessern

Um eine Lambda-Funktions-URL als Ursprung für eine CloudFront Distribution zu verwenden, geben Sie den vollständigen Domainnamen der Lambda-Funktions-URL als Ursprungsdomäne an. Ein Domänenname der Lambda-Funktions-URL weist das folgende Format auf:

`function-URL-ID``.lambda-url.`*`AWS-Region`*`.on.aws`

Wenn Sie eine Lambda-Funktions-URL als Ursprung für eine CloudFront Distribution verwenden, muss die Funktions-URL öffentlich zugänglich sein. Verwenden Sie dazu eine der folgenden Optionen:

- Wenn Sie Origin Access Control (OAC) verwenden, muss der `AuthType` Parameter der Lambda-Funktions-URL den `AWS_IAM` Wert verwenden und die `lambda:InvokeFunction` Berechtigungen `lambda:InvokeFunctionUrl` und in einer ressourcenbasierten Richtlinie zulassen. Weitere Informationen zur Verwendung von Lambda-Funktions-URLs für die OAC finden Sie unter Beschränken Sie den Zugriff auf ein AWS Lambda Herkunft der Funktions-URL.
- Wenn Sie nicht die OAC verwenden, legen Sie den Parameter `AuthType` der Funktions-URL auf `NONE` fest und erteilen Sie die Berechtigung `lambda:InvokeFunctionUrl` in einer ressourcenbasierten Richtlinie.

Sie können den Anfragen, die an den Ursprung CloudFront gesendet werden, auch einen benutzerdefinierten Origin-Header hinzufügen und Funktionscode schreiben, um eine Fehlerantwort zurückzugeben, wenn der Header nicht in der Anfrage vorhanden ist. Dadurch wird sichergestellt, dass Benutzer nur über die URL der Lambda-Funktion und nicht direkt über die CloudFront URL der Lambda-Funktion auf Ihre Webanwendung zugreifen können.

Weitere Informationen zu Lambda-Funktions-URLs finden Sie in den folgenden Themen im Entwicklerhandbuch zu AWS Lambda :

- Lambda-Funktions-URLs – Eine allgemeine Übersicht über die Funktion der Lambda-Funktions-URLs
- Aufrufen von Lambda-Funktions-URLs – Enthält Details zu den Anforderungs- und Antwortnutzlasten, die zum Codieren Ihrer Serverless-Webanwendung verwendet werden sollen
- Sicherheits- und Authentifizierungsmodell für Lambda-Funktions-URLs – enthält Details zu den Lambda-Authentifizierungstypen

Verwenden Sie eine AWS Lambda Web-Funktion

Eine Lambda-Webfunktion ist eine Webanwendung, die auf einem eigenen HTTPS-Endpunkt läuft AWS Lambda und diesen bereitstellt. Der Endpunkt kann regional sein, wodurch die Lambda-Webfunktion von einer Region aus bedient wird AWS-Region, oder global, der sie von mehreren Regionen aus bedient. Sie können eine Lambda-Webfunktion als Ursprung für eine CloudFront Distribution verwenden.

Wenn Sie eine Webanwendung über eine Lambda-Webfunktion bereitstellen, können Sie sie hinzufügen, CloudFront um die folgenden Vorteile zu erhalten:

- Beschleunigen Sie Ihre Anwendung, indem Sie Antworten näher am Betrachter zwischenspeichern, wodurch auch Ihre Lambda-Webfunktion entlastet wird
- Verwenden Sie einen benutzerdefinierten Domainnamen und Ihr eigenes TLS-Zertifikat für Ihre Webanwendung
- Mithilfe von CloudFront Cache-Verhalten verschiedene URL-Pfade an verschiedene Lambda-Webfunktionen oder andere Ursprünge weiterleiten
- Blockieren Sie bestimmte Anfragen mithilfe CloudFront geografischer Einschränkungen oder AWS WAF (oder beidem)
- Verwenden Sie AWS WAF with CloudFront , um Ihre Anwendung vor bösartigen Bots zu schützen, häufige Anwendungs-Exploits zu verhindern und den Schutz vor DDoS-Angriffen zu verbessern

- Verwenden Sie Origin Access Control (OAC), um sicherzustellen Lambda Zuschauer nur über CloudFront

Um eine Lambda-Webfunktion als Ursprung für eine CloudFront Distribution zu verwenden, geben Sie den Hostnamen ihres Endpunkts als Ursprungsdomäne an. CloudFront identifiziert den Endpunkttyp anhand des Hostnamens:

- Regionaler Endpunkt (`HomeRegion`und `PerRegion` Endpunkttypen) — Der Hostname endet auf `.lambda-web.`*`AWS-Region`*`.on.aws`
- Globaler Endpunkt (`MultiRegion`Endpunkttyp) — Der Hostname endet auf `.lambda-web.global.on.aws`

Um Origin Access Control (OAC) mit einer Lambda-Webfunktion zu verwenden, muss der Authentifizierungstyp des Endpunkts sein. `IamAuth` Die ressourcenbasierte Richtlinie der Lambda-Webfunktion muss die Genehmigung ebenfalls zulassen. `lambda:InvokeWebFunctionEndpoint` CloudFront signiert Anfragen an regionale Endpunkte mit SigV4 und an globale Endpunkte mit SigV4a. Weitere Informationen finden Sie unter Beschränken Sie den Zugriff auf ein AWS Lambda Ursprung der Webfunktion.

Weitere Informationen zu Lambda Web Functions finden Sie in den folgenden Themen im AWS Lambda Developer Guide:

- AWS Lambda Webfunktionen — Ein allgemeiner Überblick über die Lambda-Webfunktions-Funktion
- Lambda Web Function Endpoints — Enthält Details zu den regionalen und globalen Endpunkttypen und deren Hostnamen
- Endpoint Access Control — Enthält Details zu den Authentifizierungstypen für Endgeräte und zur Signierung von Anfragen CloudFront

Verwenden von Amazon EC2 (oder eines anderen benutzerdefinierten Ursprungs)

Sie können sowohl interne als auch mit dem Internet verbundene EC2-Instances mit Amazon verwenden. CloudFront Sie können interne EC2-Instances in privaten Subnetzen verwenden, CloudFront indem Sie VPC-Ursprünge verwenden. CloudFront VPC-Ursprünge ermöglichen es Ihnen, Inhalte von Anwendungen bereitzustellen, die in privaten VPC-Subnetzen gehostet werden,

ohne sie dem öffentlichen Internet zugänglich zu machen. Weitere Informationen finden Sie unter Zugriffsbeschränkung mit VPC-Ursprüngen.

Ein benutzerdefinierter Ursprung ist ein HTTP(S)-Webserver mit einem öffentlich auflösbaren DNS-Namen, der Anforderungen von Clients über das Internet an Ziele weiterleitet. Der HTTP (S) - Server kann auf AWS— z. B. einer Amazon EC2 EC2-Instance — oder an einem anderen Ort gehostet werden. Ein als Website-Endpunkt konfigurierter Amazon S3-Ursprung gilt ebenfalls als benutzerdefinierter Ursprung. Weitere Informationen finden Sie unter the section called “Verwenden eines Amazon-S3-Buckets, der als Website-Endpunkt konfiguriert ist”.

Wenn Sie Ihren eigenen HTTP-Server als benutzerdefinierten Ursprung verwenden, geben Sie den DNS-Namen des Servers zusammen mit den HTTP- und HTTPS-Ports und dem Protokoll an, das Sie beim Abrufen von Objekten von Ihrem Ursprung verwenden CloudFront möchten.

Die meisten CloudFront Funktionen werden unterstützt, wenn Sie einen benutzerdefinierten Ursprung verwenden, mit Ausnahme von privaten Inhalten. Sie können zwar eine signierte URL verwenden, um Inhalte von einem benutzerdefinierten Ursprung zu verteilen, aber CloudFront um auf den benutzerdefinierten Ursprung zugreifen zu können, muss der Ursprung öffentlich zugänglich bleiben. Weitere Informationen finden Sie unter the section called “Beschränken von Inhalten mit signierten URLs und signierten Cookies”.

Folgen Sie diesen Richtlinien für die Verwendung von Amazon EC2 EC2-Instances und anderen benutzerdefinierten Ursprüngen mitCloudFront.

- Hosten Sie denselben Inhalt auf allen Servern bzw. stellen Sie ihn auf diesen bereit, die Inhalt für den gleichen CloudFront-Ursprung bereitstellen. Weitere Informationen finden Sie unter the section called “Ursprungseinstellungen” im Thema the section called “Alle Distributionseinstellungen”.
- Protokollieren Sie die `X-Amz-Cf-Id` Header-Einträge auf allen Servern, falls Sie diesen Wert für CloudFront das Debuggen benötigen Support oder verwenden möchten.
- Beschränken Sie Anforderungen von HTTP- und HTTPS-Ports, die Ihr benutzerdefinierter Ursprung überwacht.
- Synchronisieren Sie die Uhrzeit von allen Servern in der Implementierung. Beachten Sie, dass für signierte URLs und signierte Cookies, für Protokolle und Berichte die koordinierte Weltzeit (UTC) CloudFront verwendet wird. Beachten Sie außerdem, dass bei der Überwachung von CloudFront Aktivitäten mithilfe von CloudWatch Messwerten CloudWatch auch UTC verwendet wird.
- Verwenden Sie redundante Server für die Behandlung von Ausfällen.

- Informationen zur Verwendung eines benutzerdefinierten Ursprungs für die Bereitstellung privater Inhalte finden Sie unter the section called “Beschränken des Zugriffs auf Dateien auf benutzerdefinierten Ursprungsservern”.
- Informationen zu Anfrage- und Antwortverhalten sowie zu unterstützten HTTP-Statuscodes finden Sie unter Verhalten von Anfragen und Antworten.

Wenn Sie für einen benutzerdefinierten Ursprung Amazon EC2 verwenden, sollten Sie wie folgt vorgehen:

- Verwenden Sie ein Amazon Machine Image, das die Software für einen Webserver automatisch installiert. Weitere Informationen finden Sie in der Amazon EC2-Dokumentation.
- Verwenden Sie einen Elastic Load Balancing-Load Balancer, um Datenverkehr über mehrere Amazon EC2-Instances hinweg zu verarbeiten und Ihre Anwendung von Änderungen an Amazon EC2-Instances zu isolieren. Wenn Sie einen Load Balancer verwenden, können Sie beispielsweise Amazon EC2-Instances hinzufügen und löschen, ohne die Anwendung zu ändern. Weitere Informationen finden Sie im Elastic Load Balancing-Benutzerhandbuch.
- Wenn Sie Ihre CloudFront Distribution erstellen, geben Sie die URL des Load Balancers für den Domainnamen Ihres Ursprungsservers an. Weitere Informationen finden Sie unter the section called “Eine Distribution erstellen”.

Verwenden Sie CloudFront Ursprungsgruppen

Sie können eine Ursprungsgruppe für Ihren CloudFront Ursprung angeben, wenn Sie beispielsweise das Origin-Failover für Szenarien konfigurieren möchten, in denen Sie hohe Verfügbarkeit benötigen. Verwenden Sie das Origin-Failover, um einen primären Ursprung CloudFront sowie einen zweiten Ursprung festzulegen, der CloudFront automatisch zu diesem wechselt, wenn der primäre Ursprung bestimmte HTTP-Statuscode-Fehlerantworten zurückgibt.

Weitere Informationen, einschließlich der Schritte für die Einrichtung einer Ursprungsgruppe, finden Sie unter the section called “Verbessern der Verfügbarkeit mit Origin Failover”.

Verwenden von Amazon API Gateway

Sie können API Gateway als benutzerdefinierten Ursprung für Ihre CloudFront Distribution verwenden. Weitere Informationen finden Sie unter den folgenden Themen:

- Sicherung von Amazon API Gateway mit sicheren Chiffren mithilfe eines Amazon-Blogbeitrags CloudFront AWS
- Wie richte ich API Gateway mit meiner eigenen CloudFront Distribution ein? AWS re:Post

IPv6 für CloudFront Distributionen aktivieren

Amazon CloudFront unterstützt sowohl IPv4 als auch IPv6 von Clients bis hin zu AWS Edge-Standorten. CloudFront unterstützt auch IPv6- und Dual-Stack-Konnektivität (IPv4 und IPv6) zu Ursprüngen. Dies hilft Ihnen bei der durchgängigen IPv6-Übertragung.

IPv6 ist das Internetprotokoll der nächsten Generation, das IPv4 ersetzen soll. IPv4 verwendet 32-Bit-Adressen (wie z. B. 192.0.2.44), während IPv6 128-Bit-Adressen (wie z. B. 2001:0db8:85a3::8a2e:0370:7334) verwendet. IPv6 bietet einen erweiterten Adressraum für mehr mit dem Internet verbundene Geräte.

Themen

- IPv6-Viewer-Anforderungen
- IPv6-Ursprungsanforderungen

IPv6-Viewer-Anforderungen

Im Allgemeinen sollten Sie IPv6 aktivieren, wenn Sie Benutzer haben, die über IPv6-Netzwerke auf Ihre Inhalte zugreifen möchten. Aktivieren Sie IPv6 jedoch nicht, wenn beide der folgenden Bedingungen zutreffen:

- Sie verwenden signierte URLs oder signierte Cookies, um den Zugriff auf Ihre Inhalte einzuschränken.
- Sie verwenden eine benutzerdefinierte Richtlinie, die den `IpAddress` Parameter enthält, um einzuschränken, welche IP-Adressen auf Ihre Inhalte zugreifen können.

Wenn Sie den Zugriff auf bestimmte Inhalte nach IP-Adresse einschränken und den Zugriff auf andere Inhalte nicht einschränken möchten (oder den Zugriff einschränken möchten, jedoch nicht nach IP-Adresse), können Sie zwei Verteilungen erstellen. Informationen zum Erstellen von signierten URLs mit einer benutzerdefinierten Richtlinie finden Sie unter Erstellen einer signierten URL mit einer benutzerdefinierten Richtlinie. Informationen zum Erstellen von signierten Cookies

mit einer benutzerdefinierten Richtlinie finden Sie unter Einrichten signierter Cookies mit einer benutzerdefinierten Richtlinie.

Wenn Sie einen Route 53-Alias-Ressourcendatensatz verwenden, um den Verkehr an Ihre CloudFront Distribution weiterzuleiten, müssen Sie einen zweiten Alias-Ressourcendatensatz erstellen, wenn beide der folgenden Bedingungen zutreffen:

- Sie aktivieren IPv6 für die Verteilung
- Sie verwenden alternative Domainnamen in den URLs für Ihre Objekte

Weitere Informationen finden Sie unter Weiterleiten von Traffic an eine CloudFront Amazon-Distribution mithilfe Ihres Domainnamens im Amazon Route 53-Entwicklerhandbuch.

Wenn Sie einen CNAME-Ressourcendatensatz über Route 53 oder einen anderen DNS-Service erstellt haben, müssen Sie keine Änderungen ausführen. Ein CNAME-Datensatz leitet den Datenverkehr unabhängig von dem IP-Adressformat der Viewer-Anforderung an Ihre Verteilung weiter.

Wenn Sie IPv6 und CloudFront Zugriffsprotokolle aktivieren, enthält die `c-ip` Spalte Werte im IPv4- und IPv6-Format. Weitere Informationen finden Sie unter Felder der Protokolldatei.

> **Note**
>
> CloudFront Reagiert auf Zuschaueranfragen mit IPv4, wenn unsere Daten darauf hindeuten, dass IPv4 eine bessere Benutzererfahrung bietet, um eine hohe Kundenverfügbarkeit aufrechtzuerhalten. Um herauszufinden, wie viel Prozent der Anfragen CloudFront über IPv6 bedient werden, aktivieren Sie die CloudFront Protokollierung für Ihre Distribution und analysieren Sie die `c-ip` Spalte, die die IP-Adresse des Betrachters enthält, der die Anfrage gestellt hat. Dieser Prozentsatz sollte mit der Zeit höher werden, wird jedoch weiterhin nur einen kleinen Teil des Datenverkehrs umfassen, da IPv6 noch nicht von allen Viewer-Netzwerken weltweit unterstützt wird. Einige Viewer-Netzwerke verfügen über hervorragende IPv6-Unterstützung, aber andere unterstützen IPv6 jedoch überhaupt nicht. (Ein Viewer-Netzwerk ist mit dem Betreiber Ihres stationären Internets bzw. Ihrem Mobilfunkbetreiber vergleichbar.)
> Weitere Informationen zu unserer Unterstützung für IPv6 finden Sie unter CloudFront – Häufig gestellte Fragen. Informationen zur Aktivierung von Zugriffsprotokollen finden Sie in der Beschreibung der Felder Standardprotokollierung und Protokollpräfix.

IPv6-Ursprungsanforderungen

Wenn Sie einen benutzerdefinierten Ursprung (außer Amazon S3) verwenden, können Sie die Ursprungseinstellungen für Ihre Distribution anpassen. Mit diesen Einstellungen entscheiden Sie, wie Sie mithilfe von IPv4- oder IPv6-Adressen eine CloudFront Verbindung zu Ihrem Absender herstellen. Für benutzerdefinierte Ursprünge (außer Amazon S3) haben Sie die folgenden Verbindungsoptionen:

- Nur IPv4 (Standard) — Dies ist die Standardkonfiguration, die CloudFront verwendet wird, um Verbindungen zu Ursprüngen über IPv4 herzustellen.
- Nur IPv6 (ausgenommen Amazon S3- und VPC-Ursprünge) — Erfordert, dass Ihre Ursprungsdomain in eine IPv6-Adresse aufgelöst wird. CloudFront verwendet ausschließlich IPv6-Adressen für Ausgangsverbindungen.
- Dual-stack— Ermöglicht Verbindungen über IPv4 und IPv6. CloudFront wählt automatisch die ursprüngliche IPv4- oder IPv6-Konnektivität aus, um Leistung und Verfügbarkeit zu priorisieren, sodass Sie es CloudFront als IPv6- und IPv4-Dual-Stack-Internet-Gateway für Ihre Webanwendungen verwenden können.

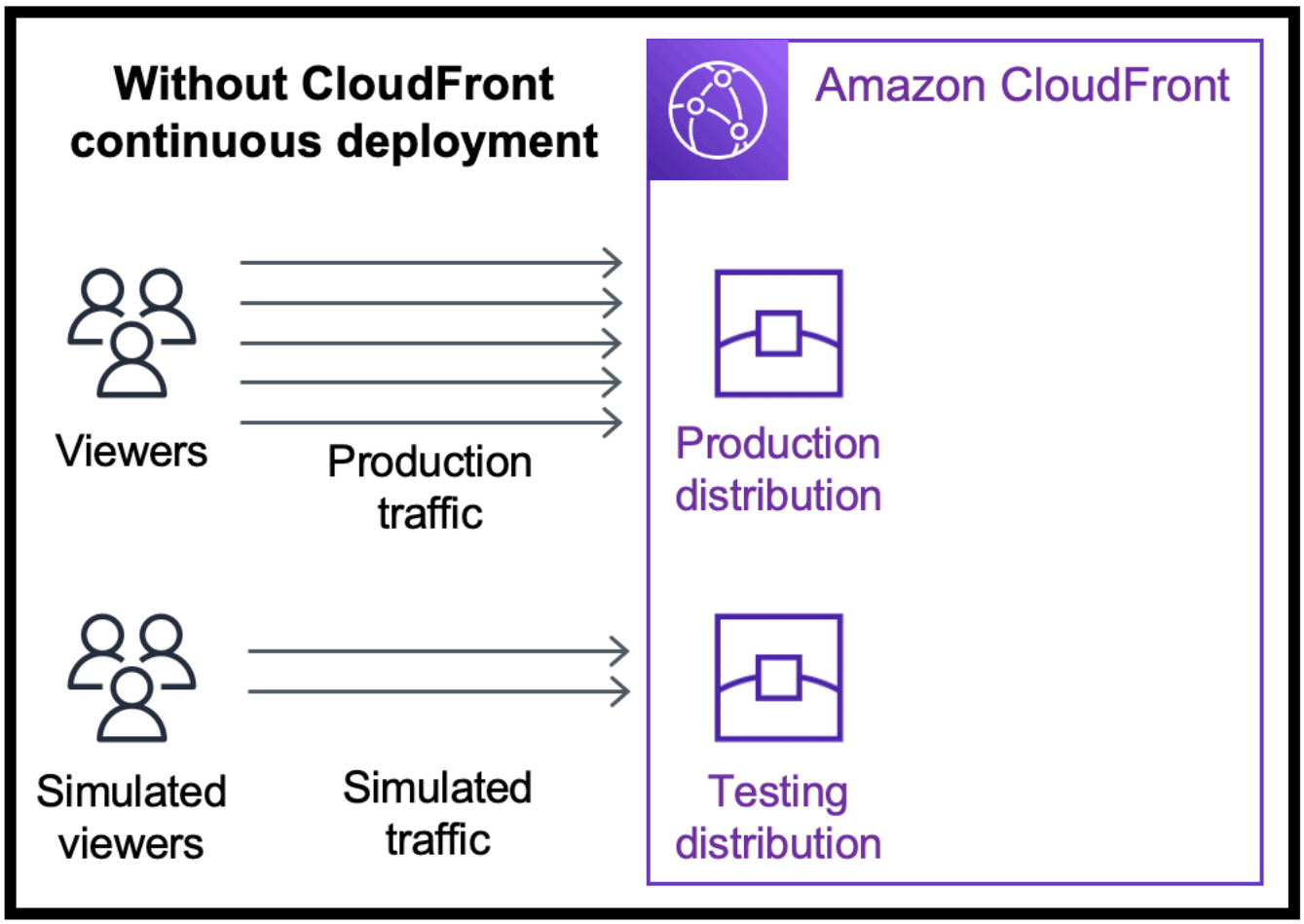
Wählen Sie die Option, die den Netzwerkkonfigurations- und Konnektivitätsanforderungen Ihres Ursprungsnetzwerks entspricht. Weitere Informationen finden Sie unter Entwerfen von DNS für IPv6 und Überlegungen zur Sicherheit und Überwachung von IPv6.

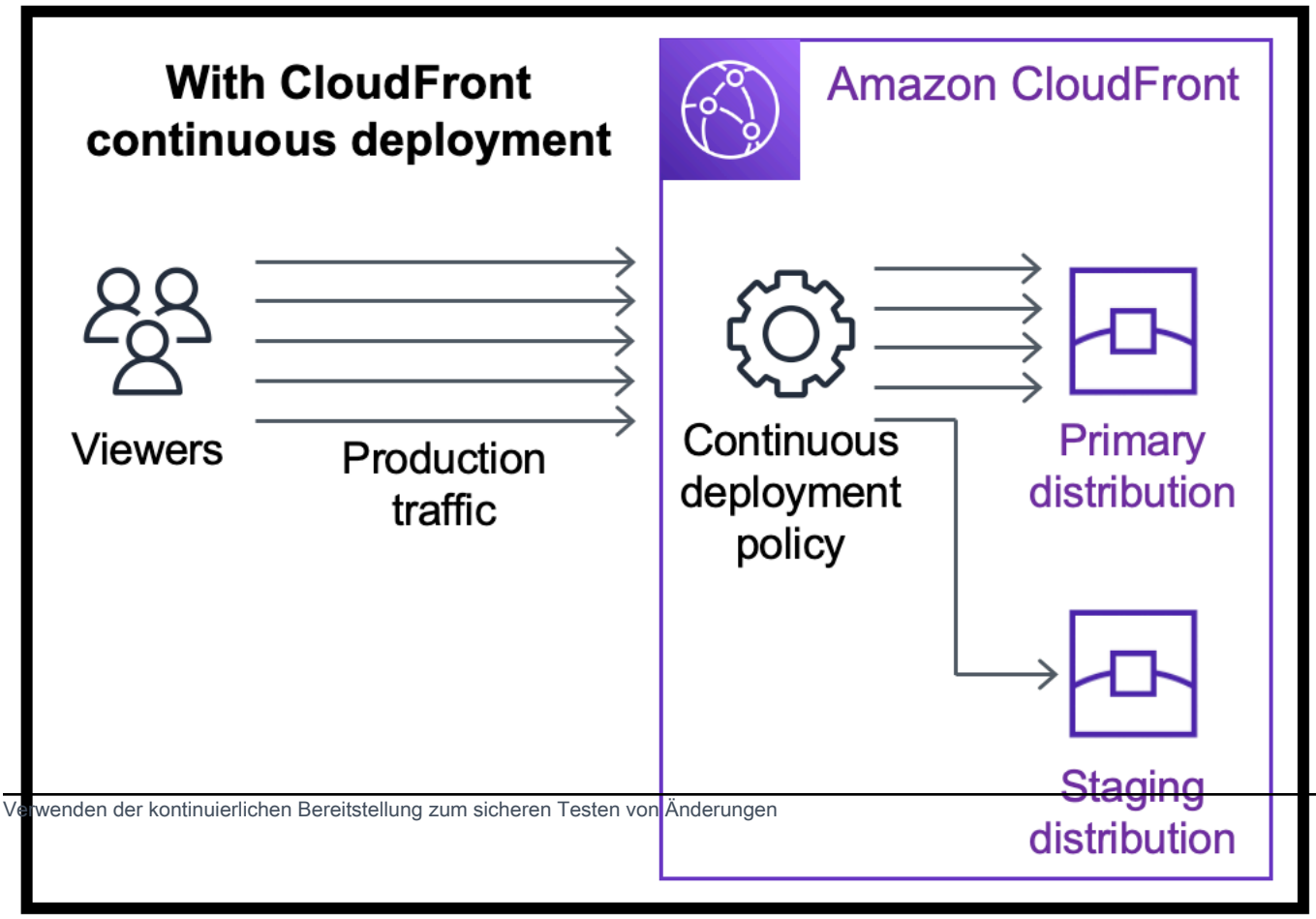
Verwenden Sie CloudFront Continuous Deployment, um CDN-Konfigurationsänderungen sicher zu testen

Mit Amazon CloudFront Continuous Deployment können Sie Änderungen an Ihrer CDN-Konfiguration sicher implementieren, indem Sie zunächst mit einer Teilmenge des Produktionsdatenverkehrs testen. Sie können eine Staging-Verteilung und eine Richtlinie für die kontinuierliche Bereitstellung verwenden, um einen Teil des Datenverkehrs von tatsächlichen Viewern (in der Produktion) an die neue CDN-Konfiguration zu senden und zu überprüfen, ob diese wie erwartet funktioniert. Sie können die Leistung der neuen Konfiguration in Echtzeit überwachen und die neue Konfiguration so hochstufen, dass sie den gesamten Datenverkehr bedient, indem Sie die Primärdistribution verwenden, wenn Sie bereit sind.

Das folgende Diagramm zeigt die Vorteile der CloudFront kontinuierlichen Bereitstellung. Ohne diese Funktionalität müssten Sie CDN-Konfigurationsänderungen mit simuliertem Datenverkehr testen. Bei Verwendung der kontinuierlichen Bereitstellung können Sie die Änderungen an einem Teil des

Produktionsdatenverkehrs testen und dann auf die primäre Verteilung hochstufen, wenn Sie dazu bereit sind.


Without CloudFront continuous deployment
Amazon CloudFront
Viewers
Production traffic
Production distribution
Simulated viewers
Simulated traffic
Testing distribution


With CloudFront continuous deployment
Amazon CloudFront
Viewers
Production traffic
Continuous deployment policy
Primary distribution
Staging distribution

In den folgenden Themen erfahren Sie mehr über die Arbeit mit kontinuierlicher Bereitstellung.

Themen

- CloudFront kontinuierlicher Bereitstellungs-Workflow
- Arbeiten mit einer Staging-Distribution und einer Richtlinie für die kontinuierliche Bereitstellung
- Überwachen einer Staging-Distribution
- Informationen zur kontinuierlichen Bereitstellung
- Kontingente und andere zu berücksichtigende Aspekte bei der kontinuierlichen Bereitstellung

CloudFront kontinuierlicher Bereitstellungs-Workflow

Der folgende allgemeine Arbeitsablauf erklärt, wie Sie Konfigurationsänderungen bei CloudFront kontinuierlicher Bereitstellung sicher testen und bereitstellen können.

1. Wählen Sie die Verteilung aus, die Sie als primäre Verteilung verwenden möchten. Die primäre Verteilung ist die Verteilung, die zurzeit für den Produktionsdatenverkehr zuständig ist.
2. Erstellen Sie von der primären Verteilung ausgehend eine Staging-Verteilung. Eine Staging-Verteilung ist zunächst eine Kopie der primären Verteilung.
3. Erstellen Sie eine Datenverkehrskonfiguration in einer Richtlinie für die kontinuierliche Bereitstellung und fügen Sie diese der primären Verteilung an. Dadurch wird bestimmt, wie CloudFront der Verkehr an die Staging-Verteilung weitergeleitet wird. Weitere Informationen zum Weiterleiten von Anforderungen an eine Staging-Verteilung finden Sie unter the section called “Weiterleiten von Anforderungen an die Staging-Distribution”.
4. Aktualisieren Sie die Konfiguration der Staging-Verteilung. Weitere Informationen zu den aktualisierbaren Einstellungen finden Sie unter the section called “Aktualisieren der primären Distribution und der Staging-Distribution”.
5. Überwachen Sie die Staging-Verteilung, um festzustellen, ob die Konfigurationsänderungen erwartungsgemäß funktionieren. Weitere Informationen zum Überwachen einer Staging-Verteilung finden Sie unter the section called “Überwachen einer Staging-Distribution”.

   Während der Überwachung der Staging-Verteilung können Sie folgende Aktionen ausführen:

   - Erneutes Aktualisieren der Konfiguration der Staging-Verteilung, um die Konfigurationsänderungen weiter zu testen
   - Aktualisieren der Richtlinie für die kontinuierliche Bereitstellung (Datenverkehrskonfiguration), um mehr oder weniger Datenverkehr an die Staging-Verteilung zu senden

6. Wenn Sie mit der Leistung der Staging-Verteilung zufrieden sind, können Sie die Konfiguration der Staging-Verteilung auf die primäre Verteilung hochstufen. Dabei wird die Konfiguration der Staging-Verteilung in die primäre Verteilung kopiert. Dadurch wird auch die Continuous Deployment Policy deaktiviert, was bedeutet, dass der gesamte CloudFront Datenverkehr an die primäre Verteilung weitergeleitet wird.

Sie können eine Automatisierung erstellen, die die Leistung der Staging-Verteilung überwacht (Schritt 5) und die Konfiguration automatisch hochstuft (Schritt 6), wenn bestimmte Kriterien erfüllt sind.

Nachdem Sie eine Konfiguration hochgestuft haben, können Sie die betreffende Staging-Verteilung erneut verwenden, wenn Sie das nächste Mal eine Konfigurationsänderung testen möchten.

Weitere Informationen zum Arbeiten mit Staging-Verteilungen und Richtlinien für die kontinuierliche Bereitstellung in der CloudFront Konsole AWS CLI, der oder der CloudFront API finden Sie im folgenden Abschnitt.

Arbeiten mit einer Staging-Distribution und einer Richtlinie für die kontinuierliche Bereitstellung

Sie können Staging-Verteilungen und Richtlinien für die kontinuierliche Bereitstellung in der CloudFront Konsole, mit der AWS Command Line Interface (AWS CLI) oder mit der CloudFront API erstellen, aktualisieren und ändern.

Erstellen einer Staging-Distribution und einer Richtlinie für die kontinuierliche Bereitstellung

Die folgenden Verfahren zeigen Ihnen, wie Sie mit einer Richtlinie für die kontinuierliche Bereitstellung eine Staging-Distribution erstellen.

Console

Sie können eine Staging-Distribution mit einer Richtlinie für die kontinuierliche Bereitstellung erstellen, indem Sie die AWS-Managementkonsole verwenden.

So erstellen Sie eine Staging-Verteilung und eine Richtlinie für die kontinuierliche Bereitstellung (Konsole)

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CloudFront Konsole unter. https://console.aws.amazon.com/cloudfront/v4/home
2. Rufen Sie im Navigationsbereich Distributions auf.
3. Wählen Sie die Verteilung aus, die Sie als primäre Verteilung verwenden möchten. Die primäre Verteilung ist die Verteilung, die zurzeit für den Produktionsdatenverkehr zuständig ist. Aus dieser Verteilung erstellen Sie die Staging-Verteilung.
4. Wählen Sie im Abschnitt Continuous deployment (Kontinuierliche Bereitstellung) die Option Create Staging Distribution (Staging-Verteilung erstellen) aus. Dadurch wird der Assistent Create Staging Distribution (Staging-Verteilung erstellen) geöffnet.
5. Führen Sie im Assistenten Create Staging Distribution (Create Staging Distribution) folgende Schritte aus:

   a. (Optional) Geben Sie eine Beschreibung für die Staging-Verteilung ein.

   b. Wählen Sie Weiter aus.

   c. Ändern Sie die Konfiguration der Staging-Verteilung. Weitere Informationen zu den aktualisierbaren Einstellungen finden Sie unter the section called “Aktualisieren der primären Distribution und der Staging-Distribution”.

      Wenn Sie mit den Änderungen an der Konfiguration der Staging-Verteilung fertig sind, wählen Sie Next (Weiter) aus.

   d. Geben Sie über die Konsole die Traffic configuration (Datenverkehrskonfiguration) an. Dadurch wird bestimmt, wie der CloudFront Datenverkehr an die Staging-Verteilung weitergeleitet wird. (CloudFront Speichert die Verkehrskonfiguration in einer Richtlinie für die kontinuierliche Bereitstellung.)

      Weitere Informationen zu den Optionen für die Datenverkehrskonfiguration finden Sie unter the section called “Weiterleiten von Anforderungen an die Staging-Distribution”.

      Wenn Sie die Traffic configuration (Datenverkehrskonfiguration) abgeschlossen haben, wählen Sie Next (Weiter) aus.

   e. Überprüfen Sie die Konfiguration der Staging-Verteilung, einschließlich der Datenverkehrskonfiguration, und wählen Sie dann Create Staging Distribution (Staging-Verteilung erstellen) aus.

Wenn Sie den Assistenten zum Erstellen einer Staging-Verteilung in der CloudFront Konsole abgeschlossen haben, CloudFront geht er wie folgt vor:

- Erstellen einer Staging-Verteilung mit den von Ihnen (in Schritt 5c) angegebenen Einstellungen
- Erstellen einer Richtlinie für die kontinuierliche Bereitstellung mit der von Ihnen (in Schritt 5d) angegebenen Datenverkehrskonfiguration
- Anfügen der Richtlinie für die kontinuierliche Bereitstellung an die primäre Verteilung, aus der Sie die Staging-Verteilung erstellt haben

Wenn die Konfiguration der Primärverteilung mit der beigefügten Continuous Deployment Policy an Edge-Standorten bereitgestellt wird, CloudFront beginnt das Senden des angegebenen Datenverkehrs auf der Grundlage der Verkehrskonfiguration an die Staging-Verteilung.

CLI

Gehen Sie wie folgt vor, um eine Staging-Verteilung und eine Richtlinie für die kontinuierliche Bereitstellung mit der AWS CLI zu erstellen.

So erstellen Sie eine Staging-Verteilung (CLI)

1. Verwenden Sie die Befehle aws cloudfront get-distribution und grep gemeinsam, um den `ETag`-Wert der Verteilung zu ermitteln, die Sie als primäre Verteilung verwenden möchten. Die primäre Verteilung ist die Verteilung, die zurzeit für den Produktionsdatenverkehr zuständig ist. Aus dieser Verteilung erstellen Sie die Staging-Verteilung.

   Im Folgenden wird ein Beispielbefehl gezeigt. Im folgenden Beispiel ersetzen Sie es *`primary_distribution_ID`* durch die ID der Primärdistribution.

   ```
   aws cloudfront get-distribution --id primary_distribution_ID | grep 'ETag'
   ```

   Kopieren Sie den `ETag`-Wert, da Sie ihn für den folgenden Schritt benötigen.
2. Verwenden Sie den Befehl aws cloudfront copy-distribution, um eine Staging-Verteilung zu erstellen. Im folgenden Beispielbefehl werden zur besseren Lesbarkeit Escape-Zeichen (\) und Zeilenumbrüche verwendet, Sie sollten diese jedoch im Befehl weglassen. Beachten Sie in dem Beispielbefehl Folgendes:

   - *`primary_distribution_ID`*Ersetzen Sie es durch die ID der Primärdistribution.

- *primary_distribution_ETag*Ersetzen Sie durch den `ETag` Wert der Primärverteilung (den Sie im vorherigen Schritt erhalten haben).
- (Optional) *CLI_example* Ersetzen Sie durch die gewünschte Anruferreferenz-ID.

```
aws cloudfront copy-distribution --primary-distribution-
id primary_distribution_ID \
                                 --if-match primary_distribution_ETag \
                                 --staging \
                                 --caller-reference 'CLI_example'
```

Die Ausgabe des Befehls enthält Informationen über die Staging-Verteilung und ihre Konfiguration. Kopieren Sie den CloudFront Domainnamen der Staging-Distribution, da Sie ihn für einen folgenden Schritt benötigen.

So erstellen Sie eine Richtlinie für die kontinuierliche Bereitstellung (CLI mit Eingabedatei)

1. Verwenden Sie den folgenden Befehl, um eine Datei mit dem Namen `continuous-deployment-policy.yaml` zu erstellen, die alle Eingabeparameter für den Befehl create-continuous-deployment-policy enthält. Im folgenden Befehl werden zur besseren Lesbarkeit Escape-Zeichen (\) und Zeilenumbrüche verwendet, Sie sollten diese jedoch im Befehl weglassen.

```
aws cloudfront create-continuous-deployment-policy --generate-cli-skeleton yaml-
input \
                                                    > continuous-deployment-
policy.yaml
```

2. Öffnen Sie die Datei mit dem Namen `continuous-deployment-policy.yaml`, die Sie gerade erstellt haben. Bearbeiten Sie die Datei, um die gewünschten Einstellungen für die Richtlinie für die kontinuierliche Bereitstellung anzugeben, und speichern Sie die Datei. Gehen Sie beim Bearbeiten der Datei wie folgt vor:

   - Im Abschnitt `StagingDistributionDnsNames`:
     - Ändern Sie den Wert von `Quantity` in 1.

- Fügen Sie für `Items` den CloudFront Domainnamen der Staging-Distribution (den Sie aus einem vorherigen Schritt gespeichert haben) ein.

- Im Abschnitt `TrafficConfig`:
  - Wählen Sie einen `Type` aus, entweder `SingleWeight` oder `SingleHeader`.
  - Entfernen Sie die Einstellungen für den anderen Typ. Wenn Sie beispielsweise eine gewichtete Datenverkehrskonfiguration wünschen, legen Sie für `Type` `SingleWeight` fest und entfernen Sie dann die Einstellungen für `SingleHeaderConfig`.
  - Um eine gewichtete Datenverkehrskonfiguration zu verwenden, legen Sie als Wert für `Weight` eine Dezimalzahl zwischen `.01` (ein Prozent) und `.15` (fünfzehn Prozent) fest.

  Weitere Informationen zu den Optionen in `TrafficConfig` finden Sie unter the section called “Weiterleiten von Anforderungen an die Staging-Distribution” und the section called “Sitzungs-Stickiness bei gewichtsbasierten Konfigurationen”.

3. Verwenden Sie den folgenden Befehl, um die Richtlinie für die kontinuierliche Bereitstellung unter Verwendung von Eingabeparametern aus der Datei `continuous-deployment-policy.yaml` zu erstellen.

```
aws cloudfront create-continuous-deployment-policy --cli-input-yaml file://
continuous-deployment-policy.yaml
```

   Kopieren Sie den `Id`-Wert in die Ausgabe des Befehls. Dies ist die ID der Richtlinie für die kontinuierliche Bereitstellung, die Sie in einem nachfolgenden Schritt benötigen werden.

So fügen Sie eine Richtlinie für die kontinuierliche Bereitstellung an eine primäre Verteilung an (CLI mit Eingabedatei)

1. Verwenden Sie den folgenden Befehl, um die Konfiguration der primären Verteilung in einer Datei namens `primary-distribution.yaml` zu speichern. *`primary_distribution_ID`*Ersetzen Sie ihn durch die ID der Primärdistribution.

```
aws cloudfront get-distribution-config --id primary_distribution_ID --output
 yaml > primary-distribution.yaml
```

2. Öffnen Sie die Datei mit dem Namen `primary-distribution.yaml`, die Sie gerade erstellt haben. Bearbeiten Sie die Datei und nehmen Sie die folgenden Änderungen vor:

- Fügen Sie die ID der Richtlinie für die kontinuierliche Bereitstellung (die Sie aus einem vorherigen Schritt kopiert haben) in das Feld `ContinuousDeploymentPolicyId` ein.
- Benennen Sie das Feld `ETag` in `IfMatch` um, ändern Sie jedoch nicht den Wert des Feldes.

Speichern Sie die Datei, wenn Sie fertig sind.

3. Verwenden Sie den folgenden Befehl, um die primäre Verteilung so zu aktualisieren, dass die Richtlinie für die kontinuierliche Bereitstellung verwendet wird. Ersetze es *`primary_distribution_ID`* durch die ID der Primärdistribution.

```
aws cloudfront update-distribution --id primary_distribution_ID --cli-input-yaml
 file://primary-distribution.yaml
```

Wenn die Konfiguration der Primärverteilung mit der beigefügten Continuous Deployment Policy an Edge-Standorten bereitgestellt wird, CloudFront beginnt, den angegebenen Teil des Datenverkehrs auf der Grundlage der Verkehrskonfiguration an die Staging-Verteilung zu senden.

API

Verwenden Sie die folgenden API-Operationen, um mit der CloudFront API eine Richtlinie für die Staging-Verteilung und kontinuierliche Bereitstellung zu erstellen:

- CopyDistribution
- CreateContinuousDeploymentPolicy

Weitere Informationen zu den Feldern, die Sie in diesen API-Aufrufen angeben, finden Sie:

- the section called “Weiterleiten von Anforderungen an die Staging-Distribution”
- the section called “Sitzungs-Stickiness bei gewichtsbasierten Konfigurationen”
- Die API-Referenzdokumentation für Ihr AWS SDK oder einen anderen API-Client

Nachdem Sie eine Staging-Verteilung und eine Continuous Deployment Policy erstellt haben, verwenden Sie UpdateDistribution(auf der Primärdistribution), um die Continuous Deployment Policy der Primärdistribution zuzuordnen.

Aktualisieren einer Staging-Distribution

Die folgenden Verfahren zeigen Ihnen, wie Sie mit einer Richtlinie für die kontinuierliche Bereitstellung eine Staging-Distribution aktualisieren.

Console

Sie können bestimmte Konfigurationen sowohl für die primäre Distribution als auch für die Staging-Distribution aktualisieren. Weitere Informationen finden Sie unter Aktualisieren der primären Distribution und der Staging-Distribution.

So aktualisieren Sie eine Staging-Verteilung (Konsole)

1. Öffnen Sie die CloudFront Konsole unter. https://console.aws.amazon.com/cloudfront/v4/home
2. Rufen Sie im Navigationsbereich Distributions auf.
3. Wählen Sie die primäre Verteilung aus. Dies ist die Verteilung, die zurzeit für den Produktionsdatenverkehr zuständig ist. Aus dieser Verteilung haben Sie die Staging-Verteilung erstellt.
4. Wählen Sie View Staging Distribution (Staging-Verteilung anzeigen) aus.
5. Ändern Sie die Konfiguration der Staging-Verteilung über die Konsole. Weitere Informationen zu den aktualisierbaren Einstellungen finden Sie unter the section called “Aktualisieren der primären Distribution und der Staging-Distribution”.

Sobald die Konfiguration der Staging-Verteilung an Edge-Standorten bereitgestellt ist, wird sie für den eingehenden Datenverkehr wirksam, der an die Staging-Verteilung weitergeleitet wird.

CLI

So aktualisieren Sie eine Staging-Verteilung (CLI mit Eingabedatei)

1. Verwenden Sie den folgenden Befehl, um die Konfiguration der Staging-Verteilung in einer Datei namens `staging-distribution.yaml` zu speichern. *`staging_distribution_ID`*Ersetzen Sie es durch die ID der Staging-Distribution.

```
aws cloudfront get-distribution-config --id staging_distribution_ID --output
 yaml > staging-distribution.yaml
```

2. Öffnen Sie die Datei mit dem Namen `staging-distribution.yaml`, die Sie gerade erstellt haben. Bearbeiten Sie die Datei und nehmen Sie die folgenden Änderungen vor:

   - Ändern Sie die Konfiguration der Staging-Verteilung. Weitere Informationen zu den aktualisierbaren Einstellungen finden Sie unter the section called “Aktualisieren der primären Distribution und der Staging-Distribution”.
   - Benennen Sie das Feld `ETag` in `IfMatch` um, ändern Sie jedoch nicht den Wert des Feldes.

   Speichern Sie die Datei, wenn Sie fertig sind.

3. Verwenden Sie den folgenden Befehl, um die Konfiguration der Staging-Verteilung zu aktualisieren. Ersetzen Sie es *`staging_distribution_ID`* durch die ID der Staging-Distribution.

```
aws cloudfront update-distribution --id staging_distribution_ID --cli-input-yaml
 file://staging-distribution.yaml
```

Sobald die Konfiguration der Staging-Verteilung an Edge-Standorten bereitgestellt ist, wird sie für den eingehenden Datenverkehr wirksam, der an die Staging-Verteilung weitergeleitet wird.

API

Um die Konfiguration einer Staging-Distribution zu aktualisieren, verwenden Sie UpdateDistribution(auf der Staging-Distribution), um die Konfiguration der Staging-Distribution zu ändern. Weitere Informationen zu den aktualisierbaren Einstellungen finden Sie unter the section called “Aktualisieren der primären Distribution und der Staging-Distribution”.

Aktualisieren einer Richtlinie für die kontinuierliche Bereitstellung

Die folgenden Verfahren zeigen Ihnen, wie Sie eine Richtlinie für die kontinuierliche Bereitstellung aktualisieren.

Console

Sie können die Datenverkehrskonfiguration Ihrer Distribution aktualisieren, indem Sie die Richtlinie für die kontinuierliche Bereitstellung aktualisieren.

So aktualisieren Sie eine Richtlinie für die kontinuierliche Bereitstellung (Konsole)

1. Öffnen Sie die Konsole unter CloudFront . https://console.aws.amazon.com/cloudfront/v4/home
2. Rufen Sie im Navigationsbereich Distributions auf.
3. Wählen Sie die primäre Verteilung aus. Dies ist die Verteilung, die zurzeit für den Produktionsdatenverkehr zuständig ist. Aus dieser Verteilung haben Sie die Staging-Verteilung erstellt.
4. Wählen Sie im Abschnitt Continuous deployment (Kontinuierliche Bereitstellung) Edit policy (Richtlinie bearbeiten) aus.
5. Ändern Sie die Datenverkehrskonfiguration in der Richtlinie für die kontinuierliche Bereitstellung. Klicken Sie auf Save changes (Änderungen speichern), wenn Sie fertig sind.

Wenn die Konfiguration der Primärverteilung mit der aktualisierten Continuous Deployment Policy an Edge-Standorten bereitgestellt wird, CloudFront beginnt das Senden von Datenverkehr an die Staging-Verteilung auf der Grundlage der aktualisierten Datenverkehrskonfiguration.

CLI

So aktualisieren Sie eine Richtlinie für die kontinuierliche Bereitstellung (CLI mit Eingabedatei)

1. Verwenden Sie den folgenden Befehl, um die Konfiguration der Richtlinie für die kontinuierliche Bereitstellung in einer Datei namens `continuous-deployment-policy.yaml` zu speichern. Ersetzen Sie es *`continuous_deployment_policy_ID`* durch die ID der Continuous Deployment Policy. Im folgenden Befehl werden zur besseren Lesbarkeit Escape-Zeichen (\) und Zeilenumbrüche verwendet, Sie sollten diese jedoch im Befehl weglassen.

```
aws cloudfront get-continuous-deployment-policy-config --
id continuous_deployment_policy_ID \
                                                    --output yaml >
 continuous-deployment-policy.yaml
```

2. Öffnen Sie die Datei mit dem Namen `continuous-deployment-policy.yaml`, die Sie gerade erstellt haben. Bearbeiten Sie die Datei und nehmen Sie die folgenden Änderungen vor:

- Ändern Sie die Konfiguration der Richtlinie für die kontinuierliche Bereitstellung wie gewünscht. Sie können beispielsweise von einer Header-basierten zu einer gewichteten Datenverkehrskonfiguration übergehen oder den Prozentsatz des Datenverkehrs (Gewichtung) durch eine gewichtete Konfiguration ersetzen. Weitere Informationen erhalten Sie unter the section called “Weiterleiten von Anforderungen an die Staging-Distribution” und the section called “Sitzungs-Stickiness bei gewichtsbasierten Konfigurationen”.
- Benennen Sie das Feld `ETag` in `IfMatch` um, ändern Sie jedoch nicht den Wert des Feldes.

Speichern Sie die Datei, wenn Sie fertig sind.

3. Verwenden Sie den folgenden Befehl, um die Richtlinie für die kontinuierliche Bereitstellung zu aktualisieren. *`continuous_deployment_policy_ID`*Ersetzen Sie es durch die ID der Richtlinie für die kontinuierliche Bereitstellung. Im folgenden Befehl werden zur besseren Lesbarkeit Escape-Zeichen (\) und Zeilenumbrüche verwendet, Sie sollten diese jedoch im Befehl weglassen.

```
aws cloudfront update-continuous-deployment-policy --
id continuous_deployment_policy_ID \
                                                    --cli-input-yaml file://
continuous-deployment-policy.yaml
```

Wenn die Konfiguration der Primärverteilung mit der aktualisierten Continuous Deployment Policy an Edge-Standorten bereitgestellt wird, CloudFront beginnt das Senden von Datenverkehr an die Staging-Verteilung auf der Grundlage der aktualisierten Verkehrskonfiguration.

API

Um eine Richtlinie für die kontinuierliche Bereitstellung zu aktualisieren, verwenden Sie. UpdateContinuousDeploymentPolicy

Hochstufen der Konfiguration einer Staging-Distribution

Die folgenden Verfahren zeigen Ihnen, wie Sie die Konfiguration einer Staging-Distribution hochstufen.

Console

Wenn Sie eine Staging-Distribution heraufstufen, wird die Konfiguration von der Staging-Distribution in die primäre Distribution CloudFront kopiert. CloudFront deaktiviert außerdem die Continuous Deployment Policy und leitet den gesamten Datenverkehr an die Primärverteilung weiter.

Nachdem Sie eine Konfiguration hochgestuft haben, können Sie die betreffende Staging-Verteilung erneut verwenden, wenn Sie das nächste Mal eine Konfigurationsänderung testen möchten.

So stufen Sie die Konfiguration einer Staging-Verteilung hoch (Konsole)

1. Öffnen Sie die CloudFront Konsole unter. https://console.aws.amazon.com/cloudfront/v4/home
2. Rufen Sie im Navigationsbereich Distributions auf.
3. Wählen Sie die primäre Verteilung aus. Dies ist die Verteilung, die zurzeit für den Produktionsdatenverkehr zuständig ist. Aus dieser Verteilung haben Sie die Staging-Verteilung erstellt.
4. Wählen Sie im Abschnitt Continuous deployment (Kontinuierliche Bereitstellung) Promote (Hochstufen) aus.
5. Geben Sie **`confirm`** ein und wählen Sie dann Promote (Hochstufen) aus.

CLI

Wenn Sie eine Staging-Distribution heraufstufen, CloudFront kopiert die Konfiguration aus der Staging-Distribution in die primäre Distribution. CloudFront deaktiviert außerdem die Continuous Deployment Policy und leitet den gesamten Datenverkehr an die Primärverteilung weiter.

Nachdem Sie eine Konfiguration hochgestuft haben, können Sie die betreffende Staging-Verteilung erneut verwenden, wenn Sie das nächste Mal eine Konfigurationsänderung testen möchten.

So stufen Sie die Konfiguration einer Staging-Verteilung hoch (CLI)

- Verwenden Sie den Befehl aws cloudfront update-distribution-with-staging-config, um die Konfiguration der Staging-Verteilung auf die primäre Verteilung hochzustufen. Im folgenden Beispielbefehl werden zur besseren Lesbarkeit Escape-Zeichen (\) und Zeilenumbrüche

verwendet, Sie sollten diese jedoch im Befehl weglassen. Beachten Sie in dem Beispielbefehl Folgendes:

- *primary_distribution_ID*Ersetzen Sie es durch die ID der Primärverteilung.
- *staging_distribution_ID*Ersetzen Sie es durch die ID der Staging-Distribution.
- Ersetzen Sie *primary_distribution_ETag* und *staging_distribution_ETag* durch die `ETag` Werte der Primär- und Stagingverteilung. Stellen Sie sicher, dass der Wert der primären Verteilung an erster Stelle steht, wie im Beispiel dargestellt.

```
aws cloudfront update-distribution-with-staging-config --
id primary_distribution_ID \
                                                    --staging-distribution-
id staging_distribution_ID \
                                                    --if-match
 'primary_distribution_ETag,staging_distribution_ETag'
```

API

Um die Konfiguration einer Staging-Distribution auf die Primärdistribution hochzustufen, verwenden Sie. UpdateDistributionWithStagingConfig

Überwachen einer Staging-Distribution

Um die Leistung einer Staging-Verteilung zu überwachen, können Sie dieselben Metriken, Protokolle und Berichte verwenden, die für alle CloudFront Verteilungen gelten. Beispiel:

- Sie können die CloudFrontStandard-Verteilungsmetriken (wie Gesamtzahl der Anfragen und Fehlerrate) in der CloudFront Konsole anzeigen und gegen Aufpreis zusätzliche Messwerte (wie Cache-Trefferrate und Fehlerrate nach Statuscode) aktivieren. Zudem können Sie Alarme basierend auf diesen Metriken erstellen.
- Sie können Standardprotokolle und Zugriffsprotokolle in Echtzeit einsehen, um detaillierte Informationen zu den Anfragen zu erhalten, die bei der Staging-Verteilung eingehen. Standardprotokolle enthalten die folgenden beiden Felder, anhand derer Sie die primäre Distribution identifizieren können, an die die Anforderung ursprünglich gesendet wurde, bevor sie

an die Staging-Verteilung CloudFront weitergeleitet wurde: und. `primary-distribution-id` `primary-distribution-dns-name`

- In der CloudFront Konsole können Sie Berichte anzeigen und herunterladen, z. B. den Cache-Statistikbericht.

Informationen zur kontinuierlichen Bereitstellung

In den folgenden Themen wird erklärt, wie CloudFront Continuous Deployment funktioniert.

Themen

- Weiterleiten von Anforderungen an die Staging-Distribution
- Sitzungs-Stickiness bei gewichtsbasierten Konfigurationen
- Aktualisieren der primären Distribution und der Staging-Distribution
- Primäre Verteilungen und Staging-Verteilungen nutzen nicht denselben Cache

Weiterleiten von Anforderungen an die Staging-Distribution

Wenn Sie CloudFront Continuous Deployment verwenden, müssen Sie nichts an den Viewer-Anfragen ändern. Viewer haben nicht die Möglichkeit, unter Verwendung eines DNS-Namens, einer IP-Adresse oder eines CNAME Anforderungen direkt an eine Staging-Verteilung zu senden. Stattdessen senden Zuschauer Anfragen an die primäre (Produktions-) Distribution und leiten einige dieser Anfragen auf CloudFront der Grundlage der Datenverkehrskonfigurationseinstellungen in der Continuous Deployment Policy an die Staging-Distribution weiter. Es gibt zwei Arten von Datenverkehrskonfigurationen:

Weight-based

Bei einer gewichtsbasierten Konfiguration wird der angegebene Prozentsatz der Viewer-Anforderungen an die Staging-Verteilung weitergeleitet. Wenn Sie eine gewichtsbasierte Konfiguration verwenden, können Sie auch die Sitzungsbindung aktivieren, wodurch sichergestellt wird, dass Anfragen von demselben Viewer als Teil einer einzigen Sitzung CloudFront behandelt werden. Weitere Informationen finden Sie unter the section called “Sitzungs-Stickiness bei gewichtsbasierten Konfigurationen”.

Header-based

Bei einer Header-basierten Konfiguration werden Anforderungen an die Staging-Verteilung weitergeleitet, wenn die Viewer-Anforderung einen bestimmten HTTP-Header enthält (den Header

und Wert geben Sie selbst an). Anforderungen, die den angegebenen Header und Wert nicht enthalten, werden an die primäre Verteilung weitergeleitet. Diese Konfiguration ist hilfreich, wenn Sie lokale Tests durchführen oder die Kontrolle über die Viewer-Anforderungen haben.

> **Note**
>
> Header, die an Ihre Staging-Verteilung weitergeleitet werden, müssen das Präfix `aws-cf-cd-` enthalten.

Sitzungs-Stickiness bei gewichtsbasierten Konfigurationen

Wenn Sie eine gewichtsbasierte Konfiguration verwenden, um Traffic an eine Staging-Verteilung weiterzuleiten, können Sie auch die Sitzungsbindung aktivieren, wodurch sichergestellt wird, dass Anfragen von demselben Viewer als eine einzelne Sitzung CloudFront behandelt werden. Wenn Sie Session Stickiness aktivieren, CloudFront wird ein Cookie gesetzt, sodass alle Anfragen desselben Viewers in einer einzigen Sitzung von einer Distribution bedient werden, entweder von der Primär- oder der Staging-Distribution.

Wenn Sie Sitzungs-Stickiness aktivieren, können Sie auch die Leerlaufdauer (idle duration) angeben. Wenn der Viewer für diese Zeit inaktiv ist (keine Anfragen sendet), läuft die Sitzung ab und CloudFront behandelt future Anfragen von diesem Viewer als neue Sitzung. Sie geben die Leerlaufdauer in Sekunden an. Möglich sind dabei Werte von 300 (fünf Minuten) bis 3 600 Sekunden (einer Stunde).

CloudFront Setzt in den folgenden Fällen alle Sitzungen (auch aktive) zurück und betrachtet alle Anfragen als neue Sitzung:

- Sie deaktivieren oder aktivieren die Richtlinie für die kontinuierliche Bereitstellung.
- Sie deaktivieren oder aktivieren die Einstellung für Sitzungs-Stickiness.

Aktualisieren der primären Distribution und der Staging-Distribution

Wenn einer primären Verteilung eine Richtlinie für die kontinuierliche Bereitstellung angefügt ist, sind die folgenden Konfigurationsänderungen sowohl für die primäre Verteilung als auch für die Staging-Verteilung verfügbar:

- Alle Einstellungen in Bezug auf das Cache-Verhalten, einschließlich des Standard-Cache-Verhaltens

- Alle Ursprungseinstellungen (Ursprünge und Ursprungsgruppen)
- Benutzerdefinierte Fehlerreaktionen (Fehlerseiten)
- Geografische Einschränkungen
- Standardstammobjekt
- Protokollierungseinstellungen
- Beschreibung (Kommentar)

Sie können auch externe Ressourcen aktualisieren, auf die in der Konfiguration einer Distribution verwiesen wird, z. B. eine Cache-Richtlinie, eine Response-Header-Richtlinie, eine CloudFront Funktion oder eine Lambda @Edge -Funktion.

Primäre Verteilungen und Staging-Verteilungen nutzen nicht denselben Cache

Primäre Verteilungen und Staging-Verteilungen nutzen nicht denselben Cache. Wenn die erste Anfrage an eine Staging-Distribution CloudFront gesendet wird, ist deren Cache leer. Wenn Anforderungen bei der Staging-Verteilung ankommen, beginnt diese mit dem Zwischenspeichern der Antworten (sofern entsprechend konfiguriert).

Kontingente und andere zu berücksichtigende Aspekte bei der kontinuierlichen Bereitstellung

CloudFront Für die kontinuierliche Bereitstellung gelten die folgenden Kontingente und weitere Überlegungen.

Kontingente

- Maximale Anzahl von Staging-Verteilungen pro AWS-Konto: 20
- Maximale Anzahl von Richtlinien für die kontinuierliche Bereitstellung pro AWS-Konto: 20
- Maximaler Prozentsatz des Datenverkehrs, den Sie bei einer gewichtsbasierten Konfiguration an eine Staging-Verteilung senden können: 15 %
- Mindest- und Höchstwerte für die Leerlaufdauer bei Sitzungs-Stickiness: 300–3 600 Sekunden

Weitere Informationen finden Sie unter Kontingente.

> **Note**
>
> Wenn Sie die kontinuierliche Bereitstellung verwenden und Ihre primäre Distribution mit OAC für den S3-Bucket-Zugriff eingerichtet ist, aktualisieren Sie Ihre S3-Bucket-Richtlinie, um den Zugriff für die Staging-Distribution zu ermöglichen. Beispiele für S3-Bucket-Richtlinien finden Sie unter the section called “Erteilen Sie die CloudFront Erlaubnis, auf den S3-Bucket zuzugreifen”.

AWS WAF Web-ACLs

Wenn Sie die kontinuierliche Bereitstellung für Ihre Distribution aktivieren, gelten die folgenden Überlegungen für AWS WAF:

- Sie können der Verteilung keine AWS WAF Web-Zugriffskontrollliste (ACL) zuordnen, wenn diese ACL der Verteilung zum ersten Mal zugeordnet wurde.
- Sie können eine AWS WAF Web-ACL nicht von der Verteilung trennen.

Bevor Sie die oben genannten Aufgaben ausführen können, müssen Sie die Richtlinie für die kontinuierliche Bereitstellung für Ihre Produktionsdistribution löschen. Dadurch wird auch die Staging-Distribution gelöscht. Weitere Informationen finden Sie unter Verwenden Sie AWS WAF Schutzmaßnahmen.

Fälle, in denen alle Anfragen an die primäre Distribution CloudFront gesendet werden

In bestimmten Fällen, z. B. in Zeiten hoher Ressourcenauslastung, werden CloudFront möglicherweise alle Anfragen an die primäre Verteilung gesendet, unabhängig davon, was in der Richtlinie für die kontinuierliche Bereitstellung festgelegt ist.

CloudFront sendet zu Spitzenzeiten alle Anfragen an die primäre Distribution, unabhängig davon, was in der Richtlinie für die kontinuierliche Bereitstellung festgelegt ist. Spitzenverkehr bezieht sich auf den Verkehr auf dem CloudFront Service und nicht auf den Verkehr auf Ihrer Distribution.

HTTP/3

Sie können Continuous Deployment nicht mit einer Distribution verwenden, die Folgendes unterstützt HTTP/3.

Verwenden von benutzerdefinierten URLs durch Hinzufügen von alternativen Domainnamen (CNAMEs)

Wenn Sie eine Distribution erstellen, CloudFront stellt sie einen Domainnamen bereit, z. B. d111111abcdef8.cloudfront.net. Anstatt diesen bereitgestellten Domainnamen zu verwenden, können Sie einen alternativen Domainnamen (auch als CNAME bezeichnet) verwenden.

Informationen zur Verwendung Ihres eigenen Domainnamens, wie z. B. www.example.com, finden Sie in den folgenden Themen:

Themen

- Voraussetzungen für die Verwendung von alternativen Domänennamen
- Einschränkungen bei der Verwendung alternativer Domänennamen
- Hinzufügen eines alternativen Domainnamens
- Verschieben eines alternativen Domainnamens
- Entfernen eines alternativen Domainnamens
- Verwenden von Platzhaltern in alternativen Domainnamen

Voraussetzungen für die Verwendung von alternativen Domänennamen

Wenn Sie einer Distribution einen alternativen Domainnamen wie www.example.com hinzufügen, gelten die folgenden Anforderungen: CloudFront

Alternative Domänennamen müssen Kleinbuchstaben verwenden

Alle alternative Domänennamen (CNAMEs) müssen Kleinbuchstaben verwenden.

Alternative Domainnamen müssen durch ein gültiges TLS-Zertifikat abgedeckt werden

Um einer Distribution einen alternativen Domainnamen (CNAME) hinzuzufügen, müssen Sie Ihrer CloudFront Distribution ein vertrauenswürdiges, gültiges TLS-Zertifikat beifügen, das den alternativen Domainnamen abdeckt. Dadurch wird sichergestellt, dass nur Personen, die Zugriff auf das Zertifikat Ihrer Domain haben, eine Verbindung zu CloudFront einem CNAME herstellen können, der sich auf Ihre Domain bezieht.

Ein vertrauenswürdiges Zertifikat ist ein Zertifikat, das von AWS Certificate Manager (ACM) oder einer anderen gültigen Zertifizierungsstelle (CA) ausgestellt wurde. Sie können ein

selbstsigniertes Zertifikat verwenden, um einen vorhandenen CNAME zu validieren, aber nicht für einen neuen CNAME. CloudFront unterstützt dieselben Zertifizierungsstellen wie Mozilla. Die aktuelle Liste finden Sie unter Liste der CA-Zertifikate für Mozilla. Informationen zu Zwischenzertifikaten von einer Drittanbieter-Zertifizierungsstelle finden Sie unter Zwischenzertifikate.

Um einen alternativen Domainnamen mithilfe des von Ihnen angehängten Zertifikats zu verifizieren, einschließlich alternativer Domainnamen, die Platzhalter enthalten, wird der alternative Name (SAN) auf dem Zertifikat CloudFront überprüft. Der alternative Domänenname, den Sie hinzufügen, muss vom SAN abgedeckt werden.

> **Note**
>
> Einer CloudFront Distribution kann jeweils nur ein Zertifikat angehängt werden.

Sie weisen Ihre Berechtigung zum Hinzufügen eines spezifischen alternativen Domänennamens zu Ihrer Distribution nach, indem Sie eine der folgenden Aktionen ausführen:

- Anfügen eines Zertifikats, das den alternativen Domänennamen enthält, z. B product-name.example.com.
- Anfügen eines Zertifikats, das mit dem Platzhalterzeichen * am Anfang eines Domänennamens beginnt, um mehrere Unterdomänen durch ein einziges Zertifikat abzudecken. Wenn Sie ein Platzhalterzeichen verwenden, können Sie mehrere Unterdomänen als alternative Domänennamen in CloudFront hinzufügen.

Die folgenden Beispiele zeigen, wie die Verwendung von Platzhalterzeichen in Domänennamen in einem Zertifikat funktioniert und Sie dazu berechtigt, spezifische alternative Domänennamen in CloudFront hinzuzufügen.

- Sie möchten marketing.example.com als einen alternativen Domänennamen hinzufügen. Sie listen in Ihrem Zertifikat den folgenden Domänennamen auf: *.example.com. Wenn Sie dieses Zertifikat anhängen CloudFront, können Sie einen beliebigen alternativen Domainnamen für Ihre Distribution hinzufügen, der den Platzhalter auf dieser Ebene ersetzt, einschließlich marketing.example.com. Sie können beispielsweise auch die folgenden alternativen Domänennamen hinzufügen:
  - product.example.com
  - api.example.com

Sie können jedoch keine alternativen Domänennamen auf höheren oder niedrigeren Ebenen als der Platzhalterebene hinzufügen. Beispielsweise können Sie nicht die alternativen Domänennamen example.com oder marketing.product.example.com hinzufügen.

- Sie möchten example.com als einen alternativen Domänennamen hinzufügen. Hierzu müssen Sie den Domänennamen example.com in dem Zertifikat auflisten, das Sie Ihrer Verteilung anfügen.
- Sie möchten marketing.product.example.com als einen alternativen Domänennamen hinzufügen. Dazu können Sie *.product.example.com im Zertifikat auflisten oder marketing.product.example.com selbst im Zertifikat auflisten.

Berechtigung zum Ändern der DNS-Konfiguration

Wenn Sie alternative Domainnamen hinzufügen, müssen Sie CNAME-Einträge erstellen, um DNS-Abfragen für die alternativen Domainnamen an Ihre Distribution weiterzuleiten. CloudFront Hierzu müssen Sie beim DNS-Serviceanbieter für die von Ihnen verwendeten alternativen Domänennamen die Berechtigung zum Erstellen von CNAME-Datensätzen besitzen. Das bedeutet in der Regel, dass Sie der Besitzer der Domänen sind. Es kann aber auch bedeuten, dass Sie eine Anwendung für den Besitzer der Domäne entwickeln.

Alternative Domänennamen und HTTPS

Wenn Sie möchten, dass Viewer in Verbindung mit alternativen Domänennamen HTTPS verwenden, sind zusätzliche Konfigurationsschritte erforderlich. Weitere Informationen finden Sie unter Verwenden von alternativen Domainnamen und HTTPS.

Einschränkungen bei der Verwendung alternativer Domänennamen

Beachten Sie die folgenden Einschränkungen bei der Verwendung alternativer Domänennamen:

Maximale Anzahl der alternativen Domänennamen

Informationen zur aktuell gültigen maximalen Anzahl von alternativen Domainnamen, die Sie einer Distribution hinzufügen können, oder zum Anfordern eines höheren Kontingents (früher als Limit bezeichnet) finden Sie unter Allgemeine Kontingente für Verteilungen.

Doppelte und sich überschneidende alternative Domänennamen

Sie können einer CloudFront Distribution keinen alternativen Domainnamen hinzufügen, wenn derselbe alternative Domainname bereits in einer anderen CloudFront Distribution existiert, auch wenn Ihnen die andere Distribution AWS-Konto gehört.

Sie können jedoch einen alternativen Domänennamen mit Platzhalterzeichen hinzufügen (z. B. *.example.com), der einen alternativen Domänennamen ohne Platzhalterzeichen enthält (d. h. sich mit diesem überschneidet), z. B. www.example.com. Wenn sich alternative Domainnamen in zwei Verteilungen überschneiden, CloudFront sendet die Anfrage an die Distribution mit der genaueren Namensübereinstimmung, unabhängig von der Verteilung, auf die der DNS-Eintrag verweist. Beispielsweise ist marketing.domain.com spezifischer als *.domain.com.

Wenn Sie bereits über einen DNS-Platzhaltereintrag verfügen, der auf eine CloudFront Verteilung verweist, und Sie beim Versuch, einen neuen CNAME mit einem genaueren Namen hinzuzufügen, einen falsch konfigurierten DNS-Fehler erhalten, finden Sie weitere Informationen unter. CloudFront gibt einen falsch konfigurierten DNS-Eintragsfehler zurück, wenn ich versuche, einen neuen CNAME hinzuzufügen

Domain-Fronting

CloudFront bietet Schutz vor Domain-Fronting, das auf verschiedenen Ebenen auftritt. AWS-Konto Dies ist ein Szenario, in dem ein nicht standardmäßiger Client eine TLS/SSL Verbindung zu einem Domainnamen in einem AWS-Konto herstellt und dann in einem anderen eine HTTPS-Anfrage für einen nicht verwandten Domainnamen stellt. AWS-Konto

Die TLS-Verbindung kann beispielsweise eine Verbindung zu www.example.com herstellen und anschließend eine Anfrage für www.example.org ausgeben.

CloudFront Führt die folgenden Prüfungen durch, um festzustellen, ob es sich bei einer Anfrage um eine Domänenanforderung handelt:

- Die SNI-Erweiterung entspricht dem `Host`-Header der HTTP-Anforderung.
- Das Zertifikat gehört zu derselben Distribution AWS-Konto wie die Distribution für die Anfrage
- Die HTTP-Anforderung `Host` wird durch das Zertifikat abgedeckt, das während des TLS-Handshakes bereitgestellt wird.

Wenn keine dieser Bedingungen erfüllt ist, wird CloudFront festgestellt, dass es sich bei der Anfrage um ein Domain-Fronting handelt. CloudFront lehnt die Anfrage mit der HTTP-Fehlerantwort 421 ab.

> **Note**
>
> Wenn der Client die SNI-Erweiterung nicht bereitstellt und stattdessen ein standardmäßiges *.cloudfront.net-Zertifikat CloudFront erhält, akzeptiert er die eingehenden Anfragen.

Wie CloudFront identifiziert die Verteilung für eine Anfrage

CloudFront identifiziert eine Verteilung für eine HTTP-Anfrage auf der Grundlage des `Host` Headers. CloudFront hängt nicht von der CloudFront IP-Adresse ab, mit der Sie eine Verbindung herstellen, oder davon, welcher SNI-Handshake während des TLS-Handshakes bereitgestellt wurde.

Wenn eine Anfrage CloudFront empfangen wird, verwendet sie den Wert des `Host` Headers, um die Anfrage der spezifischen Verteilung zuzuordnen.

Nehmen wir zum Beispiel an, Sie haben zwei Distributionen und Sie haben Ihre DNS-Konfiguration so aktualisiert, dass die alternativen Domainnamen an die folgenden Endpunkte weitergeleitet werden:

- primary.example.com verweist auf d111111primary.cloudfront.net
- secondary.example.com verweist auf d222222secondary.cloudfront.net

Wenn Sie eine Anfrage an stellen, https://primary.example.com aber den `Host` Header z. B. als secondary.example.com angeben, wird die Anfrage `curl https://primary.example.com -H "Host: secondary.example.com"` stattdessen an die sekundäre Distribution weitergeleitet.

Hinzufügen eines alternativen Domänennamens im obersten Knoten (Zone Apex) einer Domäne

Wenn Sie einer Distribution einen alternativen Domainnamen hinzufügen, erstellen Sie in der Regel einen CNAME-Eintrag in Ihrer DNS-Konfiguration, um DNS-Abfragen für den Domainnamen an Ihre Distribution weiterzuleiten. CloudFront Sie können jedoch keinen CNAME-Datensatz für den obersten Knoten eines DNS-Namespace (auch als Zone Apex bezeichnet) erstellen; das DNS-Protokoll lässt dies nicht zu. Wenn Sie beispielsweise den DNS-Namen example.com registriert haben, lautet der Zone Apex example.com. Sie können keinen CNAME-Datensatz für example.com erstellen, Sie können jedoch CNAME-Datensätze für www.example.com, newproduct.example.com und so weiter erstellen.

Wenn Sie Route 53 als DNS-Service verwenden, können Sie einen Alias-Ressourcendatensatz erstellen. Dieser hat im Vergleich zu CNAME-Datensätzen die folgenden Vorteile:

- Sie können einen Alias-Ressourcendatensatz für einen Domänennamen im obersten Knoten (example.com) erstellen.

- Sie können einen HTTPS-Eintrag für einen alternativen Domainnamen erstellen, um die Protokollaushandlung als Teil der DNS-Suche zu ermöglichen, sofern der Client dies unterstützt. Weitere Informationen finden Sie unter Create alias resource record set.
- Sie zahlen bei Verwendung eines Alias-Ressourcendatensatzes keine Gebühren für Route 53-Abfragen.

> **Note**
>
> Wenn Sie IPv6 aktivieren, müssen Sie zwei Alias-Ressourcendatensätze erstellen, einen für die Weiterleitung des IPv4-Datenverkehrs (A-Datensatz) und einen für die Weiterleitung des IPv6-Datenverkehrs (AAAA-Datensatz). Weitere Informationen finden Sie unter Aktivieren von IPv6 (Viewer-Anforderungen) im Thema Referenz für alle Distributionseinstellungen.

Weitere Informationen finden Sie unter Weiterleiten von Datenverkehr an eine CloudFront Amazon-Webdistribution mithilfe Ihres Domainnamens im Amazon Route 53-Entwicklerhandbuch.

Wenn Sie Route 53 nicht für Ihr DNS verwenden, können Sie statische Anycast-IP-Adressen anfordern, an die Apex-Domains wie example.com weitergeleitet werden. CloudFront Weitere Informationen finden Sie unter Anfordern statischer Anycast-IPs zur Verwendung für Zulassungslisten.

Hinzufügen eines alternativen Domainnamens

In der folgenden Aufgabenliste wird beschrieben, wie Sie mit der CloudFront Konsole Ihrer Distribution einen alternativen Domainnamen hinzufügen, sodass Sie in Ihren Links Ihren eigenen Domainnamen anstelle des CloudFront Domainnamens verwenden können. Informationen zur Aktualisierung Ihrer Distribution mithilfe der CloudFront API finden Sie unterKonfigurieren von Distributionen.

> **Note**
>
> Wenn Sie möchten, dass Viewer HTTPS in Verbindung mit Ihrem alternativen Domain-Namen verwenden, finden Sie weitere Informationen unter Verwenden von alternativen Domainnamen und HTTPS.

Bevor Sie beginnen: Sie müssen die folgenden Schritte ausführen, bevor Sie Ihre Verteilung aktualisieren, um einen alternativen Domänennamen hinzuzufügen:

- Registrieren Sie den Domänennamen bei Route 53 oder einem anderen Domänenregister.
- Rufen Sie ein TLS-Zertifikat von einer autorisierten Zertifizierungsstelle (CA) ab, die den Domainnamen umfasst. Fügen Sie das Zertifikat Ihrer Verteilung hinzu, um zu überprüfen, ob Sie berechtigt sind, die Domäne zu verwenden. Weitere Informationen finden Sie unter Voraussetzungen für die Verwendung von alternativen Domänennamen.

Hinzufügen eines alternativen Domainnamens

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home.
2. Wählen Sie die ID für die Distribution aus, die Sie aktualisieren möchten.
3. Wählen Sie auf der Registerkarte Allgemeines die Option Domain hinzufügen aus.
4. Geben Sie bis zu fünf zu bedienende Domains an.
5. Wählen Sie Weiter aus.
6. Wenn CloudFront Sie für ein TLS-Zertifikat kein vorhandenes AWS Certificate Manager (ACM) Zertifikat für Ihre Domain AWS-Konto in Ihrem finden, können Sie wählen `us-east-1` AWS-Region, ob Sie ein Zertifikat automatisch oder manuell in ACM erstellen möchten.
7. Wenn Ihr Zertifikat bereitgestellt wird, müssen Sie Ihre DNS-Datensätze bei Ihrem DNS-Anbieter aktualisieren, um die Domaininhaberschaft nachzuweisen. Die Einträge, die Sie in Ihren DNS-Einträgen vornehmen müssen, werden Ihnen in der CloudFront Konsole zur Verfügung gestellt.
8. Nachdem Sie Ihre DNS-Datensätze aktualisiert haben, wählen Sie Zertifikat validieren aus.
9. Wenn das Zertifikat validiert ist, wählen Sie Weiter.
10. Überprüfen Sie Ihre Änderungen und wählen Sie Domains hinzufügen aus.
11. Vergewissern Sie sich auf der Registerkarte General für die Verteilung, dass unter Distribution Status der Wert Deployed angezeigt wird. Wenn Sie versuchen, einen alternativen Domain-Namen zu verwenden, bevor die Aktualisierungen für Ihre Verteilung übertragen wurden, kann es sein, dass die Links, die Sie in den folgenden Schritten erstellen, nicht funktionieren.
12. Konfigurieren Sie den DNS-Dienst für den alternativen Domainnamen (z. B. www.example.com), um den Datenverkehr an den CloudFront Domainnamen für Ihre Distribution weiterzuleiten (z. B. d111111abcdef8.cloudfront.net). Die verwendete Methode ist davon abhängig, ob Sie Route 53 als DNS-Serviceanbieter oder einen anderen Anbieter für die Domäne verwenden. Weitere

Informationen finden Sie unter Fügen Sie Ihrer CloudFront Standarddistribution eine Domain hinzu.

Route 53

Erstellen Sie einen Alias-Ressourcendatensatz. Mit einem Alias-Ressourcendatensatz bezahlen Sie keine Gebühren für Route 53-Abfragen. Sie können auch einen Alias-Ressourcendatensatz für den Namen der Root-Domain (example.com) erstellen, was bei der Verwendung von DNS für CNAMEs nicht zulässig ist. Anweisungen zur Erstellung eines Alias-Ressourcendatensatzes finden Sie unter Weiterleiten von Datenverkehr an eine CloudFront Amazon-Webdistribution mithilfe Ihres Domainnamens im Amazon Route 53-Entwicklerhandbuch.

Optional können Sie einen HTTPS-Eintrag für einen alternativen Domainnamen erstellen, um die Protokollaushandlung als Teil der DNS-Suche zu ermöglichen, sofern der Client dies unterstützt.

So erstellen Sie einen Alias-Ressourcendatensatz mit einem HTTPS-Eintrag (optional)

1. Aktivieren Sie HTTP/2 oder HTTP/3 in Ihren CloudFront Vertriebseinstellungen. Weitere Informationen erhalten Sie unter Unterstützte HTTP-Versionen und Eine Verteilung aktualisieren.
2. Erstellen Sie in der Route 53-Konsole einen Alias-Ressourcendatensatz. Folgen Sie den Anweisungen zur Weiterleitung des Datenverkehrs an eine CloudFront Amazon-Webdistribution, indem Sie Ihr Domainnamen-Verfahren verwenden.
3. Erstellen Sie bei der Erstellung des Alias-Ressourcendatensatzes einen Alias-Datensatz mit dem Datensatztyp HTTPS.

Anderer DNS-Serviceanbieter

Verwenden Sie die von Ihrem DNS-Serviceanbieter bereitgestellte Methode, um einen CNAME-Eintrag für Ihre Domäne hinzuzufügen. Dieser neue CNAME-Eintrag leitet DNS-Anfragen von Ihrem alternativen Domainnamen (z. B. www.example.com) an den CloudFront Domainnamen für Ihre Distribution weiter (z. B. d111111abcdef8.cloudfront.net). Weitere Informationen finden Sie in der Dokumentation Ihres DNS-Serviceanbieters.

> **Important**
>
> Wenn Sie bereits einen CNAME-Eintrag für Ihren alternativen Domainnamen haben, aktualisieren Sie diesen Eintrag oder ersetzen Sie ihn durch einen neuen, der auf den Domainnamen für Ihre Distribution verweist. CloudFront

13. Wenn Sie `dig` oder ein ähnliches DNS-Tool verwenden, vergewissern Sie sich, dass die DNS-Konfiguration, die Sie im vorherigen Schritt erstellt haben, auf den Domänennamen für Ihre Verteilung verweist.

    Das folgende Beispiel zeigt eine `dig`-Anfrage für die Domäne images.example.com sowie den relevanten Teil der Antwort.

    ```
    PROMPT> dig www.example.com

    ; <<> DiG 9.3.3rc2 <<> www.example.com
    ;; global options: printcmd
    ;; Got answer:
    ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 15917
    ;; flags: qr rd ra; QUERY: 1, ANSWER: 9, AUTHORITY: 2, ADDITIONAL: 0

    ;; QUESTION SECTION:
    ;www.example.com.     IN    A

    ;; ANSWER SECTION:
    www.example.com. 10800 IN CNAME d111111abcdef8.cloudfront.net.
    ...
    ```

    Der Antwortbereich zeigt einen CNAME-Eintrag, der Anfragen für www.example.com an den CloudFront Distributionsdomänennamen d111111abcdef8.cloudfront.net weiterleitet. Wenn der Name auf der rechten Seite von der Domainname für Ihre Distribution ist, `CNAME` ist der CNAME-Eintrag korrekt konfiguriert. CloudFront Wird hier ein anderer Wert angezeigt, z. B. der Domänenname für Ihren Amazon-S3-Bucket, dann ist der CNAME-Datensatz nicht korrekt konfiguriert. Beginnen Sie in diesem Fall erneut mit Schritt 7 und korrigieren Sie den CNAME-Datensatz so, dass er auf den Domänennamen für Ihre Verteilung verweist.

14. Testen Sie den alternativen Domänennamen, indem Sie URLs mit Ihrem Domänennamen anstelle des CloudFront-Domänennamens für Ihre Verteilung besuchen.

15. Ändern Sie in Ihrer Anwendung die URLs für Ihre Objekte, sodass Ihr alternativer Domainname anstelle des Domainnamens Ihrer CloudFront Distribution verwendet wird.

Verschieben eines alternativen Domainnamens

Wenn Sie versuchen, einer Standarddistribution oder einem Distributionsmandanten einen alternativen Domainnamen hinzuzufügen, und der alternative Domainname bereits mit einer anderen Ressource verknüpft ist, erhalten Sie eine Fehlermeldung.

Sie erhalten beispielsweise die Fehlermeldung `CNAMEAlreadyExists` (Ein oder mehrere der von Ihnen angegebenen CNAMEs sind bereits mit einer anderen Ressource verknüpft), wenn Sie versuchen, www.example.com einer Standarddistribution oder einem Distributionsmandanten hinzuzufügen, dieser alternative Domainname jedoch bereits mit einer anderen Ressource verknüpft ist.

In diesem Fall möchten Sie möglicherweise den vorhandenen alternativen Domainnamen von einer Ressource in eine andere verschieben. Dies ist die Quelldistribution und die Zieldistribution. Sie können alternative Domainnamen zwischen beiden and/or Standardverteilungsmandanten verschieben.

Informationen zum Verschieben des alternativen Domainnamens finden Sie in folgenden Themen:

Themen

- Einrichten der Zielstandarddistribution oder des Zieldistributionsmandanten
- Suchen der Quellstandarddistribution oder des Quelldistributionsmandanten
- Einen alternativen Domänennamen verschieben

Einrichten der Zielstandarddistribution oder des Zieldistributionsmandanten

Bevor Sie einen alternativen Domainnamen verschieben können, müssen Sie die Zielressource einrichten. Dies ist die Zielstandarddistribution oder der Zieldistributionsmandant, auf die bzw. den Sie den alternativen Domainnamen verschieben.

Standard distribution

So richten Sie eine Zielstandarddistribution ein

1. Fordern Sie ein TLS-Zertifikat an. Dieses Zertifikat enthält den alternativen Domainnamen als Subject oder Subject Alternative Name (SAN) oder einen Platzhalter (*), der den alternativen Domainnamen abdeckt, den Sie verschieben möchten. Wenn Sie kein Zertifikat haben, können Sie eines von AWS Certificate Manager (ACM) oder von einer anderen Zertifizierungsstelle (CA) anfordern und es in ACM importieren.

   > **Note**
   >
   > Sie müssen das Zertifikat in der Region USA Ost (Nord-Virginia) (`us-east-1`) anfordern oder importieren.

   Weitere Informationen finden Sie unter Anfordern eines öffentlichen Zertifikats anhand der Konsole und Importieren eines Zertifikats im AWS Certificate Manager im Benutzerhandbuch für AWS Certificate Manager .

2. Wenn Sie die Zielstandarddistribution noch nicht erstellt haben, erstellen Sie jetzt eine. Ordnen Sie das Zertifikat im Rahmen der Erstellung der Standarddistribution dieser Standarddistribution zu. Weitere Informationen finden Sie unter Eine Distribution erstellen.

   Wenn Sie bereits über eine Zielstandarddistribution verfügen, verknüpfen Sie das Zertifikat mit der Standarddistribution. Weitere Informationen finden Sie unter Eine Verteilung aktualisieren.

3. Wenn Sie alternative Domainnamen innerhalb derselben Domain verschieben AWS-Konto, überspringen Sie diesen Schritt.

   Um einen alternativen Domainnamen von einem AWS-Konto auf einen anderen zu verschieben, müssen Sie in Ihrer DNS-Konfiguration einen TXT-Eintrag erstellen. Dieser Bestätigungsschritt trägt dazu bei, unbefugte Domainübertragungen zu verhindern. CloudFront verwendet diesen TXT-Eintrag, um Ihre Inhaberschaft des alternativen Domainnamens zu bestätigen.

   Erstellen Sie in Ihrer DNS-Konfiguration einen DNS-TXT-Eintrag, der den alternativen Domainnamen der Zielstandarddistribution zuordnet. Das Format des TXT-Eintrags kann je nach Domaintyp variieren.

- Geben Sie für Subdomains einen Unterstrich (_) vor dem alternativen Domainnamen an. Im Folgenden wird ein TXT-Beispieleintrag gezeigt.

  ```
  _www.example.com TXT d111111abcdef8.cloudfront.net
  ```

- Geben Sie für eine Apex-Domain (oder Root-Domain) einen Unterstrich und einen Punkt (`_.`) vor dem Domainnamen ein. Im Folgenden wird ein TXT-Beispieleintrag gezeigt.

  ```
  _.example.com TXT d111111abcdef8.cloudfront.net
  ```

Distribution tenant

So richten Sie den Zieldistributionsmandanten ein

1. Fordern Sie ein TLS-Zertifikat an. Dieses Zertifikat enthält den alternativen Domainnamen als Subject oder Subject Alternative Name (SAN) oder einen Platzhalter (*), der den alternativen Domainnamen abdeckt, den Sie verschieben möchten. Wenn Sie kein Zertifikat haben, können Sie eines von AWS Certificate Manager (ACM) oder von einer anderen Zertifizierungsstelle (CA) anfordern und es in ACM importieren.

   > **Note**
   >
   > Sie müssen das Zertifikat in der Region USA Ost (Nord-Virginia) (`us-east-1`) anfordern oder importieren.

   Weitere Informationen finden Sie unter Anfordern eines öffentlichen Zertifikats anhand der Konsole und Importieren eines Zertifikats im AWS Certificate Manager im Benutzerhandbuch für AWS Certificate Manager .

2. Wenn Sie den Zieldistributionsmandanten noch nicht erstellt haben, erstellen Sie jetzt einen. Beim Erstellen des Distributionsmandanten verknüpfen Sie das Zertifikat mit dem Distributionsmandanten. Weitere Informationen finden Sie unter Eine Distribution erstellen.

   Wenn Sie bereits über einen Zieldistributionsmandanten verfügen, verknüpfen Sie das Zertifikat mit dem Distributionsmandanten. Weitere Informationen finden Sie unter Hinzufügen einer Domain und eines Zertifikats (Distributionsmandant).

3. Wenn du alternative Domainnamen innerhalb derselben Domain verschiebst AWS-Konto, überspringe diesen Schritt.

Um einen alternativen Domainnamen von einem AWS-Konto auf einen anderen zu verschieben, müssen Sie in Ihrer DNS-Konfiguration einen TXT-Eintrag erstellen. Dieser Bestätigungsschritt trägt dazu bei, unbefugte Domainübertragungen zu verhindern, und CloudFront verwendet diesen TXT-Eintrag, um zu überprüfen, ob Sie der Eigentümer des alternativen Domainnamens sind.

Erstellen Sie in Ihrer DNS-Konfiguration einen DNS-TXT-Eintrag, der den alternativen Domainnamen dem Zieldistributionsmandanten zuordnet. Das Format des TXT-Eintrags kann je nach Domaintyp variieren.

- Geben Sie für Subdomains einen Unterstrich (_) vor dem alternativen Domainnamen an. Im Folgenden wird ein TXT-Beispieleintrag gezeigt.

  ```
  _www.example.com TXT d111111abcdef8.cloudfront.net
  ```

- Geben Sie für eine Apex-Domain (oder Root-Domain) einen Unterstrich und einen Punkt (_.) vor dem Domainnamen ein. Im Folgenden wird ein TXT-Beispieleintrag gezeigt.

  ```
  _.example.com TXT d111111abcdef8.cloudfront.net
  ```

Fahren Sie mit dem folgenden Thema fort, um die Quellstandarddistribution oder den Quelldistributionsmandanten zu finden, die bzw. der dem alternativen Domainnamen bereits zugeordnet ist.

Suchen der Quellstandarddistribution oder des Quelldistributionsmandanten

Bevor Sie einen alternativen Domainnamen von einer Distribution (Standard oder Mandant) in eine andere verschieben können, müssen Sie die Quelldistribution finden. Das ist die Ressource, der der alternative Domainname bereits zugeordnet ist. Wenn Sie die AWS-Konto ID sowohl der Quell- als auch der Zielvertriebsressourcen kennen, können Sie festlegen, wie der alternative Domainname verschoben werden soll.

Hinweise

- Es wird empfohlen, den ListDomainConflictsAPI-Vorgang zu verwenden, da er sowohl Standardverteilungen als auch Verteilungsmandanten unterstützt.
- Der ListConflictingAliasesAPI-Vorgang unterstützt nur Standardverteilungen.

Folgen Sie diesen Beispielen, um die Quelldistribution (Standard oder Mandant) zu finden.

list-domain-conflicts

> **Tip**
>
> - Für eine Standarddistribution benötigen Sie die Berechtigung `cloudfront:ListDomainConflicts` und `cloudfront:GetDistribution`.
> - Für einen Distributionsmandanten benötigen Sie die Berechtigung `cloudfront:ListDomainConflicts` und `cloudfront:GetDistributionTenant`.

Um **`List-Domain-Conflicts`** zu verwenden, um die Quell-Standarddistribution oder den Distributionsmandanten zu finden

1. Verwenden Sie den Befehl `list-domain-conflicts` wie im folgenden Beispiel gezeigt.

   a. Durch den Domainnamen *`www.example.com`* ersetzen.

   b. Geben Sie als `domain-control-validation-resource` die ID der Zielstandarddistribution oder des Zieldistributionsmandanten an, die bzw. den Sie zuvor eingerichtet haben. Sie benötigen eine Standarddistribution oder einen Distributionsmandanten mit einem zugeordneten Zertifikat, das die angegebene Domain abdeckt.

   c. Führen Sie diesen Befehl mit den Anmeldeinformationen aus, die sich in derselben Datei AWS-Konto wie der Ziel-Standardverteilungs- oder Distributionsmandant befinden.

   Anforderung

   In diesem Beispiel wird ein Distributionsmandant angegeben.

   ```
   aws cloudfront list-domain-conflicts \
   --domain www.example.com \
   --domain-control-validation-resource
    "DistributionTenantId=dt_2x9GhoK0TZRsohWzv1b9It8JABC"
   ```

   Antwort

Für jeden Domainnamen in der Befehlsausgabe wird Folgendes angezeigt:

- der Ressourcentyp, dem die Domain zugeordnet ist
- die Ressourcen-ID
- Die AWS-Konto ID, der die Ressource gehört

Die Ressourcen-ID und die Konto-ID sind teilweise ausgeblendet. Auf diese Weise können Sie die Standarddistribution oder den Distributionsmandanten Ihres Kontos identifizieren, ohne dass Informationen zu Distributionen Dritter preisgegeben werden.

```
{
    "DomainConflicts": [
        {
            "Domain": "www.example.com",
            "ResourceType": "distribution-tenant",
            "ResourceId": "***************ohWzv1b9It8JABC",
            "AccountId": "******112233"
        }
    ]
}
```

In der Antwort werden alle Domainnamen aufgeführt, die in Konflikt stehen oder sich mit dem angegebenen Namen überschneiden.

Beispiel

- Wenn Sie angeben *tenant1.example.com*, enthält die Antwort tenant1.example.com und den sich überschneidenden alternativen Platzhalter-Domänennamen (*.example.com, falls vorhanden).
- Wenn Sie angeben **.tenant1.example.com*, enthält die Antwort *.tenant1.example.com und alle alternativen Domainnamen, die von diesem Platzhalter abgedeckt werden (z. B. test.tenant1.example.com, dev.tenant1.example.com usw.).

2. Suchen Sie in der Antwort nach dem Quell-Standardverteilungs- oder Distributionsmandanten für den alternativen Domainnamen, den Sie verschieben möchten, und notieren Sie sich die ID. AWS-Konto
3. Vergleichen Sie die Konto-ID der Quellstandarddistribution oder des Quelldistributionsmandanten mit der Konto-ID, für die Sie im vorherigen Schritt die

Zielstandarddistribution oder den Zieldistributionsmandanten erstellt haben. Sie können dann feststellen, ob sich die Quelle und das Ziel im selben AWS-Konto befinden. Dies hilft Ihnen zu bestimmen, wie Sie den alternativen Domänennamen verschieben.

Weitere Informationen finden Sie unter list-domain-conflicts in der AWS Command Line Interface Reference.

list-conflicting-aliases (standard distributions only)

> **Tip**
>
> Sie müssen über die `cloudfront:GetDistribution`- und `cloudfront:ListConflictingAliases`-Berechtigungen für die Zielstandarddistribution verfügen.

Um die **`Quell-Standardverteilung anhand von Aliasnamen zu ermitteln, die in Konflikt`** stehen

1. Verwenden Sie den Befehl `list-conflicting-aliases` wie im folgenden Beispiel gezeigt.

   a. *`www.example.com`*Ersetzen Sie durch den alternativen Domainnamen und *`EDFDVBD6EXAMPLE`* durch die ID der Ziel-Standarddistribution, die Sie zuvor eingerichtet haben.

   b. Führen Sie diesen Befehl mit Anmeldeinformationen aus, die sich im selben AWS-Konto wie die Zielstandarddistribution befinden.

   Anforderung

   In diesem Beispiel wird eine Standarddistribution angegeben.

   ```
   aws cloudfront list-conflicting-aliases \
   --alias www.example.com \
   --distribution-id EDFDVBD6EXAMPLE
   ```

   Antwort

Für jeden alternativen Domainnamen in der Ausgabe des Befehls wird die ID der Standarddistribution angezeigt, mit der er verknüpft ist, und die AWS-Konto -ID, die die Standarddistribution besitzt. Die Standarddistributions- und Konto-IDs sind teilweise ausgeblendet, sodass Sie die Standarddistributionen und Konten identifizieren können, die Ihnen gehören, ohne dass die Informationen von Distributionen und Konten, die Sie nicht besitzen, offengelegt werden.

```
{
    "ConflictingAliasesList": {
        "MaxItems": 100,
        "Quantity": 1,
        "Items": [
            {
                "Alias": "www.example.com",
                "DistributionId": "*******EXAMPLE",
                "AccountId": "******112233"
            }
        ]
    }
}
```

In der Antwort werden die alternativen Domainnamen aufgeführt, die in Konflikt stehen oder sich mit dem angegebenen Namen überschneiden.

Beispiel

- Wenn Sie angeben*www.example.com*, enthält die Antwort www.example.com und den sich überschneidenden alternativen Platzhalter-Domänennamen (*.example.com), falls vorhanden.
- Wenn Sie angeben**.example.com*, enthält die Antwort *.example.com und alle alternativen Domainnamen, die von diesem Platzhalter abgedeckt werden (z. B. www.example.com, test.example.com, dev.example.com usw.).

2. Suchen Sie die Standardverteilung für den alternativen Domainnamen, den Sie verschieben, und notieren Sie sich die ID. AWS-Konto Vergleichen Sie diese Konto-ID mit der Konto-ID, in der Sie im vorherigen Schritt die Zielstandarddistribution erstellt haben. Sie können dann feststellen, ob diese beiden Standardverteilungen identisch sind AWS-Konto und wie der alternative Domainname verschoben werden soll.

Weitere Informationen finden Sie unter list-conflicting-aliases in der AWS Command Line Interface Reference.

Fahren Sie nun mit dem folgenden Thema zum Verschieben des alternativen Domainnamens fort.

Einen alternativen Domänennamen verschieben

Wählen Sie je nach Situation eine der folgenden Möglichkeiten, den alternativen Domänennamen zu verschieben:

Die Quell- und Zieldistributionen (Standard oder Mandant) befinden sich im selben AWS-Konto.

Verwenden Sie den update-domain-association-Befehl in der AWS Command Line Interface (AWS CLI), um den alternativen Domainnamen zu verschieben.

Dieser Befehl funktioniert für alle Verschiebungen mit demselben Konto, auch wenn der alternative Domainname eine Apex-Domain ist (auch als Stammdomain bezeichnet, z. B. example.com).

Die Quell- und Zieldistributionen (Standard oder Mandant) befinden sich in unterschiedlichen AWS-Konten.

Wenn Sie Zugriff auf die Quellstandarddistribution oder den Quelldistributionsmandanten haben, der alternative Domainname keine Apex-Domain ist und Sie nicht bereits einen Platzhalter verwenden, der sich mit diesem alternativen Domainnamen überschneidet, verwenden Sie zum Verschieben des alternativen Domainnamens einen Platzhalter. Weitere Informationen finden Sie unter the section called “Einen alternativen Domänennamen mit einem Platzhalter verschieben”.

Wenn Sie keinen Zugriff auf den Mandanten haben AWS-Konto , der die Quelle für die Standardverteilung oder -verteilung hat, können Sie versuchen, den alternativen Domainnamen mithilfe des update-domain-association Befehls zu verschieben. Die Quellstandarddistribution oder der Quelldistributionsmandant muss deaktiviert werden, bevor Sie den alternativen Domainnamen verschieben können. Weitere Informationen finden Sie unter the section called “Kontakt AWS Support um einen alternativen Domainnamen zu verschieben”.

> **Note**
>
> Sie können den associate-alias-Befehl verwenden, der allerdings nur Standarddistributionen unterstützt. Weitere Informationen finden Sie AssociateAliasin der Amazon CloudFront API-Referenz.

update-domain-association (standard distributions and distribution tenants)

So verwenden Sie **`update-domain-association`**, um einen alternativen Domainnamen zu verschieben

1. Verwenden Sie den Befehl `update-domain-association`, wie im folgenden Beispiel gezeigt.

   a. *`example.com`*Ersetzen Sie ihn durch den alternativen Domänennamen und geben Sie die ID des Zielmandanten für die Standardverteilung oder die Verteilung an.

   b. Führen Sie diesen Befehl mit den Anmeldeinformationen aus, die sich im selben AWS-Konto befinden wie die Zielstandarddistribution oder der Zieldistributionsmandant.

> **Beachten Sie die folgenden Einschränkungen**
>
> - Zusätzlich zur `cloudfront:UpdateDomainAssociation`-Berechtigung müssen Sie über die `cloudfront:UpdateDistribution`-Berechtigung verfügen, um eine Standarddistribution zu aktualisieren. Um einen Distributionsmandanten zu aktualisieren, müssen Sie über die `cloudfront:UpdateDistributionTenant`-Berechtigung verfügen.
> - Wenn sich die Quell- und Zieldistribution (Standard oder Mandant) unterscheiden AWS-Konten, muss die Quelle deaktiviert werden, bevor Sie die Domain verschieben können.
> - Die Zielverteilung muss wie in the section called “Einrichten der Zielstandarddistribution oder des Zieldistributionsmandanten” beschrieben eingerichtet werden.

Anforderung

```
aws cloudfront update-domain-association \
  --domain "www.example.com" \
  --target-resource DistributionTenantId=dt_9Fd3xTZq7Hl2KABC \
  --if-match E3UN6WX5ABC123
```

Antwort

```
{
    "ETag": "E7Xp1Y3N9DABC",
    "Domain": "www.example.com",
    "ResourceId": "dt_9Fd3xTZq7Hl2KABC"
}
```

Mit diesem Befehl wird der alternative Domainname aus der Quellstandarddistribution bzw. dem Quelldistributionsmandanten entfernt und zur Zielstandarddistribution bzw. zum Zieldistributionsmandanten hinzugefügt.

2. Nachdem die Zielverteilung vollständig bereitgestellt ist, aktualisieren Sie Ihre DNS-Konfiguration so, dass Ihr Domainname auf den CloudFront Routing-Endpunkt verweist. Ihr DNS-Eintrag würde beispielsweise Ihren alternativen Domainnamen (`www.example.com`) auf den CloudFront angegebenen Domainnamen d111111abcdef8.cloudfront.net verweisen. Wenn das Ziel ein Distributionsmandant ist, geben Sie den Endpunkt der Verbindungsgruppe an. Weitere Informationen finden Sie unter Verweisen Sie Domains auf CloudFront.

associate-alias (standard distributions only)

Um Associate-Alias zu verwenden, um einen alternativen Domainnamen zu verschieben

1. Verwenden Sie den Befehl `associate-alias`, wie im folgenden Beispiel gezeigt.

   a. *`www.example.com`*Ersetzen Sie ihn durch den alternativen Domainnamen und *`EDFDVBD6EXAMPLE`* durch die Standard-Zielvertriebs-ID.

   b. Führen Sie diesen Befehl mit Anmeldeinformationen aus, die sich im selben AWS-Konto wie die Zielstandarddistribution befinden.

Beachten Sie die folgenden Einschränkungen:

- Sie müssen über `cloudfront:AssociateAlias`- und `cloudfront:UpdateDistribution`-Berechtigungen für die Zielstandarddistribution verfügen.
- Wenn sich die Quell- und Ziel-Standarddistribution in derselben befinden AWS-Konto, benötigen Sie die entsprechenden `cloudfront:UpdateDistribution` Berechtigungen für die Quell-Standarddistribution.
- Wenn sich die Quell-Standardverteilung und die Ziel-Standardverteilung unterscheiden AWS-Konten, müssen Sie zuerst die Quell-Standardverteilung deaktivieren.
- Die Zielstandarddistribution muss wie in the section called “Einrichten der Zielstandarddistribution oder des Zieldistributionsmandanten” beschrieben eingerichtet werden.

Anforderung

```
aws cloudfront associate-alias \
--alias www.example.com \
--target-distribution-id EDFDVBD6EXAMPLE
```

Mit diesem Befehl wird der alternative Domainname aus der Quellstandarddistribution entfernt und zur Zielstandarddistribution hinzugefügt.

2. Nachdem die Zielstandarddistribution vollständig bereitgestellt wurde, aktualisieren Sie Ihre DNS-Konfiguration so, dass DNS-Eintrag des alternativen Domainnamens auf den Distributionsdomainnamen der Zielstandarddistribution verweist. Ihr DNS-Eintrag würde beispielsweise Ihren alternativen Domainnamen (`www.example.com`) auf den CloudFront angegebenen Domainnamen d111111abcdef8.cloudfront.net verweisen.

Weitere Informationen finden Sie unter dem Befehl associate-alias in der AWS CLI - Befehlsreferenz.

Einen alternativen Domänennamen mit einem Platzhalter verschieben

Wenn sich die Quellverteilung in einer anderen Verteilung AWS-Konto als der Zielverteilung befindet und die Quellverteilung aktiviert ist, können Sie einen Platzhalter verwenden, um den alternativen Domainnamen zu verschieben.

> **Note**
>
> Sie können keine Platzhalter verwenden, um eine Apex-Domäne zu verschieben (z. B. example.com). Um eine Apex-Domain zu verschieben, wenn sich die Quell- und Zieldistribution in unterschiedlichen AWS-Konten befinden, kontaktieren Sie Support. Weitere Informationen finden Sie unter the section called “Kontakt AWS Support um einen alternativen Domainnamen zu verschieben”.

Verwenden eines Platzhalters, um einen alternativen Domänennamen zu verschieben

> **Note**
>
> Dieser Prozess beinhaltet mehrere Aktualisierungen Ihrer Verteilungen. Warten Sie, bis jede Verteilung die letzte Änderung vollständig bereitgestellt hat, bevor Sie mit dem nächsten Schritt fortfahren.

1. Aktualisieren Sie die Zielverteilung, um einen alternativen Platzhalterdomänennamen hinzuzufügen, der den alternativen Domänennamen abdeckt, den Sie verschieben. Wenn der alternative Domänenname, den Sie verschieben, beispielsweise www.example.com lautet, fügen Sie der Zielverteilung den alternativen Domänennamen *.example.com hinzu. Zu diesem Zweck muss das SSL/TLS Zertifikat auf der Zieldistribution den Platzhalter-Domänennamen enthalten. Weitere Informationen finden Sie unter the section called “Eine Verteilung aktualisieren”.
2. Aktualisieren Sie die DNS-Einstellungen für den alternativen Domänennamen, sodass sie auf den Domänennamen der Zielverteilung verweisen. Wenn der alternative Domänenname, den Sie verschieben, beispielsweise www.example.com lautet, aktualisieren Sie den DNS-Eintrag für www.example.com, um den Datenverkehr an den Domänennamen der Zielverteilung weiterzuleiten (z. B. d111111abcdef8.cloudfront.net). .

> **Note**
>
> Auch nachdem Sie die DNS-Einstellungen aktualisiert haben, wird der alternative Domänenname weiterhin von der Quellverteilung bedient, da dort der alternative Domänenname derzeit konfiguriert ist.

3. Aktualisieren Sie die Quellverteilung, um den alternativen Domänennamen zu entfernen. Weitere Informationen finden Sie unter Eine Verteilung aktualisieren.
4. Aktualisieren Sie die Zielverteilung, um den alternativen Domänennamen hinzuzufügen. Weitere Informationen finden Sie unter Eine Verteilung aktualisieren.
5. Verwenden Sie dig (oder ein ähnliches DNS-Abfragetool), um zu überprüfen, ob der DNS-Eintrag für den alternativen Domänennamen in den Domänennamen der Zielverteilung aufgelöst wird.
6. (Optional) Aktualisieren Sie die Zielverteilung, um den alternativen Platzhalterdomänennamen zu entfernen.

Kontakt AWS Support um einen alternativen Domainnamen zu verschieben

Wenn sich Quell- und Zieldistribution unterscheiden AWS-Konten und Sie keinen Zugriff auf die Quelldistribution haben AWS-Konto oder die Quelldistribution nicht deaktivieren können, können Sie sich an uns wenden, Support um den alternativen Domainnamen zu verschieben.

Um Kontakt aufzunehmen Support um einen alternativen Domainnamen zu verschieben

1. Richten Sie eine Zielverteilung ein, einschließlich des DNS-TXT-Eintrags, der auf die Zielverteilung verweist. Weitere Informationen finden Sie unter Einrichten der Zielstandarddistribution oder des Zieldistributionsmandanten.
2. Wenden Sie sich an uns, Support um zu bitten, zu überprüfen, ob Sie der Eigentümer der Domain sind, und die Domain für Sie auf die neue CloudFront Distribution zu übertragen.
3. Nachdem die Zielverteilung vollständig bereitgestellt wurde, aktualisieren Sie Ihre DNS-Konfiguration, um den DNS-Eintrag des alternativen Domänennamens auf den Verteilungsdomänennamen der Zielverteilung zu verweisen.

Entfernen eines alternativen Domainnamens

Wenn Sie den Datenverkehr für eine Domain oder Subdomain nicht mehr an eine CloudFront Distribution weiterleiten möchten, folgen Sie den Schritten in diesem Abschnitt, um sowohl die DNS-Konfiguration als auch die CloudFront Verteilung zu aktualisieren.

Es ist wichtig, dass Sie die alternativen Domänennamen aus der Verteilung entfernen und Ihre DNS-Konfiguration aktualisieren. Auf diese Weise können Sie später Probleme vermeiden, wenn Sie den Domainnamen einer anderen CloudFront Distribution zuordnen möchten. Wenn ein alternativer Domänenname bereits einer Verteilung zugeordnet ist, kann er nicht mit einer anderen Verteilung eingerichtet werden.

> **Note**
>
> Wenn Sie den alternativen Domänennamen aus dieser Verteilung entfernen möchten, damit Sie ihn einer anderen Verteilung hinzufügen können, führen Sie die in Verschieben eines alternativen Domainnamens genannten Schritte aus. Wenn Sie stattdessen die hier beschriebenen Schritte ausführen (um eine Domain zu entfernen) und die Domain dann einer anderen Distribution hinzufügen, wird es einen bestimmten Zeitraum geben, in dem die Domain nicht mit der neuen Distribution verknüpft CloudFront wird, weil sie sich auf die Aktualisierungen an den Edge-Standorten ausbreitet.

So entfernen Sie einen alternativen Domänennamen aus einer Verteilung

1. Leiten Sie zunächst den Internet-Traffic für Ihre Domain an eine andere Ressource weiter, die nicht zu Ihrer CloudFront Distribution gehört, z. B. einen Elastic Load Balancing Load Balancer. Oder Sie können den DNS-Eintrag löschen, zu dem der Datenverkehr weitergeleitet CloudFront wird.

   Führen Sie einen der folgenden Schritte aus, je nach dem DNS-Service für Ihre Domäne:

   - Wenn Sie Route 53 verwenden, aktualisieren oder löschen Sie Aliasdatensätze oder CNAME-Datensätze. Weitere Informationen finden Sie unter Bearbeiten von Datensätzen oder Löschen von Datensätzen.
   - Wenn Sie einen anderen DNS-Serviceanbieter nutzen, verwenden Sie die Methode des DNS-Serviceanbieters, um den CNAME-Datensatz, der den Datenverkehr zu CloudFront

weiterleitet, zu aktualisieren oder zu löschen. Weitere Informationen finden Sie in der Dokumentation Ihres DNS-Serviceanbieters.

2. Nachdem Sie die DNS-Datensätze Ihrer Domäne aktualisiert haben, warten Sie, bis die Änderungen weitergegeben wurden und DNS-Resolver den Datenverkehr an die neue Ressource weiterleiten. Sie können feststellen, ob der Vorgang abgeschlossen ist, indem Sie einige Testlinks erstellen, die Ihre Domäne in der URL verwenden.
3. Melden Sie sich bei der AWS-Managementkonsole an, öffnen Sie die CloudFront Konsole unter und aktualisieren Sie Ihre CloudFront Distributionhttps://console.aws.amazon.com/cloudfront/v4/home, um den Domainnamen zu entfernen. Gehen Sie dazu wie folgt vor:

   a. Wählen Sie die ID für die Distribution aus, die Sie aktualisieren möchten.
   b. Wählen Sie auf der Registerkarte General die Option Edit aus.
   c. Entfernen Sie unter Alternative Domänennamen (CNAMEs) den (oder die) alternativen Domänennamen, den (oder die) Sie für Ihre Verteilung nicht mehr verwenden möchten.
   d. Wählen Sie Yes, Edit aus.

Verwenden von Platzhaltern in alternativen Domainnamen

Wenn Sie alternative Domänennamen hinzufügen, können Sie am Anfang eines Domänennamens das Platzhalterzeichen * verwenden, statt Unterdomänen einzeln hinzuzufügen. Beispiel: Mit dem alternativen Domänennamen von*.example.com können Sie alle Domänennamen in Ihren URLs verwenden, die mit example.com enden, z. B. www.example.com, product-name.example.com usw. Der Pfad eines Objekts ist derselbe, unabhängig von dem Domänennamen, z. B.:

- www.Beispiel. com/images/image.jpg
- Produktname.Beispiel. com/images/image.jpg
- marketing.produktname.beispiel. com/images/image.jpg

Beachten Sie diese Anforderungen bei alternativen Domänennamen, die Platzhalterzeichen enthalten.

- Am Anfang des alternativen Domänennamens muss ein Sternchen und ein Punkt stehen (*.).
- Sie können keine Platzhalter verwenden, um einen Teil eines Namens einer Subdomäne zu ersetzen, beispielsweise *domain.example.com.

- Sie können eine Subdomäne nicht in der Mitte eines Domänennamens ersetzen, beispielsweis: subdomain.*.example.com.
- Alle alternativen Domänennamen, einschließlich alternativer Domänennamen, die Platzhalter verwenden, müssen durch den Subject Alternative Name (SAN) auf dem Zertifikat abgedeckt sein.

Ein alternativer Domänenname mit einem Platzhalter wie *.example.com kann einen weiteren alternativen Domänennamen wie z. B. example.com enthalten.

WebSockets Mit CloudFront Distributionen verwenden

Amazon CloudFront unterstützt die Verwendung eines TCP-based Protokolls WebSocket, das nützlich ist, wenn Sie langlebige bidirektionale Verbindungen zwischen Clients und Servern benötigen. Eine persistente Verbindung ist bei Echtzeit-Anwendungen oft erforderlich. Zu den Szenarien, die Sie verwenden könnten, WebSockets gehören Social-Chat-Plattformen, Arbeitsbereiche für die Online-Zusammenarbeit, Spiele für mehrere Spieler und Dienste, die Datenfeeds in Echtzeit bereitstellen, wie z. B. Finanzhandelsplattformen. Bei der WebSocket Vollduplex-Kommunikation können Daten über eine Verbindung in beide Richtungen fließen.

WebSocket Die Funktionalität wird automatisch aktiviert, sodass sie mit jeder Distribution funktioniert. Um sie zu verwendenWebSockets, konfigurieren Sie im Cache-Verhalten, das Ihrer Distribution zugewiesen ist, eine der folgenden Optionen:

- Leiten Sie alle Viewer-Anforderungs-Header an Ihren Ursprung weiter. Sie können die Richtlinie „AllViewer Managed Origin Request“ verwenden.
- Leiten Sie die Anforderungs-Header `Sec-WebSocket-Key` und `Sec-WebSocket-Version` in Ihrer Ursprungsanforderungsrichtlinie ausdrücklich weiter.

Wie funktioniert das WebSocket Protokoll

Das WebSocket Protokoll ist ein unabhängiges TCP-based Protokoll, mit dem Sie einen Teil des Mehraufwands — und die potenziell erhöhte Latenz — von HTTP vermeiden können.

Um eine WebSocket Verbindung herzustellen, sendet der Client eine reguläre HTTP-Anfrage, die die Upgrade-Semantik von HTTP verwendet, um das Protokoll zu ändern. Der Server kann dann den Handshake abschließen. Die WebSocket Verbindung bleibt geöffnet und entweder der Client oder der Server können Datenframes aneinander senden, ohne jedes Mal neue Verbindungen herstellen zu müssen.

Standardmäßig verwendet das WebSocket Protokoll Port 80 für reguläre WebSocket Verbindungen und Port 443 für WebSocket Verbindungen über TLS. Die Optionen, die Sie für Ihre CloudFront Viewer-Protokollrichtlinien und wählen, Protokoll (nur benutzerdefinierte Ursprünge) gelten für WebSocket Verbindungen und auch für HTTP-Verkehr.

WebSocket Anforderungen

WebSocket Anfragen müssen RFC 6455 in den folgenden Standardformaten entsprechen.

Example Beispiel-Client-Anforderung

```
GET /chat HTTP/1.1
Host: server.example.com
Upgrade: websocket
Connection: Upgrade
Sec-WebSocket-Key: dGhlIHNhbXBsZSBub25jZQ==
Origin: https://example.com
Sec-WebSocket-Protocol: chat, superchat
Sec-WebSocket-Version: 13
```

Example Beispiel-Server-Antwort

```
HTTP/1.1 101 Switching Protocols
Upgrade: websocket
Connection: Upgrade
Sec-WebSocket-Accept: s3pPLMBiTxaQ9kYGzzhZRbK+xOo=
Sec-WebSocket-Protocol: chat
```

Wenn die WebSocket Verbindung durch den Client oder Server oder durch eine Netzwerkunterbrechung unterbrochen wird, wird erwartet, dass die Client-Anwendungen die Verbindung mit dem Server wieder aufnehmen.

Empfohlene Header WebSocket

Um unerwartete Probleme im Zusammenhang mit der Komprimierung bei der Verwendung zu vermeiden, empfehlen wir WebSockets, die folgenden Header in eine Origin-Request-Richtlinie aufzunehmen:

- `Sec-WebSocket-Key`
- `Sec-WebSocket-Version`

- `Sec-WebSocket-Protocol`
- `Sec-WebSocket-Accept`
- `Sec-WebSocket-Extensions`

> **Note**
>
> Unterstützt derzeit CloudFront nur WebSocket Verbindungen über das Protokoll. HTTP/1.1

Anfordern statischer Anycast-IPs zur Verwendung für Zulassungslisten

Sie können auch Ihren eigenen IP-Adressbereich verwenden, um ihn CloudFront anstelle von statischen Anycast-IPs zu verwenden. CloudFront-provided Weitere Informationen finden Sie unter Verwenden Sie IPAM CloudFront mit Ihrer eigenen IP-Adresse.

Sie können statische Anycast-IPs anfordern, um sie mit Ihren CloudFront Distributionen zu verwenden. Statische Anycast-IP-Listen können nur IPv4-IP-Adressen oder sowohl IPv4- als auch IPv6-IP-Adressen enthalten. Diese IP-Adressen sind für Sie bestimmt AWS-Konto und über mehrere geografische Regionen verteilt.

Sie können 21 statische Anycast-IP-Adressen anfordern, um sie bei Netzwerkanbietern auf die Zulassungsliste zu setzen und Viewern, die auf Ihre Anwendung zugreifen, Datengebühren zu erlassen. Alternativ können Sie diese statischen IP-Adressen innerhalb von Sicherheits-Firewalls für ausgehenden Datenverkehr verwenden, um den Datenaustausch mit zugelassenen Anwendungen zu kontrollieren. Statische Anycast-IP-Listen können mit einer oder mehreren Distributionen verwendet werden.

Wenn Sie das Routing von Apex-Domains (wie example.com) direkt an Ihre CloudFront Distributionen aktivieren möchten, können Sie für diesen Anwendungsfall 3 statische Anycast-IP-Adressen anfordern. Fügen Sie dann A-Einträge zu Ihrem DNS hinzu, auf die die Apex-Domain verweisen soll. CloudFront

Statische Anycast-IPs funktionieren mit Server Name Indication (SNI). Weitere Informationen finden Sie unter Verwenden von SNI für die Beantwortung von HTTPS-Anforderungen (funktioniert für die meisten Clients).

Voraussetzungen

Um statische Anycast-IP-Listen mit Ihrer CloudFront Distribution zu verwenden, müssen Sie für die Preisklasse für die Verteilung die Option Alle Edge-Standorte verwenden auswählen. Weitere Informationen zur Preisgestaltung finden Sie unter CloudFront – Preise.

Anfordern einer statischen Anycast-IP-Liste

Fordern Sie eine statische Anycast-IP-Liste an, die Sie mit Ihrer CloudFront Distribution verwenden möchten.

So fordern Sie eine statische Anycast-IP-Liste an

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home.
2. Wählen Sie im linken Navigationsbereich Statische IPs aus.
3. Wählen Sie unter Anfrage den Link, um den technischen CloudFront Support zu kontaktieren.
4. Geben Sie Ihre Workload-Informationen an (Anforderungs-Bytes pro Sekunde und Anforderungen pro Sekunde).
5. CloudFront Support Engineering prüft Ihre Anfrage. Das Überprüfungsverfahren kann bis zu zwei Tage dauern.

Nachdem Ihre Anforderung genehmigt wurde, können Sie eine statische Anycast-IP-Liste erstellen und sie einer oder mehreren Distributionen zuordnen.

Erstellen einer statischen Anycast-IP-Liste

Bevor Sie beginnen, fordern Sie eine statische Anycast-IP-Liste an, wie im vorherigen Abschnitt beschrieben.

So erstellen Sie eine statische Anycast-IP-Liste

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home.
2. Wählen Sie im linken Navigationsbereich Statische IPs aus.
3. Wählen Sie Anycast-IP-Liste erstellen aus.

4. Geben Sie unter Name einen Namen ein.
5. Wählen Sie für Anwendungsfälle für statische IP den entsprechenden Anwendungsfall aus.
6. Geben Sie für den IP-Adresstyp eine der folgenden Optionen an:
   - IPv4 — Ordnet eine Liste zu, die nur IPv4-Adressen enthält
   - Dualstack — Ordnet eine Liste mit IPv4- und IPv6-Adressen zu
7. Lesen Sie sich die Servicebedingungen und Preise durch und wählen Sie Absenden aus.

Nachdem Ihre statische IP-Liste erstellt wurde, können Sie die zugewiesenen IP-Adressen auf der Detailseite Ihrer statischen IP-Liste einsehen. Sie können der statischen IP-Liste auch Distributionen zuordnen.

Zuordnen einer statischen Anycast-IP-Liste zu einer vorhandenen Distribution

Bevor Sie beginnen, müssen Sie wie in den vorherigen Abschnitten beschrieben eine statische Anycast-IP-Liste anfordern und erstellen.

Stellen Sie sicher, dass die folgenden Verteilungseinstellungen mit Ihrer statischen Anycast-IP-Liste kompatibel sind:

- Preisklassehat die Einstellung Alle Edge-Standorte verwenden (beste Leistung).
- Wenn IPv6 aktiviert ist, können Sie eine statische Dual-Stack-Anycast-IP-Liste zuordnen. Eine statische Anycast-IP-Liste, die nur IPv4-Adressen enthält, kann keinen Distributionen zugeordnet werden, für die IPv6 aktiviert ist.

So ordnen Sie eine statische Anycast-IP-Liste einer vorhandenen Distribution zu

- Führen Sie eine der folgenden Aktionen aus:
  - Ordnen Sie die statische IP-Liste auf der Detailseite der statischen IP-Liste zu:
    1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Konsole unter. CloudFront https://console.aws.amazon.com/cloudfront/v4/home
    2. Wählen Sie im linken Navigationsbereich Statische IPs aus.
    3. Wählen Sie den Namen Ihrer statischen IP-Liste aus.
    4. Wählen Sie Distributionen zuordnen aus.

5. Wählen Sie eine oder mehrere Distributionen und anschließend Distributionen zuordnen aus.

- Ordnen Sie die statische IP-Liste auf der Detailseite der Distribution zu:

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home.
2. Wählen Sie im linken Navigationsbereich Distributionen aus.
3. Wählen Sie den Namen Ihrer Distribution aus.
4. Wählen Sie im Bereich Einstellungen auf der Registerkarte Allgemeines die Option Bearbeiten aus.
5. Wählen Sie für die Anycast-IP-Liste die statische Anycast-IP-Liste aus, die für diese Distribution verwendet werden soll.
6. Wählen Sie Änderungen speichern aus.

Zuordnen einer statischen Anycast-IP-Liste zu einer neuen Distribution

Bevor Sie beginnen, müssen Sie wie in den vorherigen Abschnitten beschrieben eine statische Anycast-IP-Liste anfordern und erstellen.

So ordnen Sie eine statische Anycast-IP-Liste einer neuen Distribution zu

- Legen Sie eine neue -Verteilung an. Weitere Informationen finden Sie unter Erstellen Sie eine CloudFront Distribution in der Konsole. Unter Einstellungen müssen Sie die folgenden Auswahlen treffen, um Ihre statische Anycast-IP-Liste verwenden zu können:
  - Wählen Sie unter Anycast-IP-Liste Ihre statische Anycast-IP-Liste aus der Dropdown-Liste aus.
  - Wählen Sie unter Preisklasse die Option Alle Edge-Standorte verwenden (beste Leistung) aus.
  - Hinweis: Wenn Ihre statische Anycast-IP nur IPv4 und nicht Dualstack verwendet, wählen Sie für IPv6 die Option Aus.

Schließen Sie die Erstellung der Distribution ab. Sie können je nach Bedarf andere Einstellungen konfigurieren, die für statische Anycast-IP-Listen nicht erforderlich sind.

Weitere Informationen zu Kontingenten im Zusammenhang mit statischen Anycast-IP-Listen finden Sie unter CloudFront Amazon-Endpunkte und Kontingente in der. Allgemeine AWS-Referenz

Ordnen Sie eine statische Anycast-IP-Liste einer Verbindungsgruppe zu

Bevor Sie beginnen, fordern Sie eine statische Anycast-IP-Liste an und erstellen Sie sie, wie in den vorherigen Abschnitten beschrieben.

So ordnen Sie eine statische Anycast-IP-Liste einer neuen Verbindungsgruppe zu

1. Stellen Sie sicher, dass Sie unter Einstellungen Verbindungsgruppen aktiviert haben.
2. Erstellen Sie eine Verbindungsgruppe. Weitere Informationen finden Sie unter Benutzerdefinierte Verbindungsgruppe erstellen.
3. Für Einstellungen müssen Sie die folgenden Auswahlen treffen, um Ihre statische Anycast-IP-Liste verwenden zu können.
   - Wählen Sie unter Anycast-IP-Liste Ihre statische Anycast-IP-Liste aus der Dropdown-Liste aus.
4. Beenden Sie die Erstellung Ihrer Verbindungsgruppe.

> **Note**
>
> Wenn Ihre statische Anycast-IP nur IPv4 und nicht Dualstack verwendet, wählen Sie für IPv6 die Option Aus.

Weitere Informationen zu Kontingenten im Zusammenhang mit statischen Anycast-IP-Listen finden Sie unter CloudFront Amazon-Endpunkte und Kontingente in der. Allgemeine Amazon Web Services-Referenz

Aktualisieren Sie eine statische Anycast-IP-Liste

Nachdem Sie Ihre statische Anycast-IP-Adresse erstellt und einer Distribution zugeordnet haben, können Sie den IP-Adresstyp Ihrer statischen Anycast-IP-Liste ändern.

Um eine statische Anycast-IP-Liste zu aktualisieren

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home.
2. Wählen Sie im linken Navigationsbereich Statische IPs aus.

3. Wählen Sie den Namen Ihrer statischen IP-Liste aus.
4. Wählen Sie Bearbeiten aus.
5. Geben Sie für den IP-Adresstyp eine der folgenden Optionen an:

   - IPv4 — Ordnet eine Liste zu, die nur IPv4-Adressen enthält
   - Dualstack — Ordnet eine Liste mit IPv4- und IPv6-Adressen zu

   > **Note**
   >
   > Sie können IPv4 nicht wählen, wenn Ihre zugehörige Distribution IPv6 bereits aktiviert hat. Deaktivieren Sie dazu IPv6, bevor Sie den IP-Adresstyp für Ihre statische Anycast-IP aktualisieren können. Weitere Informationen finden Sie unter IPv6 für CloudFront Distributionen aktivieren.

6. Wählen Sie Senden, um Ihre Änderungen zu speichern und die statische IP-Liste von Anycast zu aktualisieren.

Verwenden Sie IPAM CloudFront mit Ihrer eigenen IP-Adresse

Dieses Tutorial zeigt, wie Sie IPAM verwenden, um Ihre BYOIP-CIDRs für statische Anycast-IP-Listen zu verwalten. CloudFront

Themen

- Was ist BYOIP für statische Anycast-IPs?
- Warum sollten Sie diese Funktion verwenden?
- Voraussetzungen
- Schritt 1: Fordern Sie eine statische Anycast-IP-Liste an
- Schritt 2: Erstellen Sie eine statische Anycast-IP-Liste
- Schritt 3: Erstellen Sie eine CloudFront Distribution
- Schritt 4: Mit Ressourcen verknüpfen CloudFront
- Schritt 5: Bereiten Sie sich auf die Migration vor
- Schritt 6: Weltweit für CIDR werben

Was ist BYOIP für statische Anycast-IPs?

CloudFront unterstützt die Verwendung eigener IPv4- und IPv6-Adressen über BYOIP von IPAM für globale Dienste. Über die einheitliche Oberfläche von IPAM können Kunden mithilfe ihrer eigenen IP-Adressen (BYOIP) dedizierte IP-Adresspools erstellen und diese CloudFront Distributionen zuweisen. Gleichzeitig können sie das AWS weltweite Content Delivery Network für die Bereitstellung ihrer Anwendungen und Inhalte nutzen. Ihre IP-Adressen werden mithilfe von Anycast-Routing von mehreren CloudFront Edge-Standorten aus gleichzeitig beworben.

Warum sollten Sie diese Funktion verwenden?

Steuern Sie den Netzwerkzugriff in Zulassungslisten, um:

- Allow-list IP-Adressen mit Netzwerkanbietern, um Datengebühren für zugelassene Zuschauer zu vermeiden
- Konfigurieren Sie Sicherheits-Firewalls für ausgehende Verbindungen, um den Datenverkehr nur auf zugelassene Anwendungen zu beschränken

Vereinfachen Sie Abläufe und Migrationen

- Leiten Sie Apex-Domänen (example.com) direkt an weiter, CloudFront indem Sie A-Einträge hinzufügen, die auf Ihre statischen IPs verweisen
- Migrieren Sie von anderen CDNs, ohne die IP-Infrastruktur oder die Firewall-Konfigurationen zu aktualisieren
- Pflegen Sie bestehende IP-Zulassungslisten mit Partnern und Kunden
- Teilen Sie eine einzelne statische Anycast-IP-Liste für mehrere Distributionen CloudFront

Konsistentes Branding

- Behalten Sie Ihren bestehenden IP-Adressraum für ein einheitliches Branding bei, wenn Sie zu AWS

Voraussetzungen

Um statische IP-Listen von Anycast für Ihre CloudFront Distribution zu verwenden, müssen Sie für die Preisklasse für den Vertrieb die Option Alle Edge-Standorte verwenden auswählen. Weitere

Informationen zur Preisgestaltung finden Sie unter CloudFront – Preise. Für Bring Your Own IP (BYOIP) müssen Sie außerdem IPv6 für die Verteilung oder Verbindungsgruppe deaktivieren, wenn Sie BYOIP verwenden. IPv4-only Ordnen Sie für Dual-Stack-BYOIP eine statische Dual-Stack-Anycast-IP-Liste zu und aktivieren Sie IPv6 für die Verteiler- oder Verbindungsgruppe.

Gehen Sie vor dem Start wie folgt vor:

- IPAM-Setup: Weitere Informationen finden Sie unter Integrieren von IPAM in Konten und Erstellen eines IPAM.
- Domänenverifizierung: Überprüfen Sie die Domänenkontrolle.
- Erstellen Sie einen Pool auf oberster Ebene: Folgen Sie den Schritten 1 bis 2 unter Bringen Sie Ihr eigenes IPv4-CIDR zu IPAM and/or Bringen Sie Ihr eigenes IPv6-CIDR zu IPAM.
- Erstellen Sie einen IPAM-Pool mit dem Gebietsschema als global, damit Sie ihn verwenden können. CloudFront Weitere Informationen finden Sie unter Verwenden Sie Ihre eigene IP-Adresse für die CloudFront Verwendung von IPAM.

> **Note**
>
> Erfordert drei /24 and/or /48 IPv4-CIDR-Blöcke.

Schritt 1: Fordern Sie eine statische Anycast-IP-Liste an

Fordern Sie eine statische Anycast-IP-Liste an, die Sie mit Ihrer CloudFront Distribution verwenden können.

So fordern Sie eine statische Anycast-IP-Liste an

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home.
2. Wählen Sie im linken Navigationsbereich Statische IPs aus.
3. Wählen Sie unter Anfrage den Link, um den technischen CloudFront Support zu kontaktieren.
4. Geben Sie Ihre Workload-Informationen an (Anforderungs-Bytes pro Sekunde und Anforderungen pro Sekunde).
5. CloudFront Support Engineering prüft Ihre Anfrage. Das Überprüfungsverfahren kann bis zu zwei Tage dauern.

6. Nachdem Ihre Anforderung genehmigt wurde, können Sie eine statische Anycast-IP-Liste erstellen und sie einer oder mehreren Distributionen zuordnen.

Schritt 2: Erstellen Sie eine statische Anycast-IP-Liste

Bevor Sie beginnen, fordern Sie eine statische Anycast-IP-Liste an, wie im vorherigen Abschnitt beschrieben.

So erstellen Sie eine statische Anycast-IP-Liste

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home.
2. Wählen Sie im linken Navigationsbereich Statische IPs aus.
3. Wählen Sie Anycast-IP-Liste erstellen aus.
4. Geben Sie unter Name einen Namen ein.
5. Wählen Sie für statische IP-Anwendungsfälle BYOIP als Ihren Anwendungsfall aus.
6. Wählen Sie als IP-Adresstyp IPv4 oder Dualstack.
7. Wählen Sie für IPAM-Pool die IPAM-Pool (s), die Sie über IPAM bereitgestellt haben, und die CIDR-Gruppe (n) aus ihnen aus.

Die folgenden Schritte unterscheiden sich vom standardmäßigen regionalen BYOIP-Prozess und legen das Muster für globale Dienste fest:

AWS CLI

Installation oder Aktualisierung auf die neueste Version der AWS CLI. Weitere Informationen finden Sie im AWS Command Line Interface -Benutzerhandbuch.

1. Rufen Sie den IpamPoolArn IPAM-Pool ab, in dem Ihre CIDR-Blöcke bereitgestellt wurden. Weitere Informationen finden Sie unter Bringen Sie Ihr eigenes öffentliches IPv4-CIDR auf IPAM, indem Sie nur die AWS CLI verwenden oder Bringen Sie Ihr eigenes öffentliches IPv6-CIDR auf IPAM, indem Sie nur die CLI verwenden. AWS
2. Erstellen Sie eine Anycast-IP-Liste mit Ihren CIDR-Blöcken und Ihrer IPAM-Konfiguration:

   Für IPv4:

```
aws cloudfront create-anycast-ip-list \
```

```
    --name byoip-aip-1 \
    --ip-count 3 \
    --region us-east-1 \
    --ip-address-type ipv4 \
    --ipam-cidr-configs
 '[{"Cidr":"1.1.1.0/24","IpamPoolArn":"arn:aws:ec2::123456789012:ipam-pool/ipam-
pool-005d58a8aa8147abc"},
{"Cidr":"2.2.2.0/24","IpamPoolArn":"arn:aws:ec2::123456789012:ipam-pool/ipam-
pool-005d58a8aa8147abc"},
{"Cidr":"3.3.3.0/24","IpamPoolArn":"arn:aws:ec2::123456789012:ipam-pool/ipam-
pool-005d58a8aa8147abc"}]'
```

Für IPv6:

```
aws cloudfront create-anycast-ip-list \
    --name byoip-aip-dualstack \
    --ip-count 3 \
    --region us-east-1 \
    --ip-address-type dualstack \
    --ipam-cidr-configs
 '[{"Cidr":"1.1.1.0/24","IpamPoolArn":"arn:aws:ec2::123456789012:ipam-pool/ipam-
pool-005d58a8aa8147abc"},
{"Cidr":"2.2.2.0/24","IpamPoolArn":"arn:aws:ec2::123456789012:ipam-pool/ipam-
pool-005d58a8aa8147abc"},
{"Cidr":"3.3.3.0/24","IpamPoolArn":"arn:aws:ec2::123456789012:ipam-pool/ipam-
pool-005d58a8aa8147abc"},
{"Cidr":"2600:9000:a100::/48","IpamPoolArn":"arn:aws:ec2::123456789012:ipam-pool/
ipam-pool-0a3b7c9e2f41d6789"},
{"Cidr":"2600:9000:a200::/48","IpamPoolArn":"arn:aws:ec2::123456789012:ipam-pool/
ipam-pool-0a3b7c9e2f41d6789"},
{"Cidr":"2600:9000:a300::/48","IpamPoolArn":"arn:aws:ec2::123456789012:ipam-pool/
ipam-pool-0a3b7c9e2f41d6789"}]'
```

Note

Sie können die spezifische IP-Adresse nicht aus dem Pool auswählen. CloudFront wird das automatisch tun.

Schritt 3: Erstellen Sie eine CloudFront Distribution

Denn Sie können den Anweisungen folgen CloudFront, um eine Standardverteilung zu erstellen oder Mehrmandantenverteilungen zu verwenden.

Schritt 4: Mit Ressourcen verknüpfen CloudFront

- Ordnen Sie eine statische Anycast-IP-Liste einer vorhandenen Distribution zu
- Ordnen Sie eine statische Anycast-IP-Liste einer neuen Distribution zu
- Ordnen Sie eine statische Anycast-IP-Liste einer Verbindungsgruppe zu

Schritt 5: Bereiten Sie sich auf die Migration vor

Weitere Informationen finden Sie unter Schritt 4: Vorbereitung der Migration im Amazon VPC-Benutzerhandbuch.

Schritt 6: Weltweit für CIDR werben

Weitere Informationen finden Sie unter Schritt 5: Weltweit für CIDR werben im Amazon VPC-Benutzerhandbuch.

gRPC mit CloudFront Distributionen verwenden

Amazon CloudFront unterstützt gRPC, ein Open-Source-Framework für Remoteprozeduraufrufe (RPC), das darauf aufbautHTTP/2. gRPC bietet bidirektionales Streaming und ein binäres Protokoll, das Nutzlasten zwischenspeichert, wodurch es sich für Anwendungen eignet, die eine Kommunikation mit geringer Latenz erfordern.

CloudFront empfängt Ihre gRPC-Anfragen und leitet sie direkt an Ihre Ursprünge weiter. Sie können vier Arten von gRPC-Diensten als Proxy verwenden CloudFront :

- Unäres RPC
- Server-Streaming-RPC
- Client-Streaming-RPC
- Bi-directional RPC streamen

So funktioniert gRPC in CloudFront

Um gRPC zu konfigurieren CloudFront, legen Sie einen Ursprung, der einen gRPC-Dienst bereitstellt, als Ursprung Ihrer Distribution fest. Sie können Origins verwenden, die sowohl Nicht-gRPC- als auch gRPC-Dienste bereitstellen. CloudFront bestimmt, ob es sich bei der eingehenden Anfrage um gRPC gRPC-Anfrage oder eine HTTP/HTTPS Anfrage handelt, die auf dem `Content-Type` Header basiert. Wenn der `Content-Type` Header einer Anfrage den Wert von hat`application/grpc`, wird die Anfrage als gRPC-Anfrage betrachtet und CloudFront leitet die Anfrage an Ihren Ursprung weiter.

> **Note**
>
> Damit eine Distribution gRPC-Anfragen verarbeiten kann, sollten Sie eine der unterstützten HTTP-Versionen einschließen HTTP/2 und HTTP-Methoden zulassen, einschließlich`POST`. Ihr gRPC-Ausgangsendpunkt muss für die Unterstützung von HTTPS konfiguriert sein, da CloudFront nur Secure (HTTPS-based) gRPC-Verbindungen unterstützt werden. gRPC unterstützt nur Ende-zu-Ende-HTTPS. Wenn Sie einen benutzerdefinierten Ursprung verwenden, stellen Sie sicher, dass Ihre Protokolleinstellungen HTTPS unterstützen.

Um die gRPC-Unterstützung für Ihre Distribution zu aktivieren, führen Sie die folgenden Schritte aus:

1. Aktualisieren Sie das Cacheverhalten Ihrer Distribution, um HTTP-Methoden, einschließlich der `POST`-Methode, zuzulassen.
2. Wählen Sie nach Auswahl der `POST`-Methode das angezeigte gRPC-Kontrollkästchen aus.
3. Geben Sie HTTP/2als eine der unterstützten HTTP-Versionen an.

Weitere Informationen finden Sie unter den folgenden Themen:

- gRPC-Anfragen zulassen über HTTP/2
- GrpcConfigin der Amazon CloudFront API-Referenz

Da gRPC nur für API-Verkehr verwendet wird, der nicht zwischengespeichert werden kann, wirken sich Ihre Cache-Konfigurationen nicht auf gRPC-Anforderungen aus. Sie können eine Ursprungsanforderungsrichtlinie verwenden, um den gRPC-Anforderungen, die an Ihren gRPC-Ursprung gesendet werden, benutzerdefinierte Header hinzuzufügen. Sie können AWS WAF

with verwenden CloudFront , um den Zugriff auf Ihre gRPC-Distribution zu verwalten, Bots zu kontrollieren und Ihre gRPC-Anwendungen vor Web-Exploits zu schützen. CloudFront gRPC unterstützt CloudFront Funktionen.

Zusätzlich zum HTTPS-Status erhalten Sie zusammen mit Ihrer gRPC-Antwort den gRPC-Status. Eine Liste möglicher Werte für gRPC-Status finden Sie unter Statuscodes und ihre Verwendung in gRPC.

Hinweise

gRPC unterstützt die folgenden CloudFront Funktionen nicht:

- Benutzerdefinierte Fehlerantworten
- Origin-Failover wird mit gRPC nicht unterstützt, da gRPC die Methode verwendet. `POST` CloudFront führt nur dann einen Failover zum sekundären Ursprung durch, wenn die HTTP-Methode der Viewer-Anfrage`GET`,, `HEAD` oder ist. `OPTIONS`
- CloudFront leitet gRPC-Anfragen direkt an den Ursprung weiter und umgeht den Regional Edge Cache (REC). Da gRPC den REC umgeht, unterstützt gRPC weder Lambda@Edge noch Origin Shield.
- gRPC unterstützt keine Regeln für die Inspektion von AWS WAF Anforderungsstellen. Wenn Sie diese Regeln in der Web-ACL für eine Distribution aktiviert haben, ignoriert jede Anforderung, die gRPC verwendet, die Regeln zur Überprüfung des Anforderungstexts. Alle anderen AWS WAF -Regeln gelten weiterhin. Weitere Informationen finden Sie unter Aktivieren AWS WAF für Distributionen.

Arbeiten mit gemeinsam genutzten Ressourcen in CloudFront

Amazon CloudFront integriert sich in AWS Resource Access Manager (AWS RAM), um die gemeinsame Nutzung von Ressourcen zu ermöglichen. AWS RAM ermöglicht es Ihnen, einige CloudFront Ressourcen mit anderen AWS-Konten oder über zu teilen AWS Organizations. Mit AWS RAM geben Sie Ressourcen in Ihrem Besitz frei, indem Sie eine Ressourcenfreigabe erstellen. Eine Ressourcenfreigabe legt die freizugebenden Ressourcen und die Konsumenten fest, für die sie freigegeben werden sollen. Zu den Verbrauchern können folgende Angaben zählen:

- AWS-Konten Spezifisch innerhalb oder außerhalb seiner Organisation in AWS Organizations
- Eine Organisationseinheit innerhalb ihrer Organisation in AWS Organizations
- Es ist die gesamte Organisation in AWS Organizations

Weitere Informationen dazu AWS RAM finden Sie im AWS RAM Benutzerhandbuch.

In diesem Thema wird erklärt, wie Sie Ressourcen teilen, deren Eigentümer Sie sind, und wie Sie Ressourcen verwenden, die mit Ihnen gemeinsam genutzt werden.

Inhalt

- Voraussetzungen für die gemeinsame Nutzung von Ressourcen
- Gemeinsame Nutzung eines VPC-Ursprungs
- Verwenden eines gemeinsam genutzten VPC-Ursprungs
- Identifizierung eines gemeinsamen VPC-Ursprungs
- Die gemeinsame Nutzung eines VPC-Ursprungs aufheben
- Zuständigkeiten und Berechtigungen für gemeinsam genutzte VPC-Ursprünge
- Fakturierung und Messung
- Kontingente für gemeinsam genutzte Ressourcen

Voraussetzungen für die gemeinsame Nutzung von Ressourcen

- Sie müssen über die AWSRAMDefaultPermissionCloudFront verwaltete Richtlinie verfügen, um schreibgeschützten Zugriff auf die Ressourcenfreigabe zu gewähren. Weitere Informationen finden Sie unter AWSRAMDefaultPermissionCloudFront.
- Um einen VPC-Ursprung gemeinsam nutzen zu können, müssen Sie ihn in Ihrem besitzen. AWS-Konto Das bedeutet, dass die Ressource Ihrem Konto zugewiesen oder bereitgestellt werden muss. Sie können eine Ressource, die mit Ihnen geteilt wurde, nicht teilen.
- Um eine Ressource für Ihre Organisation oder eine Organisationseinheit in freizugeben AWS Organizations, müssen Sie das Teilen mit aktivieren AWS Organizations. Weitere Informationen finden Sie unter Freigabe für AWS Organizations aktivieren im AWS RAM -Benutzerhandbuch.

Gemeinsame Nutzung eines VPC-Ursprungs

> **Note**
>
> CloudFront Unterstützt derzeit die gemeinsame Nutzung von VPC-Ursprüngen. Falls Sie noch keinen erstellt haben, finden Sie weitere Informationen unterZugriffsbeschränkung mit VPC-Ursprüngen.

Wenn Sie einen VPC-Ursprung, den Sie besitzen, mit anderen teilen AWS-Konten, ermöglichen Sie diesen, diese Ressource als Ursprung für ihre CloudFront Distributionen zu verwenden.

Um einen VPC-Ursprung gemeinsam zu nutzen, müssen Sie ihn einer Ressourcenfreigabe hinzufügen. Eine Ressourcenfreigabe ist eine AWS RAM -Ressource, mit der Sie Ihre Ressourcen in mehreren AWS-Konten gemeinsam nutzen können.

Eine Ressourcenfreigabe gibt Folgendes an:

- Die Ressourcen, die Sie teilen möchten
- Die Verbraucher, mit denen sie geteilt werden
- Die verwaltete Richtlinie des Dienstes, die die Berechtigungen für die Ressourcen festlegt

Wenn Sie einen VPC-Ursprung mithilfe der CloudFront Konsole gemeinsam nutzen, fügen Sie ihn einer vorhandenen Ressourcenfreigabe hinzu. Wenn Sie noch keine Ressourcenfreigabe

haben, können Sie eine erstellen, wenn Sie einen VPC-Ursprung von der CloudFront Konsole aus gemeinsam nutzen. Sie können auch die AWS RAM Konsole verwenden oder AWS CLI eine separate Konsole erstellen.

Sie können VPC-Ursprünge mit anderen AWS-Konten und AWS Organizations teilen.

- Wenn Sie die Ressource mit einer AWS Organisation teilen, erhalten alle Verbraucher in dieser bestimmten Organisation Zugriff auf den VPC-Ursprung.
- Wenn Sie Inhalte mit einer AWS-Konto oder einer Organisation teilen, der Sie nicht angehören, erhalten die Verbraucher eine Einladung, die gemeinsame Nutzung der Ressource zu akzeptieren. Nach der Annahme können sie den VPC-Ursprung verwenden.

Sie können einen VPC-Ursprung, den Sie besitzen, mit der CloudFront Konsole, der AWS RAM Konsole oder dem teilen. AWS CLI

So erstellen Sie eine Ressourcenfreigabe mithilfe der Konsole CloudFront

1. Öffnen Sie die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home.
2. Wählen Sie im Navigationsbereich VPC-Ursprünge aus.
3. Wählen Sie eine oder mehrere Ressourcen aus und wählen Sie VPC-Ursprung teilen.
4. Wählen Sie Create resource share (Ressourcenfreigabe erstellen) aus.
5. Geben Sie unter Name einen Namen für die Ressourcenfreigabe ein.
6. Wählen Sie für den Typ Principal eine der folgenden Optionen aus:

   - AWS-Konto— Zugriff auf eine bestimmte Person gewähren AWS-Konto.
   - Organisationseinheit — Gewähren Sie Zugriff auf eine bestimmte Organisationseinheit (OU).
   - Organisation — Gewähren Sie Zugriff auf Ihre gesamte Organisation, einschließlich der untergeordneten Organisationseinheiten und AWS-Konten.

   a. Wenn Sie möchten AWS-Konto, geben Sie die Konto-ID-Nummer ein. Sie können Neues Konto hinzufügen wählen, um bis zu 5 Konten hinzuzufügen AWS-Konten.

   b. Wenn Sie Organisationseinheit ausgewählt haben, geben Sie den ARN der OU-Einheit ein. Sie können nur 1 OU eingeben.

   c. Wenn Sie Organisation ausgewählt haben, geben Sie den ARN der Organisation ein. Sie können nur eine Organisation eingeben.

7. Wählen Sie Ressourcen teilen aus.

   CloudFront Wendet standardmäßig die AWSRAMDefaultPermissionCloudFront AWS verwaltete Richtlinie auf die gemeinsame Nutzung von Ressourcen an. Diese Richtlinie ermöglicht schreibgeschützte Aktionen für die gemeinsam genutzte Ressource, sodass nutzende Konten die gemeinsam genutzte Ressource nicht aktualisieren oder löschen können. Sie können diese Richtlinie nicht bearbeiten oder aus der Ressourcenfreigabe entfernen.

   > **Tip**
   >
   > Nachdem Sie die Ressourcenfreigabe erstellt haben, können Sie weitere über die AWS-Konten AWS RAM Konsole hinzufügen. Weitere Informationen finden Sie im AWS RAM Benutzerhandbuch unter Aktualisieren einer Ressourcenfreigabe im AWS RAM.

Um einen VPC-Ursprung, den Sie besitzen, mithilfe der CloudFront Konsole gemeinsam zu nutzen

1. Öffnen Sie die CloudFront Konsole unter. https://console.aws.amazon.com/cloudfront/v4/home
2. Wählen Sie im Navigationsbereich VPC-Ursprünge aus.
3. Wählen Sie eine Ressource aus und wählen Sie VPC-Ursprung teilen.
4. Auf der Seite VPC-Ursprung teilen können Sie eine vorhandene Ressourcenfreigabe auswählen, zu der Sie diesen VPC-Ursprung hinzufügen möchten.
5. Wählen Sie Ressource teilen aus.

   Auf der Ressourcendetailseite unter Geteilt mit können Sie sehen, dass Ihr VPC-Ursprung mit den folgenden Details geteilt wurde:

   - Namen der gemeinsam genutzten Ressourcen
   - Status teilen
   - Datum der letzten Änderung

Nachdem Sie die Ressourcenfreigabe erstellt und für die nutzenden Konten freigegeben haben, haben diese 12 Stunden Zeit, die Einladung anzunehmen. Weitere Informationen finden Sie im AWS RAM Benutzerhandbuch unter Annehmen und Ablehnen von Einladungen zur gemeinsamen Nutzung von Ressourcen.

> **Important**
>
> Damit verbrauchende Konten Ihren VPC-Ursprung für ihre CloudFront Verteilung verwenden können, müssen Sie ihnen auch den Elastic Load Balancing- oder Amazon EC2-Endpunkt des VPC-Ursprungs zur Verfügung stellen.

Um einen VPC-Ursprung, dessen Eigentümer Sie sind, gemeinsam zu nutzen, verwenden Sie den AWS RAM Konsole

Erstellen Sie eine Ressourcenfreigabe und wählen Sie dann die CloudFront Ressourcen aus, die Sie hinzufügen möchten. Weitere Informationen finden Sie im AWS RAM Benutzerhandbuch unter Erstellen einer Ressourcenfreigabe.

Um einen VPC-Ursprung, den Sie besitzen, gemeinsam zu nutzen, verwenden Sie den AWS CLI

Verwenden Sie den Befehl create-resource-share.

Verwenden eines gemeinsam genutzten VPC-Ursprungs

Um einen gemeinsam genutzten VPC-Ursprung zu verwenden, muss das Konto, das die Einladung erhält, die gemeinsame Nutzung der Ressource akzeptieren. Navigieren Sie dazu zur AWS Resource Access Manager Konsole in der Region USA Ost (Nord-Virginia) und akzeptieren Sie alle ausstehenden Anfragen auf der Registerkarte Ausstehend. Weitere Informationen finden Sie im AWS RAM Benutzerhandbuch unter Akzeptieren gemeinsam genutzter Ressourcen.

Nachdem Sie die Ressourcenfreigabe akzeptiert haben, können Sie den VPC-Ursprung als Ursprung für Ihre CloudFront Distributionen verwenden.

Um einen gemeinsam genutzten VPC-Ursprung zu verwenden

1. Öffnen Sie die CloudFront Konsole unter. https://console.aws.amazon.com/cloudfront/v4/home
2. Führen Sie im Navigationsbereich für Distributionen eine der folgenden Aktionen aus:
   - Wählen Sie für eine neue Verteilung die Option Verteilung erstellen aus.
   - Wählen Sie für eine bestehende Verteilung die Distributions-ID aus.
3. Wählen Sie für den Origin-Typ VPC-Ursprung aus und geben Sie dann den VPC-Ursprung an, der mit Ihnen geteilt wurde.

4. Geben Sie für den VPC-Ausgangsendpunkt den privaten DNS-Namen Ihrer Amazon EC2-Instance oder Ihres Elastic Load Balancing-Load Balancers oder die Ursprungsdomain ein. Wenn Sie diesen Wert noch nicht haben, müssen Sie ihn von dem abrufen, dem der AWS-Konto VPC-Ursprung gehört. Wenn Sie diesen Endpunkt noch nicht haben, können Sie ihn von dem abrufen AWS-Konto , dem der VPC-Ursprung gehört.
5. Folgen Sie den restlichen Schritten der Konsole, um Ihre Distribution zu erstellen oder zu aktualisieren.

Identifizierung eines gemeinsamen VPC-Ursprungs

Eigentümer und Verbraucher können gemeinsam genutzte VPC-Ursprünge mithilfe der CloudFront Konsole und identifizieren. AWS CLI

Um einen gemeinsam genutzten VPC-Ursprung mithilfe der Konsole zu identifizieren CloudFront

1. Öffnen Sie die CloudFront Konsole unter. https://console.aws.amazon.com/cloudfront/v4/home
2. Wählen Sie im Navigationsbereich VPC-Ursprünge aus. Sie können die Spalte Besitzer-ID verwenden, um zu identifizieren AWS-Konto , zu welcher die Ressource gehört.
3. Wählen Sie eine Ressource aus.
4. Auf der Ressourcendetailseite unter Geteilt mit können Sie sehen, dass Ihr VPC-Ursprung mit den folgenden Details geteilt wird:

   - Namen der gemeinsam genutzten Ressourcen
   - Status teilen
   - Datum der letzten Änderung

Die gemeinsame Nutzung eines VPC-Ursprungs aufheben

Wenn Sie die gemeinsame Nutzung einer Ressource aufheben, können die AWS-Konten (verbrauchenden Konten) diese Ressource nicht mehr für neue Verteilungen oder die Aktualisierung vorhandener Verteilungen verwenden.

> **Note**
>
> Wenn Sie die gemeinsame Nutzung einer Ressource aufheben, bleiben bestehende Distributionen, die diese Ressource noch verwenden, aktiv und bedienen weiterhin

den Datenverkehr. Diese Verteilungen können jedoch erst bearbeitet werden, wenn die nicht gemeinsam genutzte Ressource als Quelle entfernt wird. Wir empfehlen, dass du sicherstellst, dass alle nutzenden Konten die nicht gemeinsam genutzte Ressource nicht mehr verwenden, bevor du die gemeinsame Nutzung aufnimmst.

Um die gemeinsame Nutzung eines VPC-Ursprungs, dessen Eigentümer Sie sind, aufzuheben, müssen Sie ihn aus der Ressourcenfreigabe entfernen. Sie können dies mit der CloudFront Konsole, AWS RAM der Konsole oder dem tun. AWS CLI

Um die Freigabe eines gemeinsam genutzten VPC-Ursprungs, dessen Eigentümer Sie sind, mithilfe der Konsole aufzuheben CloudFront

1. Öffnen Sie die CloudFront Konsole unter. https://console.aws.amazon.com/cloudfront/v4/home
2. Wählen Sie im Navigationsbereich VPC-Ursprünge aus.
3. Wählen Sie eine Ressource aus und klicken Sie auf Unshare.
4. Prüfen Sie die Details im Dialogfeld „Ressource nicht teilen“ und wählen Sie dann „Teilen aufheben“. Die aufgelisteten Principals haben keinen Zugriff mehr auf Ihre gemeinsam genutzte Ressource.

Um die Freigabe eines gemeinsam genutzten VPC-Ursprungs, dessen Eigentümer Sie sind, aufzuheben, verwenden Sie den AWS RAM Konsole

Siehe Aktualisieren einer Ressourcenfreigabe im AWS RAM -Benutzerhandbuch.

Um die Freigabe eines gemeinsam genutzten VPC-Ursprungs, dessen Eigentümer Sie sind, aufzuheben, verwenden Sie den AWS CLI

Verwenden Sie den Befehl disassociate-resource-share.

Zuständigkeiten und Berechtigungen für gemeinsam genutzte VPC-Ursprünge

Berechtigungen für Besitzer

Stellen Sie als Konto, das die Ressource besitzt, sicher, dass alle nutzenden Konten die Ressource nicht mehr verwenden, bevor Sie die Freigabe aufheben oder sie löschen.

Berechtigungen für Konsumenten

Nutzerkonten können gemeinsam genutzte Ressourcen als Quelle für ihre CloudFront Verteilungen verwenden, aber sie können die Ressourcen nicht bearbeiten oder löschen. Standardmäßig wird die AWSRAMDefaultPermissionCloudFront AWS verwaltete Richtlinie auf den Ressourcenanteil im Sharing-Konto (dem Konto, dem die Ressource gehört) angewendet.

AWSRAMDefaultPermissionCloudFront

Wenn Sie eine Ressourcenfreigabe in erstellen CloudFront, CloudFront verwendet die AWSRAMDefaultPermissionCloudFront AWS verwaltete Richtlinie und wendet sie auf Ihre Ressourcenfreigabe an. Diese Richtlinie gewährt nur Leseberechtigungen für CloudFront Ressourcen, die vom Ressourcenbesitzer für das nutzende Konto gemeinsam genutzt werden können.

Weitere Informationen zur Verwaltung von Berechtigungen finden Sie unter Verwalten von Berechtigungen in AWS RAM im AWS Resource Access Manager Benutzerhandbuch. AWS RAM

Fakturierung und Messung

Für die gemeinsame Nutzung von VPC-Ursprüngen mit anderen AWS-Konten fallen keine zusätzlichen Gebühren an. Die Nutzungskosten für den Traffic einer Distribution, die einen gemeinsam genutzten VPC-Ursprung verwendet, gehen zu Lasten des Nutzerkontos, dem die Distribution gehört.

Kontingente für gemeinsam genutzte Ressourcen

CloudFront verwendet dieselben Kontingente für die gemeinsame Nutzung von Ressourcen wie unter angegeben AWS RAM. Von der CloudFront Konsole aus können Sie bis zu 5 AWS-Konten, 1 OU oder 1 Organisation hinzufügen. Verwenden Sie die AWS RAM Konsole oder die AWS RAM API, um weitere hinzuzufügen.

Weitere Informationen finden Sie unter Service Quotas für AWS RAM im AWS RAM User Guide.

Caching und Verfügbarkeit

Sie können CloudFront es verwenden, um die Anzahl der Anfragen zu reduzieren, auf die Ihr Ursprungsserver direkt antworten muss. Beim CloudFront Caching werden mehr Objekte von CloudFront Edge-Standorten aus bereitgestellt, die näher an Ihren Benutzern liegen. Dies reduziert die Belastung Ihres Ursprungs-Servers sowie die Latenz.

Je mehr Anfragen von Edge-Caches verarbeitet werden CloudFront können, desto weniger Viewer-Anfragen CloudFront müssen an Ihren Ursprung weitergeleitet werden, um die neueste Version oder eine eindeutige Version eines Objekts abzurufen. Erwägen Sie CloudFront die Verwendung eines Origin Shields, um die Optimierung so wenig Anfragen wie möglich an Ihren CloudFront Ursprung zu richten. Weitere Informationen finden Sie unter Verwenden Sie Amazon CloudFront Origin Shield.

Der Anteil der Anfragen, die direkt aus dem CloudFront Cache bedient werden, im Vergleich zu allen Anfragen wird als Cache-Trefferquote bezeichnet. Sie können den Prozentsatz der Zuschaueranfragen, bei denen es sich um Treffer, Fehlschläge und Fehler handelt, in der CloudFront Konsole einsehen. Weitere Informationen finden Sie unter Berichte zu CloudFront Cache-Statistiken anzeigen.

Die Cache-Trefferrate wird von einer Reihe von Faktoren beeinflusst. Sie können Ihre CloudFront Verteilungskonfiguration anpassen, um die Cache-Trefferquote zu verbessern, indem Sie den Anweisungen unter folgenErhöhen Sie den Anteil der Anfragen, die direkt aus den CloudFront Caches bedient werden (Cache-Trefferquote).

Weitere Informationen zum Hinzufügen und Entfernen der Inhalte, die Sie bereitstellen CloudFront möchten, finden Sie unterInhalte hinzufügen, entfernen oder ersetzen, die CloudFront verbreiten.

Themen

- Erhöhen Sie den Anteil der Anfragen, die direkt aus den CloudFront Caches bedient werden (Cache-Trefferquote)
- Verwenden Sie Amazon CloudFront Origin Shield
- Optimieren Sie die Hochverfügbarkeit mit CloudFront Origin-Failover
- Verwalten der Dauer, die Inhalte im Cache bleiben (Ablauf)
- Zwischenspeichern von Inhalten auf der Grundlage von Abfragezeichenfolgeparametern
- Zwischenspeichern von Inhalten auf der Grundlage von Cookies
- Zwischenspeichern von Inhalten auf der Grundlage von Anforderungsheadern

Erhöhen Sie den Anteil der Anfragen, die direkt aus den CloudFront Caches bedient werden (Cache-Trefferquote)

Sie können die Leistung verbessern, indem Sie den Anteil Ihrer Zuschaueranfragen erhöhen, die direkt aus dem CloudFront Cache bedient werden, anstatt die Inhalte auf Ihren Ursprungsservern abzurufen. Dies ist als Verbesserung der Cache-Trefferquote bekannt.

In den folgenden Abschnitten wird erläutert, wie Sie die Cache-Trefferrate erhöhen.

Themen

- Geben Sie an, wie lange Ihre CloudFront Objekte zwischengespeichert werden
- Verwenden von Origin Shield
- Zwischenspeichern auf der Grundlage von Abfragezeichenfolgeparametern
- Zwischenspeichern auf der Grundlage von Cookie-Werten
- Zwischenspeichern auf der Grundlage von Anfrage-Headern
- Entfernen des Accept-Encoding-Headers, wenn keine Kompression erforderlich ist
- Bereitstellen von Medieninhalten über HTTP

Geben Sie an, wie lange Ihre CloudFront Objekte zwischengespeichert werden

Um Ihre Cache-Trefferquote zu erhöhen, können Sie Ihren Ursprung so konfigurieren, dass Ihren Objekten eine Cache-Control maximale Altersanweisung hinzugefügt wird, und den längsten praktischen Wert für angeben. `max-age` Je kürzer die Cachedauer ist, desto häufiger werden Anfragen CloudFront an Ihren Ursprung gesendet, um festzustellen, ob sich ein Objekt geändert hat, und um die neueste Version abzurufen. Sie können die `max-age` mit den `stale-while-revalidate`- und `stale-if-error`-Anweisungen hinzufügen, um die Cache-Trefferquote unter bestimmten Bedingungen weiter zu verbessern. Weitere Informationen finden Sie unter Verwalten der Dauer, die Inhalte im Cache bleiben (Ablauf).

Verwenden von Origin Shield

CloudFront Origin Shield kann dir helfen, die Cache-Trefferquote deiner CloudFront Distribution zu verbessern, da es eine zusätzliche Cache-Ebene vor deinem Ursprung bietet. Wenn du Origin Shield verwendest, kommen alle Anfragen von allen CloudFront Caching-Ebenen an deinen Ursprung

von einem einzigen Standort. CloudFront kann jedes Objekt mithilfe einer einzigen Quellanfrage von Origin Shield abrufen, und alle anderen Ebenen des CloudFront Caches (Edge-Standorte und regionale Edge-Caches) können das Objekt aus Origin Shield abrufen.

Weitere Informationen finden Sie unter Verwenden Sie Amazon CloudFront Origin Shield.

Zwischenspeichern auf der Grundlage von Abfragezeichenfolgeparametern

Wenn Sie das Zwischenspeichern CloudFront auf der Grundlage von Abfragezeichenfolgenparametern konfigurieren, können Sie das Caching verbessern, indem Sie wie folgt vorgehen:

- Konfigurieren CloudFront Sie so, dass nur die Abfragezeichenfolgenparameter weitergeleitet werden, für die Ihr Ursprung eindeutige Objekte zurückgibt.
- Verwenden Sie die gleiche Schreibweise (Groß- oder Kleinschreibung) für alle Instances desselben Parameters. Wenn beispielsweise eine Anfrage enthält `parameter1=A` und eine andere enthält`parameter1=a`, CloudFront leitet sie separate Anfragen an Ihren Ursprung weiter, wenn eine Anfrage Folgendes enthält `parameter1=A` und wenn eine Anfrage Folgendes enthält`parameter1=a`. CloudFront speichert dann separat die entsprechenden Objekte, die von Ihrem Ursprung zurückgegeben werden, separat im Cache, auch wenn die Objekte identisch sind. Wenn Sie nur `A` oder `a` verwenden, leitet CloudFront weniger Anfragen an Ihren Ursprung weiter.
- Listen Sie Parameter in der gleichen Reihenfolge auf. Wie bei den Unterschieden in der Schreibweise gilt auch hier, dass wenn eine Anfrage für ein Objekt die Abfragezeichenfolge `parameter1=a&parameter2=b` und eine weitere Anfrage für dasselbe Objekt `parameter2=b&parameter1=a` enthält, leitet CloudFront beide Anfragen an Ihren Ursprung weiter und speichert die entsprechenden Objekte separat zwischen, auch wenn sie identisch sind. Wenn Sie für Parameter immer dieselbe Reihenfolge verwenden, werden weniger Anfragen an Ihren Ursprung CloudFront weitergeleitet.

Weitere Informationen finden Sie unter Zwischenspeichern von Inhalten auf der Grundlage von Abfragezeichenfolgeparametern. Wenn Sie die Abfragezeichenfolgen überprüfen möchten, die CloudFront an Ihren Ursprung weitergeleitet werden, sehen Sie sich die Werte in der `cs-uri-query` Spalte Ihrer CloudFront Protokolldateien an. Weitere Informationen finden Sie unter Zugriffsprotokolle (Standardprotokolle).

Zwischenspeichern auf der Grundlage von Cookie-Werten

Wenn Sie CloudFront das Caching auf der Grundlage von Cookie-Werten konfigurieren, können Sie das Caching verbessern, indem Sie wie folgt vorgehen:

- Konfigurieren CloudFront Sie so, dass nur bestimmte Cookies weitergeleitet werden, anstatt alle Cookies weiterzuleiten. Leitet bei den Cookies, die Sie so konfigurieren CloudFront , dass sie an Ihren Ursprung CloudFront weitergeleitet werden, jede Kombination aus Cookie-Name und -Wert weiter. Die Objekte, die Ihr Ursprung zurückgibt, werden dann separat gespeichert, auch wenn sie alle identisch sind.

  Nehmen wir zum Beispiel an, dass Viewer zwei Cookies in jede Anfrage aufnehmen, dass jedes Cookie drei mögliche Werte hat und dass alle Kombinationen von Cookie-Werten möglich sind. CloudFront leitet für jedes Objekt bis zu neun verschiedene Anfragen an Ihren Ursprung weiter. Wenn Ihr Ursprung verschiedene Versionen eines Objekts zurückgibt, die nur auf einem der Cookies basieren, leitet er mehr Anfragen an Ihren Ursprung weiter als nötig und speichert unnötig mehrere identische Versionen des Objekts im Cache. CloudFront
- Erstellen Sie separate Cache-Verhaltensweisen für statische und dynamische Inhalte und konfigurieren Sie CloudFront sie so, dass Cookies nur für dynamische Inhalte an Ihren Ursprung weitergeleitet werden.

  Nehmen wir zum Beispiel an, Sie haben nur ein Cache-Verhalten für Ihre Distribution und verwenden die Distribution sowohl für dynamische Inhalte wie `.js` Dateien als auch für `.css` Dateien, die sich selten ändern. CloudFront speichert separate Versionen Ihrer `.css` Dateien auf der Grundlage von Cookie-Werten im Cache, sodass jeder CloudFront Edge-Standort für jeden neuen Cookiewert oder jede Kombination von Cookie-Werten eine Anfrage an Ihren Ursprung weiterleitet.

  Angenommen, Sie erstellen ein Cache-Verhalten, dessen Pfadmuster lautet`*.css`, und das Verhalten wird nicht auf der Grundlage von Cookie-Werten zwischengespeichert. Leitet in diesem Fall CloudFront `.css` Dateianfragen nur für die erste Anfrage, die ein Edge-Standort für eine bestimmte `.css` Datei erhält, und für die erste Anforderung nach Ablauf einer `.css` Datei an Ihren Ursprung weiter.
- Erstellen Sie nach Möglichkeit separate Cache-Verhaltensweisen für dynamische Inhalte, wenn Cookie-Werte für jeden Benutzer eindeutig sind (z. B. eine Benutzer-ID), und dynamische Inhalte, die je nach einer geringeren Anzahl eindeutiger Werte variieren.

Weitere Informationen finden Sie unter Zwischenspeichern von Inhalten auf der Grundlage von Cookies. Wenn Sie die Cookies überprüfen möchten, die an Ihren CloudFront Ursprung weitergeleitet werden, sehen Sie sich die Werte in der `cs(Cookie)` Spalte Ihrer CloudFront Protokolldateien an. Weitere Informationen finden Sie unter Zugriffsprotokolle (Standardprotokolle).

Zwischenspeichern auf der Grundlage von Anfrage-Headern

Wenn Sie das Caching auf der Grundlage von Anforderungsheadern konfigurieren CloudFront , können Sie das Caching verbessern, indem Sie wie folgt vorgehen:

- Konfigurieren Sie so CloudFront , dass Weiterleitung und Zwischenspeicherung nur auf der Grundlage bestimmter Header statt Weiterleitung und Zwischenspeicherung auf der Grundlage aller Header erfolgen. Leitet für die von Ihnen angegebenen Header jede Kombination aus CloudFront Header-Name und Wert weiter. Die Objekte, die Ihr Ursprung zurückgibt, werden dann separat gespeichert, auch wenn sie alle identisch sind.

  > **Note**
  >
  > CloudFront leitet die in den folgenden Themen angegebenen Header immer an Ihren Ursprung weiter:
  >
  > - Wie CloudFront verarbeitet und leitet Anfragen an Ihren Amazon S3-Ursprungsserver weiter > HTTP-Anforderungsheader, die CloudFront entfernt oder aktualisiert werden
  > - Wie CloudFront verarbeitet und leitet Anfragen an Ihren benutzerdefinierten Ursprungsserver weiter > Header und CloudFront Verhalten von HTTP-Anfragen (benutzerdefiniert und Amazon S3-Ursprünge)

  Wenn Sie CloudFront den Cache auf der Grundlage von Anforderungsheadern konfigurieren, ändern Sie nicht die Header, die CloudFront weiterleiten, sondern nur, ob Objekte auf der Grundlage der CloudFront Header-Werte zwischengespeichert werden.
- Vermeiden Sie die Zwischenspeicherung auf der Grundlage von Anfrage-Headern mit einer hohen Anzahl von eindeutigen Werten.

  Wenn Sie beispielsweise Bilder in unterschiedlichen Größen bereitstellen möchten, die auf dem Gerät des Benutzers basieren, konfigurieren Sie CloudFront das Zwischenspeichern nicht auf der Grundlage des `User-Agent` Headers, der eine enorme Anzahl möglicher Werte enthält. Konfigurieren Sie stattdessen CloudFront den Cache auf der Grundlage der CloudFront

Geräte-Typ-Header`CloudFront-Is-Desktop-Viewer`,`CloudFront-Is-Mobile-Viewer`, `CloudFront-Is-SmartTV-Viewer` und. `CloudFront-Is-Tablet-Viewer` Wenn Sie zudem dieselbe Version des Images für Tablets und Desktops zurückgeben, leiten Sie nur den `CloudFront-Is-Tablet-Viewer`-Header weiter und nicht den `CloudFront-Is-Desktop-Viewer`-Header.

Weitere Informationen finden Sie unter Zwischenspeichern von Inhalten auf der Grundlage von Anforderungsheadern.

Entfernen des **`Accept-Encoding`**-Headers, wenn keine Kompression erforderlich ist

Wenn die Komprimierung nicht aktiviert ist, weil der Ursprung sie nicht unterstützt, nicht unterstützt oder der Inhalt CloudFront nicht komprimierbar ist, können Sie die Cache-Trefferquote erhöhen, indem Sie ein Cache-Verhalten in Ihrer Distribution einem Ursprung zuordnen, das Folgendes festlegt: Custom Origin Header

- Header name (Header-Name: `Accept-Encoding`
- Header value (Header-Wert): (Frei lassen)

Wenn Sie diese Konfiguration verwenden, wird der Header aus dem Cache-Schlüssel CloudFront entfernt und der `Accept-Encoding` Header wird nicht in Quellanforderungen aufgenommen. Diese Konfiguration gilt für alle Inhalte, die mit CloudFront der Distribution von diesem Ursprung aus bereitgestellt werden.

Bereitstellen von Medieninhalten über HTTP

Weitere Informationen zum Optimieren von Video-on-Demand-(VOD)- und Streaming-Videoinhalten finden Sie unter Video auf Abruf und Live-Streaming-Video mit CloudFront.

Verwenden Sie Amazon CloudFront Origin Shield

CloudFront Origin Shield ist eine zusätzliche Ebene in der CloudFront Caching-Infrastruktur, die dazu beiträgt, die Auslastung Ihres Origins zu minimieren, seine Verfügbarkeit zu verbessern und die Betriebskosten zu senken. CloudFront Origin Shield bietet die folgenden Vorteile:

Bessere Cache-Zugriffsrate

Origin Shield kann dazu beitragen, die Cache-Trefferquote deiner CloudFront Distribution zu verbessern, da es eine zusätzliche Caching-Ebene vor deinem Origin bietet. Wenn du Origin Shield verwendest, werden alle Anfragen von allen CloudFront Caching-Ebenen an deinen Ursprung über Origin Shield weitergeleitet, was die Wahrscheinlichkeit eines Cache-Treffers erhöht. Anfragen, die in derselben Region wie dein Absender gestellt werden, umgehen Origin Shield. CloudFrontkann jedes Objekt mit einer einzigen Ursprungsanfrage von Origin Shield an deinen Ursprung abrufen, und alle anderen Ebenen des CloudFront Caches (Edge-Standorte und regionale Edge-Caches) können das Objekt von Origin Shield abrufen.

Reduzierte Ursprungslast

Origin Shield kann die Anzahl der gleichzeitigen Anfragen die für dasselbe Objekt an Ihren Ursprung gesendet werden, weiter reduzieren. Anfragen für Inhalte, die sich nicht im Origin-Shield-Cache befinden, werden mit anderen Anforderungen für dasselbe Objekt konsolidiert, was dazu führt, dass nur eine Anfrage an Ihren Ursprung gelangt. Die Verarbeitung von weniger Anfragen an Ihrem Ursprung kann die Verfügbarkeit Ihres Ursprungs bei Spitzenlasten oder unerwarteten Datenverkehrsspitzen bewahren und Kosten z. B. bei Just-in-Time-Packaging, Image-Transformationen und ausgehenden Datentransfer (DTO) senken.

Bessere Netzwerkleistung

Wenn du Origin Shield in der AWS Region aktivierst, die die niedrigste Latenz zu deinem Ursprung hat, kannst du eine bessere Netzwerkleistung erzielen. Bei Ursprüngen in einer AWS Region verbleibt der CloudFront Netzwerkverkehr bis zu deinem Ursprung im CloudFront Netzwerk mit hohem Durchsatz. Bei Ursprüngen außerhalb von AWS verbleibt der CloudFront Netzwerkverkehr im CloudFront Netzwerk bis hin zu Origin Shield, das über eine Verbindung mit niedriger Latenz zu deinem Ursprung verfügt.

Für die Nutzung von Origin Shield fallen zusätzliche Gebühren an. Weitere Informationen finden Sie unter CloudFront Preise.

> **Note**
>
> Origin Shield wird für gRPC-Anforderungen nicht unterstützt. Wenn in einer Distribution, die gRPC unterstützt, Origin Shield aktiviert ist, funktionieren gRPC-Anforderungen weiterhin. Die Anforderungen werden jedoch direkt an den gRPC-Ursprung weitergeleitet, ohne

Origin Shield zu durchlaufen. Weitere Informationen finden Sie unter gRPC mit CloudFront Distributionen verwenden.

Themen

- Anwendungsfälle für Origin Shield
- Wählen Sie das Symbol AWS Region für Origin Shield
- Origin Shield aktivieren
- Schätzung der Kosten für Origin Shield
- Hochverfügbarkeit bei Origin Shield
- Wie Origin Shield mit anderen CloudFront Funktionen interagiert

Anwendungsfälle für Origin Shield

CloudFront Origin Shield kann für viele Anwendungsfälle von Vorteil sein, darunter die folgenden:

- Viewer, die über verschiedene geografische Regionen verteilt sind
- Ursprünge, die Just-in-Time-Paketerstellung für Live-Streaming oder schnelle Image-Verarbeitung bieten
- On-premises Ursprünge mit Kapazitäts- oder Bandbreitenbeschränkungen
- Workloads, die mehrere CDNs (Content Delivery Networks) verwenden

Origin Shield eignet sich möglicherweise in anderen Fällen nicht gut, z. B. dynamische Inhalte, die an den Ursprung weitergeleitet werden, Inhalte mit geringer Cachefähigkeit oder Inhalte, die selten angefordert werden.

In den folgenden Abschnitten werden die Vorteile von Origin Shield für die folgenden Anwendungsfälle erläutert.

Anwendungsfälle

- Viewer in verschiedenen geografischen Regionen
- Mehrere CDNs

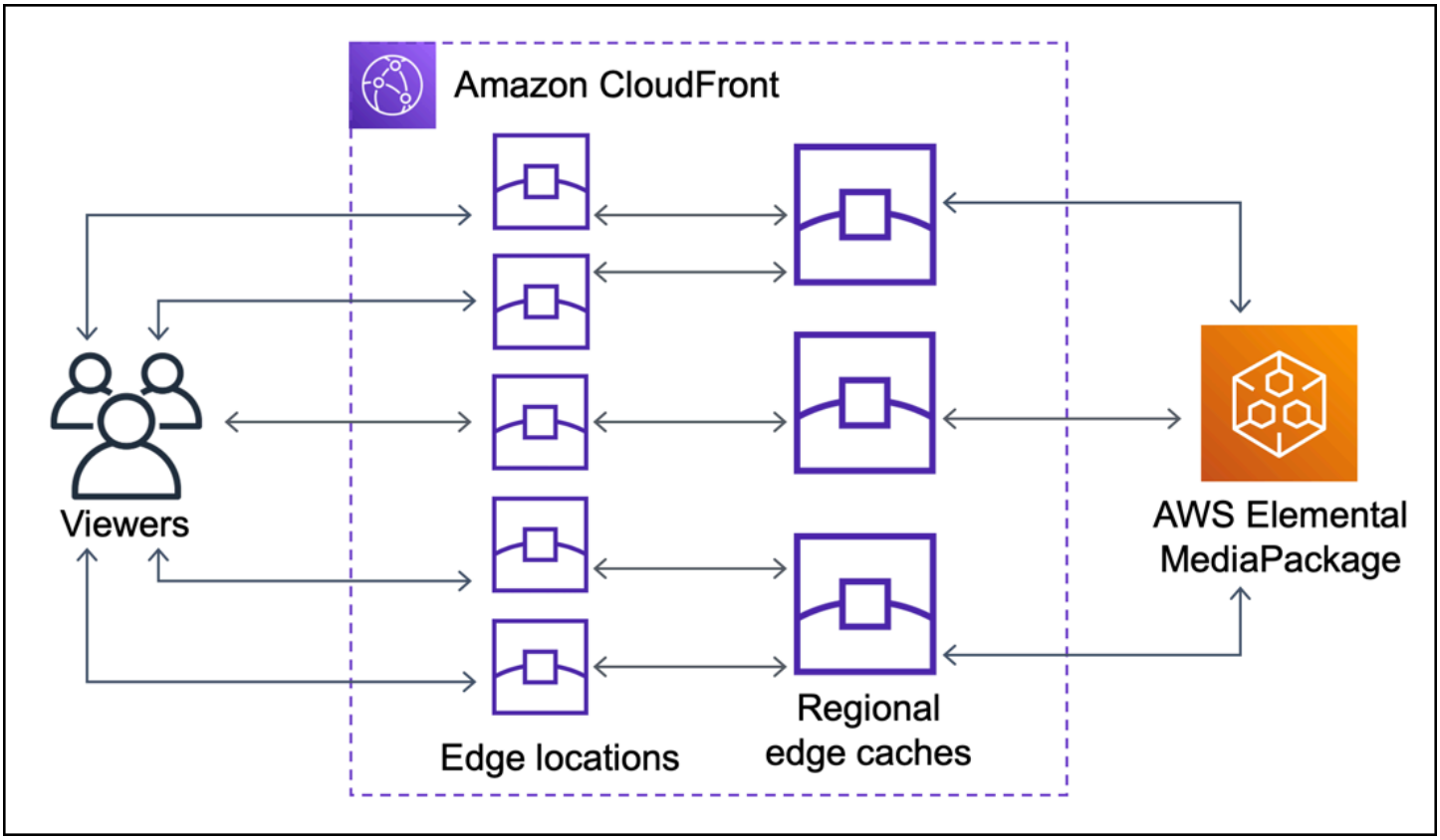
Viewer in verschiedenen geografischen Regionen

Mit Amazon wird die Belastung Ihres Ursprungs von Natur aus reduziert CloudFront, da Anfragen, die über den Cache bedient werden CloudFront können, nicht an Ihren Ursprung weitergeleitet werden. Zusätzlich zum CloudFront globalen Netzwerk von Edge-Standorten dienen regionale Edge-Caches als mittlere Caching-Ebene, um Cache-Treffer bereitzustellen und Herkunftsanfragen für Zuschauer in nahegelegenen geografischen Regionen zu konsolidieren. Betrachteranfragen werden zuerst an einen nahe gelegenen CloudFront -Edge-Standort weitergeleitet. Wenn das Objekt an diesem Speicherort nicht zwischengespeichert wird, wird die Anforderung an einen regionalen Edge-Cache gesendet.

Wenn sich Viewer in verschiedenen geografischen Regionen befinden, können Anfragen über verschiedene regionale Edge-Caches weitergeleitet werden, von denen jeder eine Anfrage für denselben Inhalt an Ihren Ursprung senden kann. Aber mit Origin Shield erhalten Sie eine zusätzliche Caching-Ebene zwischen den regionalen Edge-Caches und Ihrem Ursprung. Alle Anfragen von allen regionalen Edge-Caches laufen durch Origin Shield, wodurch die Belastung Ihres Ursprungs weiter reduziert wird. Das folgende Diagramm verdeutlicht dies. In den folgenden Diagrammen ist der Ursprung AWS Elemental MediaPackage.

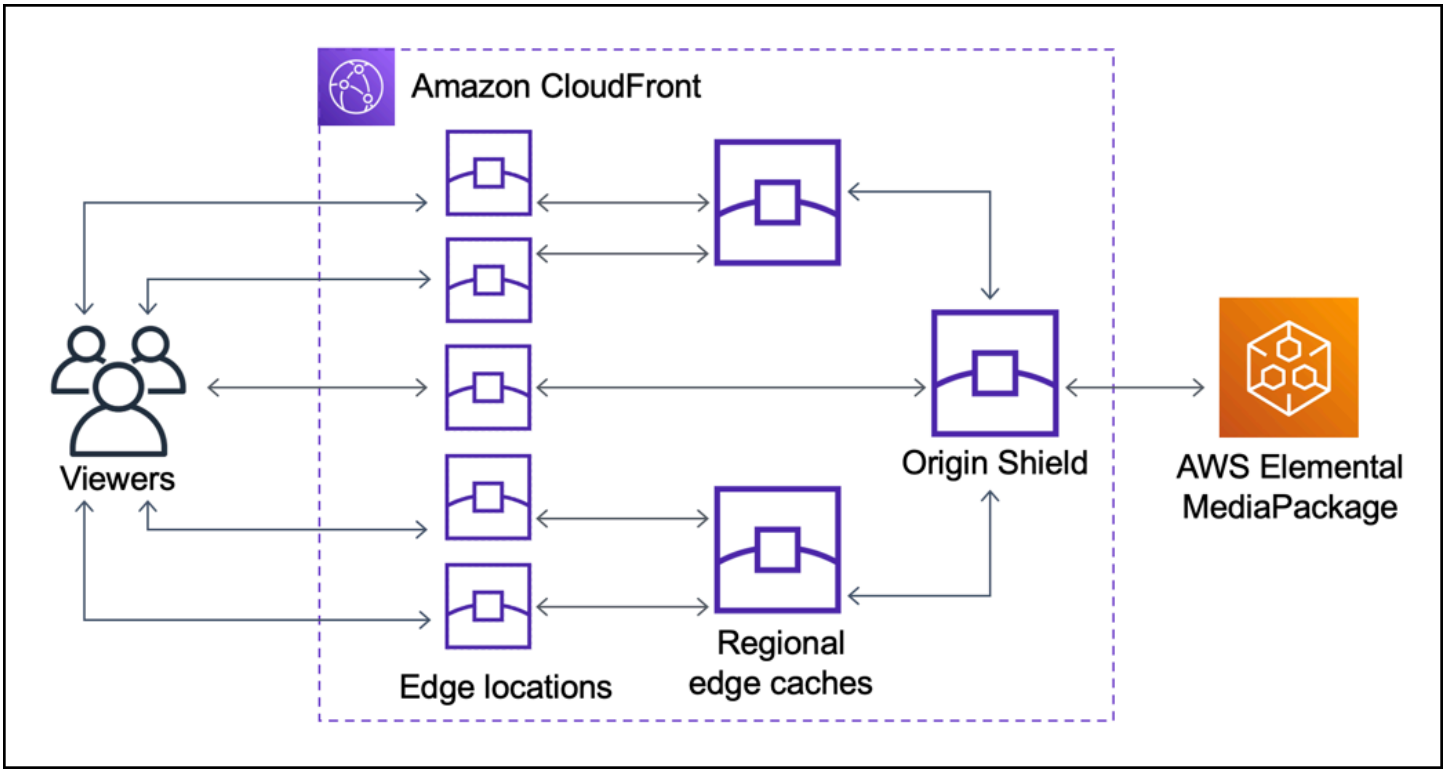
Ohne Origin Shield

Ohne Origin Shield erhält Ihr Ursprung möglicherweise doppelte Anfragen für denselben Inhalt, wie im folgenden Diagramm dargestellt.



Mit Origin Shield

Die Verwendung von Origin Shield kann dazu beitragen, die Belastung Ihres Ursprungs zu reduzieren, wie in der folgenden Abbildung gezeigt.



Mehrere CDNs

Um Live-Videoereignisse oder beliebte On-Demand-Inhalte bereitzustellen, können Sie mehrere CDNs (Content Delivery Networks) verwenden. Die Verwendung mehrerer CDNs kann bestimmte Vorteile bieten, aber es bedeutet auch, dass Ihr Ursprung viele doppelte Anfragen für denselben Inhalt erhält, die jeweils von verschiedenen CDNs oder verschiedenen Orten innerhalb desselben CDN stammen. Diese redundanten Anfragen können sich negativ auf die Verfügbarkeit Ihres Ursprungs auswirken oder zusätzliche Betriebskosten für Prozesse wie Just-in-Time-Packaging oder ausgehendem Datentransfer (DTO) ins Internet verursachen.

Wenn du Origin Shield mit der Nutzung deiner CloudFront Distribution als Quelle für andere CDNs kombinierst, kannst du die folgenden Vorteile nutzen:

- Weniger redundante Anfragen, die an Ihrem Ursprung empfangen werden, was dazu beiträgt, die negativen Auswirkungen der Verwendung mehrerer CDNs zu reduzieren.
- Ein gemeinsamer Cache-Schlüssel für CDNs und zentralisierte Verwaltung für Funktionen mit Ursprung.
- Verbesserte Netzwerkleistung. Der Netzwerkverkehr von anderen CDNs wird an einem nahegelegenen CloudFront Edge-Standort beendet, was zu einem Treffer aus dem lokalen Cache führen kann. Wenn sich das angeforderte Objekt nicht im Edge-Location-Cache befindet,

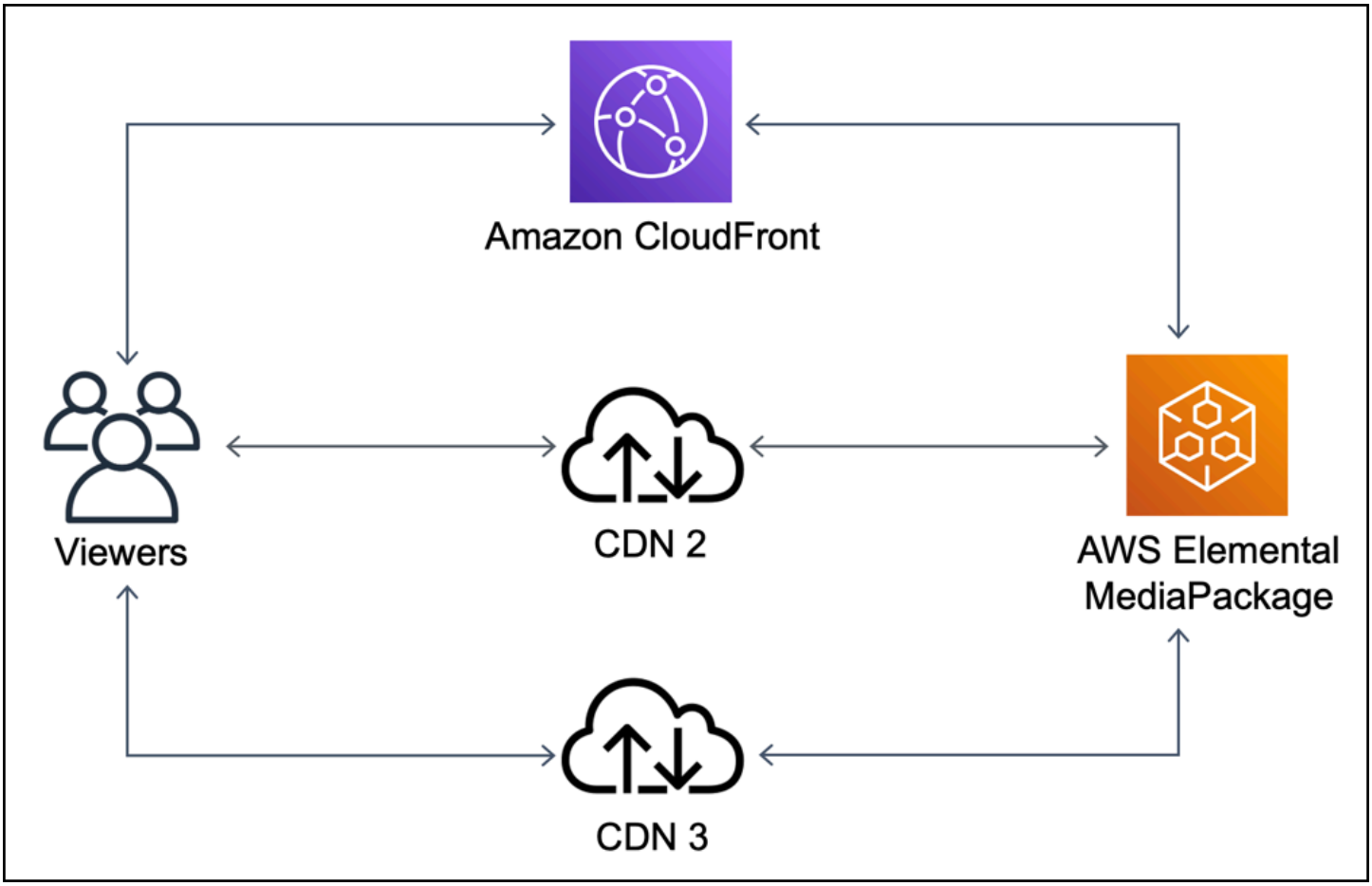
verbleibt die Anfrage an den Ursprung bis Origin Shield im CloudFront Netzwerk, was einen hohen Durchsatz und eine geringe Latenz für den Ursprung bietet. Wenn sich das angeforderte Objekt im Origin-Shield-Cache befindet, wird die Anfrage an Ihren Ursprung vollständig vermieden.

> **Important**
>
> Wenn Sie Origin Shield in einer Multi-CDN-Architektur verwenden möchten und vergünstigte Preise erhalten möchten, kontaktieren Sie uns oder Ihren AWS Vertriebsmitarbeiter für weitere Informationen. Zusätzliche Gebühren können anfallen.

Die folgenden Diagramme zeigen, wie diese Konfiguration dazu beitragen kann, die Belastung Ihres Ursprungs zu minimieren, wenn Sie beliebte Live-Videoereignisse mit mehreren CDNs anbieten. In den folgenden Diagrammen ist der Ursprung AWS Elemental MediaPackage.

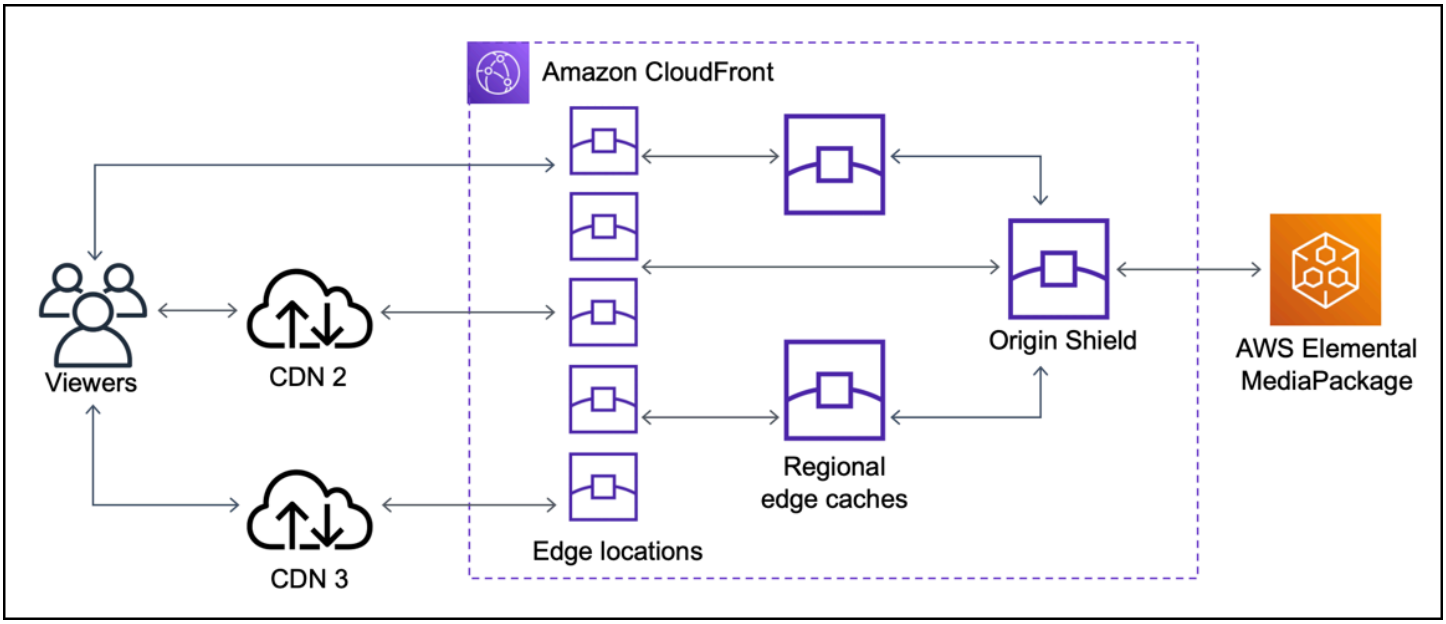
Ohne Origin Shield (mehrere CDNs)

Ohne Origin Shield kann Ihr Ursprung viele doppelte Anfragen für denselben Inhalt erhalten, die jeweils von einem anderen CDN stammen, wie im folgenden Diagramm dargestellt.



Mit Origin Shield (mehrere CDNs)

Die Verwendung von Origin Shield CloudFront als Ursprung für deine anderen CDNs kann dazu beitragen, die Belastung deines Originals zu reduzieren, wie in der folgenden Abbildung dargestellt.



Wählen Sie das Symbol AWS Region für Origin Shield

Amazon CloudFront bietet Origin Shield in AWS Regionen an, in denen CloudFront es einen regionalen Edge-Cache gibt. Wenn du Origin Shield aktivierst, wählst du die AWS Region für Origin Shield aus. Sie sollten die AWS -Region auswählen, die die niedrigste Latenz zu Ihrem Ursprung aufweist. Du kannst Origin Shield mit Ursprüngen verwenden, die in einer AWS Region liegen, und mit Ursprüngen, die sich nicht in einer Region befinden AWS.

Für Ursprünge in einem AWS Region

Wenn dein Ursprung in einer AWS Region liegt, stelle zunächst fest, ob dein Ursprung in einer Region liegt, in der Origin Shield CloudFront angeboten wird. CloudFront bietet Origin Shield in den folgenden AWS Regionen an.

- US East (Ohio) – `us-east-2`
- USA Ost (Nord-Virginia) – `us-east-1`
- USA West (Oregon) – `us-west-2`
- Asia Pacific (Mumbai) – `ap-south-1`
- Asien-Pazifik (Seoul) – `ap-northeast-2`
- Asia Pacific (Singapore) – `ap-southeast-1`
- Asien-Pazifik (Sydney) – `ap-southeast-2`
- Asien-Pazifik (Tokio) – `ap-northeast-1`

- Europa (Frankfurt) – `eu-central-1`
- Europa (Ireland) – `eu-west-1`
- Europa (London) – `eu-west-2`
- Europa (Paris) – `eu-west-3`
- Südamerika (São Paulo) – `sa-east-1`
- Naher Osten (VAE) — `me-central-1`

Wenn dein Ursprung in einer AWS Region liegt, in der Origin Shield CloudFront angeboten wird

Wenn dein Ursprung in einer AWS Region liegt, in der Origin Shield CloudFront angeboten wird (siehe vorherige Liste), aktiviere Origin Shield in derselben Region wie dein Heimatland.

Wenn dein Ursprung nicht in einer AWS Region liegt, in der Origin Shield CloudFront angeboten wird

Wenn dein Ursprung nicht in einer AWS Region liegt, in der Origin Shield CloudFront angeboten wird, kannst du anhand der folgenden Tabelle herausfinden, in welcher Region Origin Shield aktiviert werden soll.

Bei einem Ursprung in...	Origin Shield aktivieren in...
US West (N. California) – `us-west-1`	US West (Oregon) – `us-west-2`
Africa (Cape Town) – `af-south-1`	Europe (Ireland) – `eu-west-1`
Asia Pacific (Hong Kong) – `ap-east-1`	Asia Pacific (Singapore) – `ap-southeast-1`
Canada (Central) – `ca-central-1`	US East (N. Virginia) – `us-east-1`
Europe (Milan) – `eu-south-1`	Europe (Frankfurt) – `eu-central-1`
Europe (Stockholm) – `eu-north-1`	Europe (London) – `eu-west-2`
Middle East (Bahrain) – `me-south-1`	Asia Pacific (Mumbai) – `ap-south-1`

Für Ursprünge außerhalb von AWS

Sie können Origin Shield mit einem Ursprung verwenden, der lokal ist oder sich nicht in einer AWS -Region befindet. Aktiviere in diesem Fall Origin Shield in der AWS Region, die die niedrigste Latenz

zu deinem Ursprung hat. Wenn du dir nicht sicher bist, welche AWS Region die niedrigste Latenz zu deinem Ursprung hat, kannst du die folgenden Vorschläge verwenden, um dir bei der Entscheidung zu helfen.

- In der obigen Tabelle finden Sie eine Näherung darüber, welche AWS -Region basierend auf der geografischen Lage Ihres Ursprungs die niedrigste Latenz zu Ihrem Ursprung hat.
- Sie können Amazon EC2 EC2-Instances in verschiedenen AWS Regionen starten, die sich geografisch in der Nähe Ihres Ursprungs befinden, und einige Tests durchführen, `ping` um die typischen Netzwerklatenzen zwischen diesen Regionen und Ihrem Ursprung zu messen.

Origin Shield aktivieren

Sie können Origin Shield aktivieren, um Ihre Cache-Trefferquote zu verbessern, die Belastung Ihres Ursprungs zu reduzieren und die Leistung zu verbessern. Um Origin Shield zu aktivieren, ändere die Origin-Einstellungen in einer CloudFront Distribution. Origin Shield ist eine Eigenschaft des Ursprungs. Für jeden Ursprung in deinen CloudFront Distributionen kannst du Origin Shield separat in der AWS Region aktivieren, die für diesen Ursprung die beste Leistung bietet.

Du kannst Origin Shield in der CloudFront Konsole CloudFormation, mit oder mit der CloudFront API aktivieren.

Console

So aktivieren Sie Origin Shield für einen vorhandenen Ursprung (Konsole)

1. Melde dich bei der an AWS-Managementkonsole und öffne die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home.
2. Wählen Sie die Verteilung mit dem Ursprung aus, den Sie aktualisieren möchten.
3. Wählen Sie den Tab Ursprünge aus.
4. Wählen Sie den zu aktualisierenden Ursprung und dann Edit (Bearbeiten) aus.
5. Wählen Sie bei Enable Origin Shield (Origin Shield aktivieren) die Option Yes (Ja) aus.
6. Wählen Sie bei Origin Shield Region (Origin-Shield-Region) die AWS -Region aus, in der Sie Origin Shield aktivieren möchten. Informationen zur Auswahl einer Region finden Sie unter Wählen Sie das Symbol AWS Region für Origin Shield.
7. Wählen Sie Änderungen speichern aus.

Wenn der Verteilungsstatus Deployed (Bereitgestellt) lautet, ist Origin Shield bereit. Das dauert ein paar Minuten.

So aktivieren Sie Origin Shield für einen neuen Ursprung (Konsole)

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home.
2. Gehen Sie wie folgt vor, um den neuen Ursprung in einer vorhandenen Verteilung zu erstellen:

   1. Wählen Sie die Verteilung aus, in der Sie den Ursprung erstellen möchten.
   2. Wählen Sie Create Origin (Ursprung erstellen) aus und fahren Sie mit Schritt 3 fort.

   Gehen Sie wie folgt vor, um den neuen Ursprung in einer neuen Standarddistribution zu erstellen:

   1. Gehen Sie wie folgt vor, um eine Standarddistribution in der Konsole zu erstellen. Weitere Informationen finden Sie unter Erstellen Sie eine CloudFront Distribution in der Konsole.
   2. Wählen Sie im Abschnitt Einstellungen die Option Ursprungseinstellungen anpassen aus. Fahren Sie mit Schritt 3 fort.
3. Wählen Sie bei Enable Origin Shield (Origin Shield aktivieren) die Option Yes (Ja) aus.
4. Wählen Sie bei Origin Shield Region (Origin-Shield-Region) die AWS -Region aus, in der Sie Origin Shield aktivieren möchten. Informationen zur Auswahl einer Region finden Sie unter Wählen Sie das Symbol AWS Region für Origin Shield.
5. Folgen Sie den Schritten in der Konsole, um die Erstellung Ihres Ursprungs oder Ihrer Distribution abzuschließen.

Wenn der Verteilungsstatus Deployed (Bereitgestellt) lautet, ist Origin Shield bereit. Das dauert ein paar Minuten.

CloudFormation

Um Origin Shield mit zu aktivieren CloudFormation, verwende die `OriginShield` `Origin` Eigenschaft im Eigenschaftstyp einer `AWS::CloudFront::Distribution` Ressource. Sie können die `OriginShield`-Eigenschaft einem vorhandenen `Origin` hinzufügen oder sie einschließen, wenn Sie einen neuen `Origin` erstellen.

Das folgende Beispiel zeigt die Syntax im YAML-Format für die Aktivierung von `OriginShield` in der Region USA West (Oregon) (`us-west-2`). Informationen zur Auswahl einer Region finden Sie unter the section called “Wählen Sie das Symbol AWS Region für Origin Shield”. In diesem Beispiel wird nur der `Origin`-Eigenschaftstyp und nicht die gesamte `AWS::CloudFront::Distribution`-Ressource angezeigt.

```
Origins:
- DomainName: 3ae97e9482b0d011.mediapackage.us-west-2.amazonaws.com
  Id: Example-EMP-3ae97e9482b0d011
  OriginShield:
    Enabled: true
    OriginShieldRegion: us-west-2
  CustomOriginConfig:
    OriginProtocolPolicy: match-viewer
    OriginSSLProtocols: TLSv1
```

Weitere Informationen findest du unter AWS::CloudFront: :Distribution Origin im Abschnitt Ressourcen- und Eigenschaftenreferenzen des AWS CloudFormation Benutzerhandbuchs.

API

Um Origin Shield mit der CloudFront API mithilfe der AWS SDKs oder AWS Command Line Interface (AWS CLI) zu aktivieren, verwende den `OriginShield` Typ. Sie geben `OriginShield` in einem `Origin` in einer `DistributionConfig` an. Informationen zu diesem `OriginShield` Typ finden Sie in den folgenden Informationen in der Amazon CloudFront API-Referenz.

- OriginShield(Typ)
- Ursprung (Typ)
- DistributionConfig(Typ)
- UpdateDistribution(Betrieb)
- CreateDistribution(Betrieb)

Die spezifische Syntax für die Verwendung dieser Typen und Vorgänge variiert je nach SDK, CLI oder API-Client. Weitere Informationen finden Sie in der Referenzdokumentation zu Ihrem SDK, der CLI oder dem Client.

Schätzung der Kosten für Origin Shield

Für Origin Shield fallen Gebühren basierend auf der Anzahl der Anfragen an, die als inkrementelle Ebene an Origin Shield gelangen.

Bei dynamischen (nicht zwischenspeicherbaren) Anforderungen, die an den Ursprung weitergeleitet werden, ist Origin Shield immer eine inkrementelle Ebene. Dynamische Anforderungen verwenden die HTTP-Methoden `PUT`, `POST`, `PATCH` und `DELETE`.

`GET`- und `HEAD`-Anforderungen mit einer Time-to-Live (TTL)-Einstellung von weniger als 3600 Sekunden werden als dynamische Anforderungen betrachtet. Darüber hinaus gelten `GET`- und `HEAD`-Anforderungen, bei denen das Caching deaktiviert wurde, auch als dynamische Anforderungen.

Verwenden Sie die folgende Formel, um Ihre Gebühren für Origin Shield für dynamische Anforderungen zu schätzen:

Gesamtzahl der dynamischen Anforderungen x Origin Shield-Ladung pro 10 000 Anforderungen/10 000

Bei nicht dynamischen Anforderungen mit den HTTP-Methoden `GET`, `HEAD` und `OPTIONS` ist Origin Shield manchmal eine inkrementelle Ebene. Wenn du Origin Shield aktivierst, wählst du das AWS-Region für Origin Shield. Bei Anfragen, die natürlich an den regionalen Edge-Cache in derselben Region wie Origin Shield gehen, wird Origin Shield übersprungen. Für diese Anforderungen fallen keine Origin-Shield-Gebühren an. Für Anforderungen, die an einen regionalen Edge-Cache in einer anderen Region als Origin Shield gelangen und dann zu Origin Shield wechseln, ist Origin Shield eine inkrementelle Ebene. Für diese Anfragen fallen Origin-Shield-Gebühren an.

Verwenden Sie die folgende Formel, um Ihre Gebühren für Origin Shield für zwischenspeicherbare Anfragen zu schätzen:

Gesamtzahl der zwischengespeicherten Anforderungen x (1 – Cache-Zugriffsrate) x Prozentsatz der Anforderungen, die von einem regionalen Edge-Cache in einer anderen Region an Origin Shield gelangen x Origin Shield-Ladung pro 10 000 Anforderungen /10 000

Weitere Informationen über die Gebühr pro 10 000 Anfragen für Origin Shield finden Sie unter CloudFront – Preise.

Hochverfügbarkeit bei Origin Shield

Origin Shield nutzt die Funktion für CloudFront regionale Edge-Caches. Jeder dieser Edge-Caches wird in einer AWS Region erstellt und verwendet mindestens drei Availability Zones mit Flotten von auto-scaling Amazon EC2 EC2-Instances. Verbindungen von CloudFront-Standorten mit Origin Shield verwenden auch die aktive Fehlerverfolgung für jede Anforderung, um die Anforderung automatisch an einen sekundären Origin Shield-Speicherort weiterzuleiten, wenn der primäre Speicherort von Origin Shield nicht verfügbar ist.

Wie Origin Shield mit anderen CloudFront Funktionen interagiert

In den folgenden Abschnitten wird erläutert, wie Origin Shield mit anderen CloudFront-Funktionen interagiert.

Origin Shield und CloudFront Protokollierung

Um zu sehen, wann Origin Shield eine Anforderung verarbeitet hat, müssen Sie eine der folgenden Optionen aktivieren:

- CloudFront Standardprotokolle (Zugriffsprotokolle). Standardprotokolle werden kostenlos zur Verfügung gestellt.
- CloudFront Zugriffsprotokolle in Echtzeit. Für die Verwendung von Echtzeit-Zugriffsprotokollen fallen zusätzliche Gebühren an. Siehe CloudFrontAmazon-Preise.

Cache-Treffer von Origin Shield werden wie `OriginShieldHit` im `x-edge-detailed-result-type` Feld in den CloudFront Protokollen angezeigt. Origin Shield nutzt die CloudFront regionalen Edge-Caches von Amazon. Wenn eine Anfrage von einem CloudFront Edge-Standort an den regionalen Edge-Cache weitergeleitet wird, der als Origin Shield fungiert, wird sie `Hit` in den Protokollen als a gemeldet, nicht als`OriginShieldHit`.

Origin Shield und Ursprungsgruppen

Origin Shield ist mit CloudFront -Ursprungsgruppen kompatibel. Da das Origin Shield eine Eigenschaft des Ursprungs ist, werden Anfragen für jeden Ursprung immer über Origin Shield geleitet, selbst wenn der Ursprung Teil einer Ursprungsgruppe ist. Leitet die Anfrage für eine bestimmte Anfrage über das Origin Shield des primären Ursprungs an den primären Ursprung in der Ursprungsgruppe weiter. CloudFront Wenn diese Anfrage fehlschlägt (gemäß den Failover-Kriterien der Ursprungsgruppe), wird CloudFront die Anfrage über Origin Shield des sekundären Ursprungs an den sekundären Ursprung weitergeleitet.

Origin Shield und Lambda@Edge

Origin Shield wirkt sich nicht auf die Funktionalität von Lambda @Edge-Funktionen aus, kann sich jedoch auf die AWS -Region auswirken, in der diese Funktionen ausgeführt werden.

Wenn du Origin Shield mit Lambda @Edge verwendest, werden auf den Ursprung gerichtete Trigger (Origin Request und Origin Response) in der AWS Region ausgeführt, in der Origin Shield aktiviert ist. Wenn der primäre Origin Shield-Standort nicht verfügbar ist und Anfragen CloudFront an einen sekundären Origin Shield-Standort weitergeleitet werden, wechseln Lambda @Edge -Trigger ebenfalls dazu, den sekundären Origin Shield-Standort zu verwenden.

Viewer-facing Auslöser sind nicht betroffen.

Optimieren Sie die Hochverfügbarkeit mit CloudFront Origin-Failover

Sie können das Origin-Failover für Szenarien einrichten CloudFront , die eine hohe Verfügbarkeit erfordern. Zunächst erstellen Sie eine Ursprungsgruppe mit zwei Ursprüngen: einem primären und einem sekundären. Wenn der primäre Ursprung nicht verfügbar ist oder bestimmte HTTP-Antwortstatuscodes zurückgibt, die auf einen Fehler hinweisen, wechselt CloudFront automatisch zum sekundären Ursprung.

Zum Einrichten eines Origin Failovers müssen Sie eine Verteilung mit mindestens zwei Ursprüngen haben. Anschließend erstellen Sie eine Ursprungsgruppe für Ihre Verteilung, die die beiden Ursprünge enthält, und von denen Sie einen als primär einstellen. Schließlich erstellen oder aktualisieren Sie ein Cache-Verhalten, um die Ursprungsgruppe zu verwenden.

Informationen zu den Schritten zum Einrichten von Ursprungsgruppen und zum Konfigurieren bestimmter Ursprungs-Failover-Optionen finden Sie unter Erstellen einer Ursprungsgruppe.

Nachdem Sie das Origin-Failover für ein Cache-Verhalten konfiguriert haben, gehen CloudFront Sie für Viewer-Anfragen wie folgt vor:

- Bei einem Cache-Treffer wird das angeforderte Objekt CloudFront zurückgegeben.
- Bei einem Cache-Fehler wird die CloudFront Anfrage an den primären Ursprung in der Ursprungsgruppe weitergeleitet.
- Wenn der primäre Ursprung einen Statuscode zurückgibt, der nicht für ein Failover konfiguriert ist, z. B. ein HTTP 2xx- oder 3xx-Statuscode, wird das angeforderte CloudFront Objekt dem Viewer zugestellt.

- Wenn eine der folgenden Bedingungen eintritt:
  - Der primäre Ursprung gibt einen HTTP-Statuscode zurück, den Sie für das Failover konfiguriert haben
  - CloudFront kann keine Verbindung zum primären Ursprung herstellen (wenn 503 als Failovercode festgelegt ist)
  - Die Antwort vom primären Ursprung dauert zu lange (Timeout) (wenn 504 als Failover-Code festgelegt ist)

  CloudFront Leitet die Anfrage dann an den sekundären Ursprung in der Ursprungsgruppe weiter.

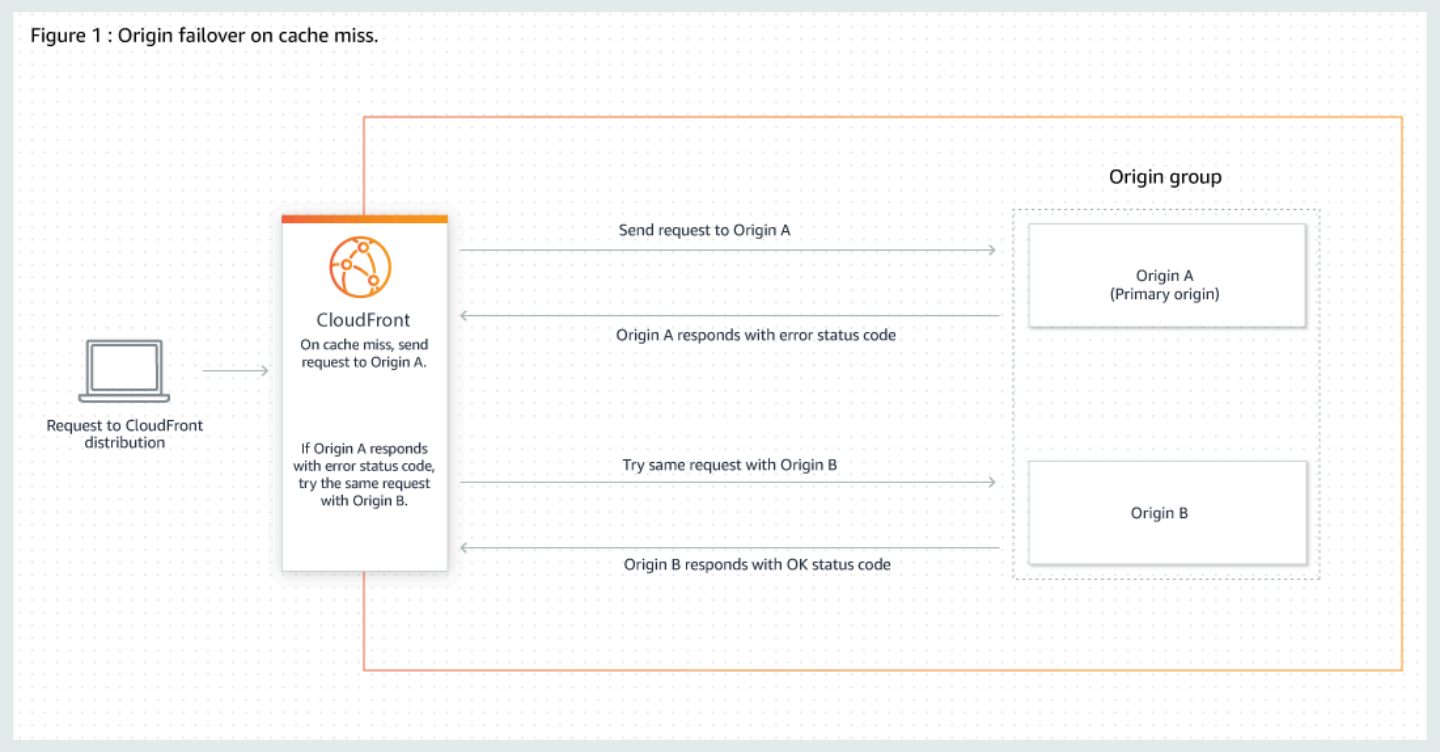
> **Note**
>
> In einigen Anwendungsfällen, z. B. beim Streamen von Videoinhalten, CloudFront möchten Sie möglicherweise schnell einen Failover zum sekundären Ursprung durchführen. Informationen dazu, wie schnell ein CloudFront Failover zum sekundären Ursprung erfolgen soll, finden Sie unterSteuern von Timeouts und Verbindungsversuchen für Ursprünge.

CloudFront leitet alle eingehenden Anfragen an den primären Ursprung weiter, auch wenn bei einer früheren Anfrage ein Failover an den sekundären Ursprung durchgeführt wurde. CloudFront sendet Anfragen nur an den sekundären Ursprung, nachdem eine Anfrage an den primären Ursprung fehlgeschlagen ist.

CloudFront Failover zum sekundären Ursprung erfolgt nur, wenn die HTTP-Methode der Viewer-Anfrage `GETHEAD`, oder lautet`OPTIONS`. CloudFront führt kein Failover durch, wenn der Viewer eine andere HTTP-Methode sendet (z. B. `POSTPUT`, usw.).

> **Note**
>
> CloudFront führt kein Failover durch, wenn sie nicht Zwischengespeicherte HTTP-Methoden in Ihrem Cache als Verhalten festgelegt `OPTIONS` sind.

Das folgende Diagramm veranschaulicht, wie Origin Failover funktioniert.



Themen

- Erstellen einer Ursprungsgruppe
- Steuern von Timeouts und Verbindungsversuchen für Ursprünge
- Verwenden von Origin Failover mit Lambda@Edge-Funktionen
- Verwenden von benutzerdefinierten Fehlerseiten mit Ursprung-Failover

Erstellen einer Ursprungsgruppe

So erstellen Sie eine Ursprungsgruppe

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home.
2. Wählen Sie die Verteilung aus, für die Sie die Ursprungsgruppe erstellen möchten.
3. Wählen Sie den Tab Ursprünge aus.
4. Stellen Sie sicher, dass die Verteilung mehr als einen Ursprung hat. Wenn dies nicht der Fall ist, fügen Sie einen zweiten Ursprung hinzu.
5. Wählen Sie auf der Registerkarte Ursprünge im Bereich Ursprungsgruppen die Option Ursprungsgruppe erstellen aus.

6. Wählen Sie die Ursprünge für die Ursprungsgruppe aus. Nachdem Sie Ursprünge hinzugefügt haben, verwenden Sie die Pfeile, um die Priorität festzulegen, d. h., welcher Ursprung primär und welcher sekundär ist.
7. Geben Sie einen Namen für die Ursprungsgruppe ein.
8. Wählen Sie die HTTP-Statuscodes aus, die als Failover-Kriterien verwendet werden sollen. Sie können eine beliebige Kombination der folgenden Statuscodes wählen: 400, 403, 404, 416, 429, 500, 502, 503 oder 504. Wenn eine Antwort mit einem der von Ihnen angegebenen Statuscodes CloudFront empfangen wird, erfolgt ein Failover zum sekundären Ursprung.

   > **Note**
   >
   > CloudFront Ein Failover zum sekundären Ursprung erfolgt nur, wenn die HTTP-Methode der Viewer-Anfrage auf `GETHEAD`, oder gesetzt ist`OPTIONS`. CloudFront führt kein Failover durch, wenn der Viewer eine andere HTTP-Methode sendet (z. B. `POSTPUT`, usw.).

9. Geben Sie unter Auswahlkriterien für den Ursprung an, wie Ihre Ursprünge ausgewählt werden, wenn Ihre Distribution Viewer-Anforderungen weiterleitet. Sie können die folgenden Optionen auswählen:

   Standard

   CloudFront verwendet die Standardurftspriorität, die Sie auf der Seite „Einstellungen“ angeben.

   Medienqualitätswert

   CloudFront verfolgt und verwendet diese Punktzahl, um den ersten Ursprung zu ermitteln, an den die Anfrage weitergeleitet werden soll. Dies berechtigt auch CloudFront dazu, asynchrone `HEAD` Anfragen an den alternativen Absender in der Ursprungsgruppe zu stellen, um dessen Medienqualitätsfaktor zu ermitteln. Sie können diese Option nur für AWS Elemental MediaPackage v2-Ursprünge wählen. Weitere Informationen finden Sie unter Media Quality-Aware Resiliency.
10. Wählen Sie Ursprungsgruppe erstellen aus.

Stellen Sie sicher, dass Sie Ihre Ursprungsgruppe als Ursprung für das Cacheverhalten Ihrer Distribution angeben. Weitere Informationen finden Sie unter Name.

Steuern von Timeouts und Verbindungsversuchen für Ursprünge

CloudFront Versucht standardmäßig 30 Sekunden lang, eine Verbindung zum primären Ursprung in einer Ursprungsgruppe herzustellen (3 Verbindungsversuche von jeweils 10 Sekunden), bevor ein Failover zum sekundären Ursprung erfolgt. In einigen Anwendungsfällen, z. B. beim Streamen von Videoinhalten, CloudFront möchten Sie möglicherweise schneller einen Failover zum sekundären Ursprung durchführen. Sie können die folgenden Einstellungen anpassen, um zu beeinflussen, wie schnell ein CloudFront Failover zum sekundären Ursprung erfolgt. Wenn der Ursprung ein sekundärer Ursprung ist oder ein Ursprung, der nicht Teil einer Ursprungsgruppe ist, wirken sich diese Einstellungen darauf aus, wie schnell eine HTTP 504-Antwort an den Viewer CloudFront zurückgegeben wird.

Wenn Sie ein Failover schneller ausführen möchten, geben Sie ein kürzeres Verbindungstimeout, weniger Verbindungsversuche, oder beides an. Für benutzerdefinierte Ursprünge (einschließlich Amazon S3-Bucket-Ursprünge, die mit statischem Website-Hosting konfiguriert sind) können Sie auch das Timeout der Ursprungsantwort anpassen.

Timeout der Ursprungsverbindung

Die Timeout-Einstellung für die Originalverbindung wirkt sich darauf aus, wie lange beim Versuch, eine Verbindung zum Ursprung herzustellen, CloudFront gewartet wird. CloudFront Wartet standardmäßig 10 Sekunden, um eine Verbindung herzustellen. Sie können jedoch 1—10 Sekunden (einschließlich) angeben. Weitere Informationen finden Sie unter Verbindungstimeout.

Verbindungsversuche zum Ursprung

Die Einstellung für Verbindungsversuche über den Ursprung wirkt sich darauf aus, wie oft CloudFront versucht wird, eine Verbindung zum Ursprung herzustellen. CloudFront Versucht standardmäßig dreimal, eine Verbindung herzustellen, Sie können jedoch 1—3 (einschließlich) angeben. Weitere Informationen finden Sie unter Verbindungsversuche.

Für einen benutzerdefinierten Ursprung (einschließlich eines Amazon S3-Buckets, der mit statischem Website-Hosting konfiguriert ist) wirkt sich diese Einstellung auch darauf aus, wie oft CloudFront versucht wird, im Falle eines Timeouts der Ursprungsantwort eine Antwort vom Ursprung zu erhalten.

Ursprungs-Reaktions-Timeout

Der Timeout der ursprünglichen Antwort, auch bekannt als Timeout für den ursprünglichen Lesevorgang, wirkt sich darauf aus, wie lange auf den Empfang einer Antwort (oder auf den

Erhalt der vollständigen Antwort) vom Ursprung CloudFront gewartet wird. CloudFront Wartet standardmäßig 30 Sekunden, Sie können jedoch 1—120 Sekunden (einschließlich) angeben. Weitere Informationen finden Sie unter Antwort-Timeout.

So ändern Sie diese Einstellungen

So ändern Sie diese Einstellungen in der CloudFront-Konsole

- Für einen neuen Ursprung oder eine neue Verteilung geben Sie diese Werte an, wenn Sie die Ressource erstellen.
- Für einen vorhandenen Ursprung in einer vorhandenen Verteilung geben Sie diese Werte an, wenn Sie den Ursprung bearbeiten.

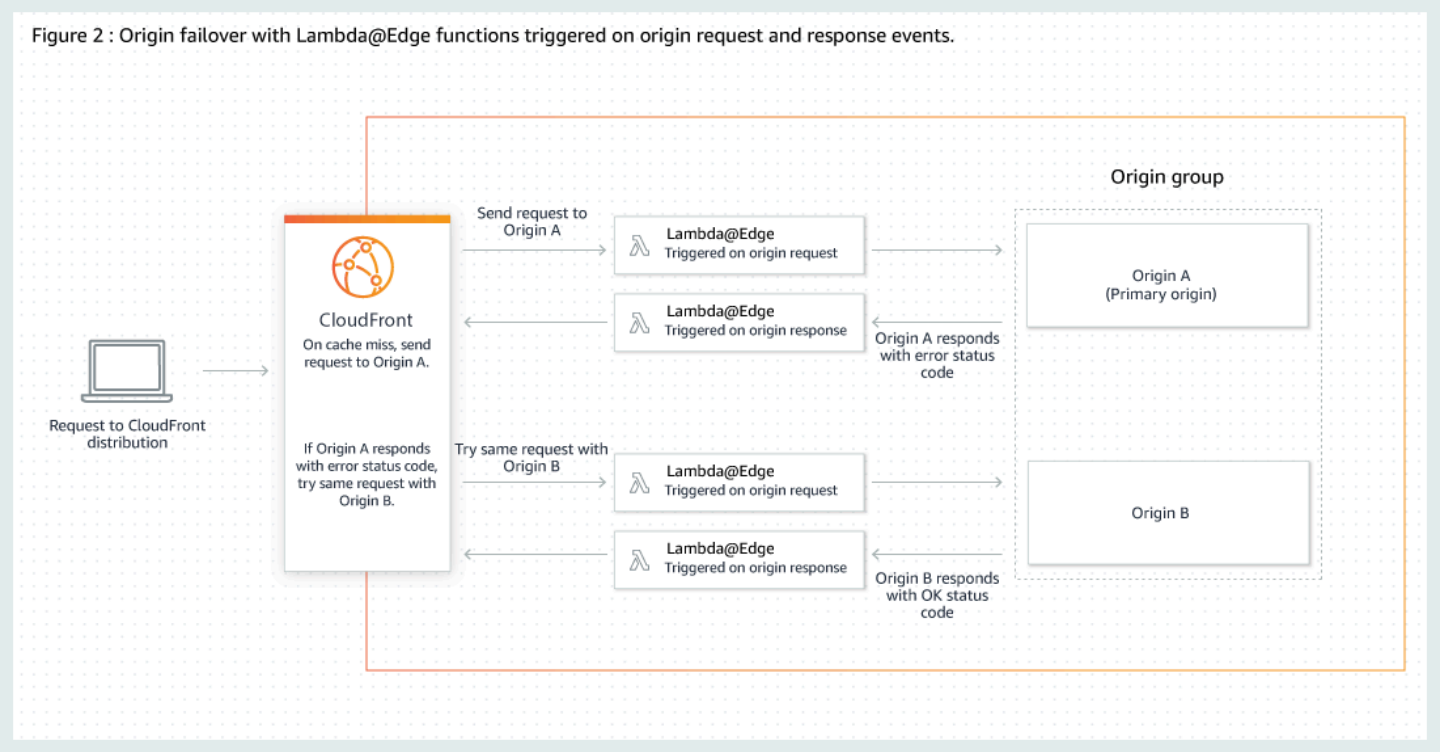
Weitere Informationen finden Sie unter Referenz für alle Distributionseinstellungen.

Verwenden von Origin Failover mit Lambda@Edge-Funktionen

Sie können Lambda @Edge -Funktionen mit CloudFront Verteilungen verwenden, die Sie mit Ursprungsgruppen eingerichtet haben. Um eine Lambda-Funktion zu verwenden, geben Sie sie in einem Ursprungsanforderungs- oder Ursprungsantwort-Auslöser für eine Ursprungsgruppe an, wenn Sie die Cache-Verhaltenweise erstellen. Wenn Sie eine Lambda@Edge-Funktion mit einer Ursprungsgruppe verwenden, kann die Funktion zweimal für eine einzelne Viewer-Anfrage ausgelöst werden. Betrachten Sie beispielsweise folgendes Szenario:

1. Sie erstellen eine Lambda@Edge-Funktion mit einem Ursprungsanfrage-Auslöser.
2. Die Lambda-Funktion wird einmal ausgelöst, wenn eine Anfrage an den primären Ursprung CloudFront gesendet wird (bei einem Cache-Fehler).
3. Der primäre Ursprung antwortet mit einem HTTP-Statuscode, der für das Failover konfiguriert ist.
4. Die Lambda-Funktion wird erneut ausgelöst, wenn dieselbe Anfrage CloudFront an den sekundären Ursprung gesendet wird.

Das folgende Diagramm illustriert die Funktionsweise von Origin Failover, wenn Sie eine Lambda@Edge-Funktion in eine Ursprungsanfrage oder einen Antwort-Auslöser einbeziehen.



Weitere Informationen zur Verwendung von Lambda@Edge-Auslösern finden Sie unter the section called “Hinzufügen von Auslösern für eine Lambda@Edge-Funktion”.

Weitere Informationen zur Verwaltung des DNS-Failovers finden Sie unter Konfigurieren von DNS-Failover im Entwicklerhandbuch für Amazon Route 53.

Verwenden von benutzerdefinierten Fehlerseiten mit Ursprung-Failover

Sie können benutzerdefinierte Fehlerseiten mit Ursprungsgruppen ähnlich verwenden wie Ursprünge, die nicht für Origin Failover konfiguriert wurden.

Wenn Sie Origin-Failover verwenden, können Sie so konfigurieren, CloudFront dass eine benutzerdefinierte Fehlerseite für den primären oder sekundären Ursprung (oder für beide) zurückgegeben wird:

- Eine benutzerdefinierte Fehlerseite für den primären Ursprung zurückgeben — Wenn der primäre Ursprung einen HTTP-Statuscode zurückgibt, der nicht für das Failover konfiguriert ist, wird die benutzerdefinierte Fehlerseite an die Zuschauer CloudFront zurückgegeben.
- Eine benutzerdefinierte Fehlerseite für den sekundären Ursprung zurückgeben — Wenn ein Fehlerstatuscode vom sekundären Ursprung CloudFront empfangen wird, wird die benutzerdefinierte Fehlerseite CloudFront zurückgegeben.

Weitere Hinweise zur Verwendung benutzerdefinierter Fehlerseiten mit CloudFront finden Sie unterErstellen von benutzerdefinierten Fehlerantworten.

Verwalten der Dauer, die Inhalte im Cache bleiben (Ablauf)

Sie können steuern, wie lange Ihre Dateien in einem CloudFront Cache bleiben, bevor eine weitere Anfrage an Ihren Ursprung CloudFront weitergeleitet wird. Indem Sie die Dauer reduzieren, können Sie dynamische Inhalte bereitstellen. Eine Erhöhung der Dauer bedeutet, dass Ihre Benutzer eine bessere Leistung erhalten, da es wahrscheinlicher ist, dass Ihre Dateien direkt vom Edge-Cache bereitgestellt werden. Eine längere Dauer verringert darüber hinaus die Last auf Ihrem Ursprung.

Stellt in der CloudFront Regel eine Datei von einem Edge-Standort aus bereit, bis die von Ihnen angegebene Cachedauer abgelaufen ist, d. h. bis die Datei abläuft. Wenn der Edge-Standort nach Ablauf das nächste Mal eine Anforderung für die Datei erhält, CloudFront leitet er die Anfrage an den Ursprung weiter, um zu überprüfen, ob der Cache die neueste Version der Datei enthält. Die Antwort vom Ursprung hängt davon ab, ob die Datei geändert wurde:

- Wenn der CloudFront Cache bereits über die neueste Version verfügt, gibt der Ursprung einen Statuscode `304 Not Modified` zurück.
- Wenn der CloudFront Cache nicht über die neueste Version verfügt, gibt der Ursprung einen Statuscode `200 OK` und die neueste Version der Datei zurück.

Wenn eine Datei an einem Edge-Speicherort nicht häufig angefordert wird, löschen Sie die Datei CloudFront möglicherweise — entfernen Sie die Datei vor ihrem Ablaufdatum —, um Platz für Dateien zu schaffen, die in jüngerer Zeit angefordert wurden.

Wir empfehlen, die Cache-Dauer zu verwalten, indem Sie die Cache-Richtlinie Ihrer Distribution aktualisieren. Wenn Sie keine Cache-Richtlinie verwenden, beträgt die Standard-TTL (Time to Live) 24 Stunden. Sie können jedoch die folgenden Einstellungen aktualisieren, um die Standardeinstellung zu überschreiben:

- Um die Cachedauer für alle Dateien zu ändern, die demselben Pfadmuster entsprechen, können Sie die CloudFront Einstellungen für Minimale TTL, Maximale TTL und Standard-TTL für ein Cacheverhalten ändern. Weitere Informationen zu den einzelnen Einstellungen finden Sie unter Mindest-TTL, Höchst-TTL und Standard-TTL.
- Um die Cache-Dauer für eine einzelne Datei zu ändern, können Sie Ihren Ursprung so konfigurieren, dass ein `Cache-Control`-Header mit der `max-age`- oder `s-maxage`-Richtlinie

oder ein `Expires`-Header zu der Datei hinzugefügt wird. Weitere Informationen finden Sie unter Verwenden von Headern zum Steuern der Cache-Dauer für einzelne Objekte.

Weitere Informationen dazu, wie Mindest-TTL, Standard-TTL und Höchst-TTL mit `max-age`- und `s-maxage`-Richtlinien und dem `Expires`-Header-Feld interagieren, finden Sie unter the section called "Geben Sie an, wie lange Objekte zwischengespeichert werden CloudFront".

Sie können auch steuern, wie lange Fehler (z. B. `404 Not Found`) in einem CloudFront Cache verbleiben, bevor erneut CloudFront versucht wird, das angeforderte Objekt abzurufen, indem Sie eine weitere Anfrage an Ihren Ursprung weiterleiten. Weitere Informationen finden Sie unter the section called "Wie CloudFront verarbeitet die HTTP 4xx- und 5xx-Statuscodes von Ihrem Ursprung".

Themen

- Verwenden von Headern zum Steuern der Cache-Dauer für einzelne Objekte
- Bereitstellung veralteter (abgelaufener) Inhalte
- Geben Sie an, wie lange Objekte zwischengespeichert werden CloudFront
- Hinzufügen von Headern zu Ihren Objekten mithilfe der Amazon-S3-Konsole

Verwenden von Headern zum Steuern der Cache-Dauer für einzelne Objekte

Sie können die Header `Cache-Control` und `Expires` verwenden, um zu steuern, wie lange Objekte im Cache zwischengespeichert werden. Die Einstellungen für Mindest-TTL, Standard-TTL und Höchst-TTL wirken sich auch auf die Cache-Dauer aus. Im Folgenden finden Sie einen Überblick darüber, wie sich Header auf die Cache-Dauer auswirken können:

- Mit der `Cache-Control max-age` Direktive können Sie angeben, wie lange (in Sekunden) ein Objekt im Cache verbleiben soll, bevor CloudFront es erneut vom Originalserver abgerufen wird. Die unterstützte Mindestablaufzeit CloudFront beträgt 0 Sekunden. Die Höchstwert beträgt 100 Jahre. Geben Sie den Wert im folgenden Format an:

  ```
  Cache-Control: max-age=seconds
  ```

  Die folgende Direktive weist beispielsweise CloudFront an, dass das zugehörige Objekt 3600 Sekunden (eine Stunde) lang im Cache belassen werden soll:

  ```
  Cache-Control: max-age=3600
  ```

Wenn Sie möchten, dass Objekte für eine andere Dauer in CloudFront Edge-Caches verbleiben als in Browser-Caches, können Sie die Direktiven `Cache-Control max-age` und `Cache-Control s-maxage` zusammen verwenden. Weitere Informationen finden Sie unter Geben Sie an, wie lange Objekte zwischengespeichert werden CloudFront.

- Mit dem `Expires` Header-Feld können Sie ein Ablaufdatum und eine Ablaufzeit in dem in RFC 2616, Hypertext Transfer Protocol -- HTTP/1.1 Abschnitt 3.3.1, Vollständiges Datum, angegebenen Format angeben, zum Beispiel:

  ```
  Sat, 27 Jun 2015 23:59:59 GMT
  ```

Wir empfehlen, die `Cache-Control max-age`-Richtlinie anstelle des `Expires`-Header-Felds zum Steuern der Zwischenspeicherung von Objekten zu verwenden. Wenn Sie sowohl für `Cache-Control max-age` als auch für `Expires` Werte festlegen, verwendet CloudFront nur den Wert von `Cache-Control max-age`.

Weitere Informationen finden Sie unter Geben Sie an, wie lange Objekte zwischengespeichert werden CloudFront.

Sie können die HTTP `Cache-Control` - oder `Pragma` Header-Felder in einer `GET` Anfrage eines Viewers nicht verwenden, um die Rückkehr CloudFront zum Originalserver für das Objekt zu erzwingen. CloudFront ignoriert diese Header-Felder in Viewer-Anfragen.

Weitere Informationen zu den `Expires` Header-Feldern `Cache-Control` und finden Sie in den folgenden Abschnitten in RFC 2616, Hypertext Transfer Protocol --: HTTP/1.1

- Abschnitt 14.9 Cache-Kontrolle
- Abschnitt 14.21 Läuft ab

Bereitstellung veralteter (abgelaufener) Inhalte

CloudFront unterstützt die Steueranweisungen `Stale-While-Revalidate` und `Stale-If-Error` Cache. Mit diesen Direktiven können Sie angeben, wie lange veraltete Inhalte für Viewer verfügbar sind.

Themen

- Stale-While-Revalidate
- Stale-If-Error

- Verwenden beider Direktiven

`Stale-While-Revalidate`

Diese Direktive CloudFront ermöglicht es, veraltete Inhalte aus dem Cache bereitzustellen, während CloudFront asynchron eine neue Version vom Ursprung abgerufen wird. Dies verbessert die Latenz, da Viewer sofort Antworten von Edge-Standorten erhalten, ohne auf den Abruf im Hintergrund warten zu müssen. Neue Inhalte werden für zukünftige Anforderungen im Hintergrund geladen.

Example Beispiel: **`Stale-While-Revalidate`**

CloudFront führt Folgendes aus, wenn Sie den `Cache-Control` Header so einstellen, dass er diese Direktiven verwendet.

```
Cache-Control: max-age=3600, stale-while-revalidate=600
```

1. CloudFront speichert eine Antwort für eine Stunde (`max-age=3600`) im Cache.
2. Wenn eine Anfrage nach Ablauf dieser Dauer gestellt wird, wird der CloudFront veraltete Inhalt bereitgestellt und gleichzeitig eine Anfrage an den Ursprung gesendet, um den zwischengespeicherten Inhalt erneut zu validieren und zu aktualisieren.
3. Während der Überprüfung des Inhalts wird der veraltete Inhalt bis CloudFront zu 10 Minuten lang bereitgestellt (). `stale-while-revalidate=600`

> **Note**
>
> CloudFront stellt den veralteten Inhalt bis zum Wert der `stale-while-revalidate` Direktive oder bis zum Wert der CloudFront maximalen TTL bereit, je nachdem, welcher Wert niedriger ist. Nach Ablauf der maximalen TTL-Dauer ist das veraltete Objekt unabhängig vom Wert `stale-while-revalidate` nicht mehr im Edge-Cache verfügbar.

`Stale-If-Error`

Diese Direktive CloudFront ermöglicht es, veraltete Inhalte aus dem Cache bereitzustellen, wenn der Ursprung nicht erreichbar ist oder einen Fehlercode zwischen 500 und 600 zurückgibt. Dadurch wird sichergestellt, dass Zuschauer auch während eines Origin-Ausfalls auf Inhalte zugreifen können.

Example Beispiel: `Stale-If-Error`

CloudFront führt Folgendes aus, wenn Sie den `Cache-Control` Header so einstellen, dass er diese Anweisungen verwendet.

```
Cache-Control: max-age=3600, stale-if-error=86400
```

1. CloudFront speichert die Antwort für eine Stunde (`max-age=3600`) im Cache.
2. Wenn der Ursprung nach Ablauf dieser Zeit ausgefallen ist oder einen Fehler zurückgibtCloudFront , wird der veraltete Inhalt bis zu 24 Stunden lang bereitgestellt () `stale-if-error=86400`
3. Wenn Sie benutzerdefinierte Fehlerantworten konfiguriert haben, CloudFront wird versucht, den veralteten Inhalt bereitzustellen, wenn innerhalb der angegebenen `stale-if-error` Dauer ein Fehler auftritt. Wenn der veraltete Inhalt nicht verfügbar ist, CloudFront werden die benutzerdefinierten Fehlerantworten bereitgestellt, die Sie für den entsprechenden Fehlerstatuscode konfiguriert haben. Weitere Informationen finden Sie unter Erstellen von benutzerdefinierten Fehlerantworten.

> **Hinweise**
>
> - CloudFront stellt den veralteten Inhalt bis zum Wert der `stale-if-error` Direktive oder dem Wert der CloudFront maximalen TTL bereit, je nachdem, welcher Wert niedriger ist. Nach Ablauf der maximalen TTL-Dauer ist das veraltete Objekt unabhängig vom Wert `stale-if-error` nicht mehr im Edge-Cache verfügbar.
> - Wenn Sie keine Fehlerantworten konfigurieren `stale-if-error` oder anpassen, CloudFront gibt das veraltete Objekt zurück oder leitet die Fehlerantwort zurück an den Viewer weiter, je nachdem, ob sich das angeforderte Objekt im Edge-Cache befindet oder nicht. Weitere Informationen finden Sie unter Wie CloudFront verarbeitet Fehler, wenn Sie keine benutzerdefinierten Fehlerseiten konfiguriert haben.

Verwenden beider Direktiven

`stale-while-revalidate` und `stale-if-error` sind unabhängige Cache-Kontrolldirektiven, die zusammen verwendet werden können, um die Latenz zu reduzieren und einen Puffer hinzuzufügen, damit Ihr Ursprung reagieren oder sich erholen kann.

Example Beispiel: Verwendung beider Direktiven

CloudFront führt Folgendes aus, wenn Sie den `Cache-Control` Header so einstellen, dass er die folgenden Anweisungen verwendet.

```
Cache-Control: max-age=3600, stale-while-revalidate=600, stale-if-error=86400
```

1. CloudFront speichert die Antwort für eine Stunde (`max-age=3600`) im Cache.
2. Wenn eine Anfrage nach Ablauf dieser Dauer gestellt wird, wird CloudFront der veraltete Inhalt für bis zu 10 Minuten (`stale-while-revalidate=600`) bereitgestellt, während der Inhalt erneut überprüft wird.
3. Wenn der Originalserver beim CloudFront Versuch, den Inhalt erneut zu validieren, einen Fehler zurückgibt, CloudFront wird der veraltete Inhalt bis zu 24 Stunden lang () bereitgestellt. `stale-if-error=86400`

Caching stellt ein Gleichgewicht zwischen Leistung und Aktualität her. Die Verwendung von Richtlinien wie `stale-while-revalidate` und `stale-if-error` kann die Leistung und den Benutzerkomfort verbessern. Achten Sie jedoch darauf, dass die Konfigurationen darauf abgestimmt sind, wie aktuell Ihre Inhalte sein sollen. Richtlinien für veraltete Inhalte eignen sich am besten für Anwendungsfälle, in denen Inhalte aktualisiert werden müssen, die neueste Version jedoch nicht unbedingt erforderlich ist. Wenn sich Ihr Inhalt nicht oder nur selten ändert, kann `stale-while-revalidate` außerdem zu unnötigen Netzwerkanforderungen führen. Erwägen Sie stattdessen, eine lange Cachedauer festzulegen.

Geben Sie an, wie lange Objekte zwischengespeichert werden CloudFront

Um zu steuern, wie lange ein Objekt im Cache CloudFront verbleibt, bevor eine weitere Anfrage an den Ursprung gesendet wird, können Sie:

- Legen Sie die minimalen, maximalen und standardmäßigen TTL-Werte für das Cache-Verhalten einer CloudFront Distribution fest. Sie können diese Werte in einer Cache-Richtlinie festlegen, die an das Cache-Verhalten (empfohlen) oder in den Legacy-Cache-Einstellungen angehängt ist.
- Die `Cache-Control`- oder `Expires`-Header in Antworten vom Ursprung einschließen. Diese Header helfen auch dabei zu bestimmen, wie lange ein Browser ein Objekt im Browser-Cache aufbewahrt, bevor eine weitere Anfrage an gesendet wird. CloudFront

In der folgenden Tabelle wird erläutert, wie die vom Ursprung gesendeten `Cache-Control`- und `Expires`-Header mit den TTL-Einstellungen in einem Cache-Verhalten zusammenarbeiten, um das Caching zu beeinflussen.

Ursprung-Header	Mindest-TTL = 0	Mindest-TTL > 0
Der Ursprung fügt dem Objekt eine **`Cache-Control: max-age`** -Richtlinie hinzu	CloudFront Zwischens peichern CloudFront speichert das Objekt für den niedrigeren Wert der `Cache-Control: max-age` Direktive oder den Wert der CloudFront maximalen TTL im Cache. Browser-Caching Browser speichern das Objekt für den Wert der `Cache-Control: max-age` -Richtlinie zwischen.	CloudFront zwischenspeichern CloudFront Das Caching hängt von den Werten der CloudFront minimalen TTL und der maximalen TTL und der Direktive ab: `Cache-Control max-age` • Wenn die minimale TTL `max-age` < die maximale TTL ist, wird das Objekt für den CloudFront Wert der Direktive zwischeng espeichert. `Cache-Control: max-age` • Wenn `max-age` < minimaler TTL, wird das Objekt für den Wert der minimalen TTL CloudFront zwischeng espeichert. CloudFront • Wenn `max-age` > maximale TTL, wird das Objekt für den Wert der maximalen TTL CloudFront zwischeng espeichert. CloudFront

Ursprung-Header	Mindest-TTL = 0	Mindest-TTL > 0
		Browser-Caching Browser speichern das Objekt für den Wert der `Cache-Control: max-age` -Richtlinie zwischen.
Der Ursprung fügt dem Objekt keine **`Cache-Control: max-age`**-Richtlinie hinzu	CloudFront zwischenspeichern CloudFront speichert das Objekt für den Wert der CloudFront Standard-TTL im Cache. Browser-Caching Abhängig vom Browser.	CloudFront zwischenspeichern CloudFront speichert das Objekt im Cache, je nachdem, welcher Wert der CloudFront TTL-Mindestwerte oder der Standard-TTL ist. Browser-Caching Abhängig vom Browser.

Ursprung-Header	Mindest-TTL = 0	Mindest-TTL > 0
Der Ursprung fügt dem Objekt **`Cache-Control: max-age`** - und **`Cache-Control: s-maxage`** -Richtlinien hinzu	CloudFront zwischenspeichern CloudFront speichert das Objekt für den niedrigeren Wert der `Cache-Control: s-maxage` Direktive oder den Wert der CloudFront maximalen TTL im Cache. Browser-Caching Browser speichern das Objekt für den Wert der `Cache-Control max-age` -Richtlinie zwischen.	CloudFront zwischenspeichern CloudFront Das Caching hängt von den Werten der CloudFront minimalen TTL und der maximalen TTL und der Direktive ab: `Cache-Control: s-maxage` • Wenn die minimale TTL `s-maxage` < die maximale TTL ist, wird das Objekt für den CloudFront Wert der Direktive zwischengespeichert. `Cache-Control: s-maxage` • Wenn `s-maxage` < minimaler TTL, wird das Objekt für den Wert der minimalen TTL CloudFront zwischengespeichert. CloudFront • Wenn `s-maxage` > maximale TTL, wird das Objekt für den Wert der maximalen TTL CloudFront zwischengespeichert. CloudFront Browser-Caching

Ursprung-Header	Mindest-TTL = 0	Mindest-TTL > 0
		Browser speichern das Objekt für den Wert der `Cache-Control: max-age` -Richtlinie zwischen.

Ursprung-Header	Mindest-TTL = 0	Mindest-TTL > 0
Der Ursprung fügt dem Objekt einen **`Expires`**-Header hinzu	CloudFront zwischenspeichern CloudFront speichert das Objekt bis zum Datum im `Expires` Header oder bis zum Wert der CloudFront maximalen TTL im Cache, je nachdem, welches Datum früher eintritt. Browser-Caching Browser speichern das Objekt bis zum Datum im `Expires`-Header zwischen.	CloudFront Zwischens peichern CloudFront Das Caching hängt von den Werten der CloudFront minimalen TTL und der maximalen TTL sowie vom Header ab: `Expires` • Wenn die minimale TTL < `Expires` < die maximale TTL ist, wird das Objekt bis zum Datum und der Uhrzeit im Header CloudFron t zwischengespeichert. `Expires` • Wenn `Expires` < minimaler TTL, wird das Objekt für den Wert der minimalen TTL CloudFront zwischeng espeichert. CloudFront • Wenn `Expires` > maximale TTL, wird das Objekt für den Wert der maximalen TTL CloudFront zwischeng espeichert. CloudFront Browser-Caching Browser speichern das Objekt bis zum Datum und der

Ursprung-Header	Mindest-TTL = 0	Mindest-TTL > 0
		Uhrzeit im `Expires`-Header zwischen.
Origin fügt dem Objekt**`Cache-Control: no-cache`**,**`no-store`**, and/or **`private`** Direktiven hinzu	CloudFront und Browser respektieren die Header.	CloudFront Zwischens peichern CloudFront speichert das Objekt für den Wert des CloudFront TTL-Mindestwerts im Cache. Siehe die Warnung unter dieser Tabelle. Browser-Caching Browser berücksichtigen die Header.

Warning

- Wenn Ihre Mindest-TTL größer als 0 ist, wird die Mindest-TTL der Cache-Richtlinie CloudFront verwendet, auch wenn die and/or `private` Direktiven,, in den `Cache-Control: no-cache` `no-store` Quell-Headern vorhanden sind.
  - Wenn der Ursprung erreichbar ist, CloudFront ruft das Objekt vom Ursprung ab und gibt es an den Viewer zurück.
  - Wenn der Ursprung nicht erreichbar ist und der minimale oder maximale TTL-Wert größer als 0 ist, CloudFront wird das Objekt bereitgestellt, das es zuvor vom Ursprung erhalten hat.

  Um dieses Verhalten zu vermeiden, schließen Sie die `Cache-Control: stale-if-error=0`-Richtlinie in das vom Ursprung zurückgegebene Objekt ein. Dies führt CloudFront dazu, dass als Antwort auf zukünftige Anfragen ein Fehler zurückgegeben wird, wenn der Ursprung nicht erreichbar ist, anstatt das Objekt zurückzugeben, das es zuvor vom Ursprung erhalten hat.

- CloudFront speichert den HTTP 501-Statuscode (nicht implementiert) von einem S3-Ursprung nicht im Cache, wenn die Quell-Header die Direktiven`Cache-Control: no-cache`,`no-store`, and/or `private` enthalten. Dies ist das Standardverhalten für einen S3-Ursprung, auch wenn Ihre Einstellung für die Mindest-TTL größer als 0 ist.

Hinweise zum Ändern der Einstellungen für Distributionen mithilfe der CloudFront Konsole finden Sie unter. Eine Verteilung aktualisieren Informationen zum Ändern der Einstellungen für Distributionen mithilfe der CloudFront API finden Sie unter. UpdateDistribution

Hinzufügen von Headern zu Ihren Objekten mithilfe der Amazon-S3-Konsole

Sie können das Header-Feld `Cache-Control` oder `Expires` zu Ihren Amazon-S3-Objekten hinzufügen. Ändern Sie dazu die Metadatenfelder für das Objekt.

So fügen Sie Amazon **`Cache-Control`** S3-Objekten **`ein`** Header-Feld oder Expires hinzu

1. Folgen Sie den Anleitungen im Abschnitt Ersetzen systemdefinierter Metadaten im Thema Bearbeiten von Objektmetadaten in der Amazon-S3-Konsole im Benutzerhandbuch für Amazon S3.
2. Wählen Sie für Key den Namen des Headers, den Sie hinzufügen (Cache-Controloder Expires).
3. Geben Sie für Wert einen Header-Wert ein. Zum Beispiel könnten Sie für einen `Cache-Control`-Header `max-age=86400` eingeben. Für `Expires` könnten Sie ein Ablaufdatum und eine Uhrzeit wie `Wed, 30 Jun 2021 09:28:00 GMT` eingeben.
4. Folgen Sie den restlichen Schritten, um Ihre Metadatenänderungen zu speichern.

Zwischenspeichern von Inhalten auf der Grundlage von Abfragezeichenfolgeparametern

Einige Webanwendungen verwenden zum Senden von Informationen an den Ursprung Abfragezeichenfolgen. Eine Abfragezeichenfolge ist der Teil einer Webanfrage nach dem Zeichen `?`. Die Zeichenfolge kann einen oder mehrere durch das Zeichen & getrennte Parameter enthalten. Im folgenden Beispiel enthält die Abfragezeichenfolge zwei Parameter *`color=red`* und*`size=large`*:

`https://d111111abcdef8.cloudfront.net/images/image.jpg?color=red&size=large`

Bei Verteilungen können Sie wählen, ob Sie Abfragezeichenfolgen CloudFront an Ihren Ursprung weiterleiten möchten und ob Ihre Inhalte auf der Grundlage aller Parameter oder anhand ausgewählter Parameter zwischengespeichert werden sollen. Warum kann dies sinnvoll sein? Betrachten Sie das folgende Beispiel.

Angenommen, Ihre Website ist in fünf Sprachen verfügbar. Die Verzeichnisstruktur und Dateinamen für alle fünf Versionen der Website sind identisch. Wenn ein Benutzer Ihre Website besucht, werden Anfragen weitergeleitet, um einen Zeichenfolgenparameter für die Sprachabfrage zu CloudFront enthalten, der auf der vom Benutzer ausgewählten Sprache basiert. Sie können so konfigurieren CloudFront , dass Abfragezeichenfolgen an den Ursprung weitergeleitet und basierend auf dem Sprachparameter zwischengespeichert werden. Wenn Sie Ihren Webserver so konfigurieren, dass er die Version einer bestimmten Seite zurückgibt, die der ausgewählten Sprache entspricht, CloudFront speichert er jede Sprachversion separat im Cache, basierend auf dem Wert des Parameters für die Sprachabfragezeichenfolge.

Wenn in diesem Beispiel die Hauptseite Ihrer Website lautet`main.html`, führen die folgenden fünf Anfragen CloudFront dazu, dass fünfmal zwischengespeichert `main.html` wird, einmal für jeden Wert des Zeichenfolgenparameters für die Sprachabfrage:

- `https://d111111abcdef8.cloudfront.net/main.html?language=de`
- `https://d111111abcdef8.cloudfront.net/main.html?language=en`
- `https://d111111abcdef8.cloudfront.net/main.html?language=es`
- `https://d111111abcdef8.cloudfront.net/main.html?language=fr`
- `https://d111111abcdef8.cloudfront.net/main.html?language=jp`

Beachten Sie Folgendes:

- Einige HTTP-Server verarbeiten keine Abfragezeichenfolgeparameter und geben daher nicht unterschiedliche Versionen eines Objekts auf der Grundlage der Parameterwerte zurück. Wenn Sie für diese Ursprünge so konfigurieren, CloudFront dass Abfragezeichenfolgenparameter an den Ursprung weitergeleitet werden, wird der Cache CloudFront trotzdem auf der Grundlage der Parameterwerte zwischengespeichert, obwohl der Ursprung CloudFront für jeden Parameterwert identische Versionen des Objekts zurückgibt.
- Damit Abfragezeichenfolgeparameter wie im obigen Beispiel beschrieben mit den Sprachen funktionieren, müssen Sie das Zeichen & als Trennzeichen zwischen den

Abfragezeichenfolgeparametern verwenden. Wenn Sie ein anderes Trennzeichen verwenden, erhalten Sie möglicherweise unerwartete Ergebnisse, je nachdem, welche Parameter Sie als Grundlage für CloudFront das Caching angeben und in welcher Reihenfolge die Parameter in der Abfragezeichenfolge erscheinen.

Die folgenden Beispiele zeigen, was passiert, wenn Sie ein anderes Trennzeichen verwenden und den Cache nur auf der Grundlage des CloudFront Parameters konfigurieren: `color`

- In der folgenden Anfrage wird Ihr Inhalt auf der Grundlage des `color` Parameterwerts CloudFront zwischengespeichert, der Wert wird jedoch wie folgt CloudFront interpretiert: *`red;size=large`*

  `https://d111111abcdef8.cloudfront.net/images/image.jpg?`*`color=red;size=large`*
- In der folgenden Anfrage wird Ihr Inhalt CloudFront zwischengespeichert, das Caching basiert jedoch nicht auf den Parametern der Abfragezeichenfolge. Dies liegt daran, dass Sie das Zwischenspeichern CloudFront auf der Grundlage des `color` Parameters konfiguriert haben, die folgende Zeichenfolge jedoch so CloudFront interpretiert, dass sie nur einen `size` Parameter mit einem Wert von enthält: *`large;color=red`*

  `https://d111111abcdef8.cloudfront.net/images/image.jpg?`*`size=large;color=red`*

Sie können eine der folgenden Optionen konfigurieren CloudFront :

- Keine Abfragezeichenfolge an den Ursprung weiterleiten. Wenn Sie keine Abfragezeichenfolgen weiterleiten, wird der Cache CloudFront nicht auf der Grundlage von Abfragezeichenfolgenparametern zwischengespeichert.
- Abfragezeichenfolgen an den Ursprung weiterleiten und auf der Grundlage aller Parameter in der Abfragezeichenfolge zwischenspeichern.
- Abfragezeichenfolgen an den Ursprung weiterleiten und auf der Grundlage spezifischer Parameter in der Abfragezeichenfolge zwischenspeichern.

Weitere Informationen finden Sie unter the section called “Optimieren der Zwischenspeicherung”.

Themen

- Konsolen- und API-Einstellungen für die Weiterleitung und Zwischenspeicherung von Abfragezeichenfolgen
- Optimieren der Zwischenspeicherung
- Parameter für Abfragezeichenfolgen und CloudFront Standardprotokolle (Zugriffsprotokolle)

Konsolen- und API-Einstellungen für die Weiterleitung und Zwischenspeicherung von Abfragezeichenfolgen

Wenn Sie eine Verteilung in der CloudFront Konsole erstellen, CloudFront konfiguriert die Weiterleitung und das Zwischenspeichern von Abfragezeichenfolgen für Sie basierend auf Ihrem Ursprungstyp. Sie können diese Einstellungen auch manuell bearbeiten. Weitere Informationen zu den folgenden Einstellungen finden Sie in der the section called “Alle Distributionseinstellungen”:

- the section called “Weiterleitung und Zwischenspeicherung von Abfragezeichenfolgen”
- the section called “Zulassungsliste für Abfragezeichenfolgen”

Informationen zur Konfiguration der Weiterleitung und des Caches von Abfragezeichenfolgen mit der CloudFront API finden Sie unter CachePolicy und OriginRequestPolicy in der CloudFront Amazon-API-Referenz.

Optimieren der Zwischenspeicherung

Wenn Sie das Zwischenspeichern CloudFront auf der Grundlage von Abfragezeichenfolgenparametern konfigurieren, können Sie die folgenden Schritte ausführen, um die Anzahl der Anfragen zu reduzieren, die an CloudFront Ihren Ursprung weitergeleitet werden. Wenn CloudFront Edge-Standorte Objekte bereitstellen, reduzieren Sie die Last auf Ihrem Originalserver und reduzieren die Latenz, da Objekte von Standorten aus bedient werden, die näher an Ihren Benutzern liegen.

Zwischenspeichern auf der ausschließlichen Grundlage von Parametern, für die Ihr Ursprung verschiedene Versionen eines Objekts zurückgibt

> Für jeden Abfragezeichenfolgenparameter, an den Ihre Webanwendung CloudFront weiterleitetCloudFront, werden für jeden Parameterwert Anfragen an Ihren Ursprung weitergeleitet und für jeden Parameterwert eine separate Version des Objekts zwischengespeichert. Dies gilt auch, wenn Ihr Ursprung unabhängig vom Parameterwert immer dasselbe Objekt zurückgibt. Bei mehreren Parametern multiplizieren sich die Anzahl der Anfragen und die Anzahl der Objekte.

Wir empfehlen, die Konfiguration so CloudFront zu konfigurieren, dass das Caching nur auf der Grundlage der Abfragezeichenfolgenparameter erfolgt, für die Ihr Ursprung unterschiedliche Versionen zurückgibt, und dass Sie die Vorzüge des Cachings auf der Grundlage der einzelnen Parameter sorgfältig abwägen. Nehmen wir beispielsweise an, dass Sie eine Einzelhandels-Website besitzen. Sie haben Bilder einer Jacke in sechs verschiedenen Farben und es gibt die Jacke in zehn verschiedenen Größen. Die Bilder, die Sie von der Jacke haben, zeigen die verschiedenen Farben, jedoch nicht die verschiedenen Größen. Um das Caching zu optimieren, sollten Sie so konfigurieren, CloudFront dass das Caching nur auf der Grundlage des Farbparameters und nicht auf dem Größenparameter basiert. Dadurch steigt die Wahrscheinlichkeit, dass eine Anfrage aus dem Cache bearbeitet werden CloudFront kann, was die Leistung verbessert und die Belastung Ihres Ursprungscomputers verringert.

Parameter immer in der gleichen Reihenfolge auflisten

Die Reihenfolge der Parameter in Abfragezeichenfolgen ist wichtig. Im folgenden Beispiel sind die Abfragezeichenfolgen identisch, abgesehen davon, dass die Parameter eine andere Reihenfolge aufweisen. Dies führt CloudFront dazu, dass zwei separate Anfragen für image.jpg an Ihren Ursprung weitergeleitet werden und zwei separate Versionen des Objekts zwischengespeichert werden:

- `https://d111111abcdef8.cloudfront.net/images/image.jpg?color=red&size=large`
- `https://d111111abcdef8.cloudfront.net/images/image.jpg?size=large&color=red`

Wir empfehlen, Parameternamen immer in der gleichen Reihenfolge aufzulisten, beispielsweise in alphabetischer Reihenfolge.

Immer dieselbe Schreibweise für Parameternamen und -werte verwenden

CloudFront berücksichtigt die Groß- und Kleinschreibung von Parameternamen und -werten beim Zwischenspeichern auf der Grundlage von Abfragezeichenfolgenparametern. Im folgenden Beispiel sind die Abfragezeichenfolgen identisch, abgesehen von der Schreibweise der Parameternamen und -werte. Dies führt CloudFront dazu, dass vier separate Anfragen für image.jpg an Ihren Ursprung weitergeleitet und vier separate Versionen des Objekts zwischengespeichert werden:

- `https://d111111abcdef8.cloudfront.net/images/image.jpg?color=red`
- `https://d111111abcdef8.cloudfront.net/images/image.jpg?color=Red`
- `https://d111111abcdef8.cloudfront.net/images/image.jpg?Color=red`

- `https://d111111abcdef8.cloudfront.net/images/image.jpg?Color=Red`

  Wir empfehlen, konsistent dieselbe Schreibweise für Parameternamen und -werte zu verwenden, beispielsweise nur Kleinbuchstaben.

Keine Parameternamen verwenden, die mit signierten URLs in Konflikt stehen

Wenn Sie signierte URLs verwenden, um den Zugriff auf Ihre Inhalte einzuschränken (wenn Sie Ihrer Distribution vertrauenswürdige Unterzeichner hinzugefügt haben), werden die folgenden Abfragezeichenfolgenparameter CloudFront entfernt, bevor der Rest der URL an Ihren Ursprung weitergeleitet wird:

- `Expires`
- `Key-Pair-Id`
- `Policy`
- `Signature`

Wenn Sie signierte URLs verwenden und so konfigurieren möchten, dass Abfragezeichenfolgen CloudFront an Ihren Ursprung weitergeleitet werden, können Ihre eigenen Abfragezeichenfolgenparameter nicht mit`Expires`, `Key-Pair-Id`Policy, oder `Signature` benannt werden.

Parameter für Abfragezeichenfolgen und CloudFront Standardprotokolle (Zugriffsprotokolle)

Wenn Sie die Protokollierung aktivieren, wird die vollständige URL CloudFront protokolliert, einschließlich der Parameter für die Abfragezeichenfolge. Dies gilt unabhängig davon, ob Sie die Weiterleitung von Abfragezeichenfolgen an den Ursprung konfiguriert CloudFront haben. Weitere Hinweise zur CloudFront Protokollierung finden Sie unterthe section called “Zugriffsprotokolle (Standardprotokolle)”.

Zwischenspeichern von Inhalten auf der Grundlage von Cookies

Berücksichtigt standardmäßig CloudFront keine Cookies bei der Verarbeitung von Anfragen und Antworten oder beim Zwischenspeichern Ihrer Objekte an Edge-Speicherorten. Wenn zwei Anfragen CloudFront empfangen werden, die bis auf den Inhalt des `Cookie` Headers identisch sind, CloudFront behandelt es die Anfragen standardmäßig als identisch und gibt für beide Anfragen dasselbe Objekt zurück.

Sie können so konfigurieren CloudFront , dass einige oder alle Cookies in Viewer-Anfragen an Ihren Ursprung weitergeleitet werden und dass separate Versionen Ihrer Objekte auf der Grundlage der weitergeleiteten Cookie-Werte zwischengespeichert werden. Wenn Sie dies tun, CloudFront verwendet es einige oder alle Cookies in Viewer-Anfragen — unabhängig davon, welche Cookies für die Weiterleitung konfiguriert sind —, um ein Objekt im Cache eindeutig zu identifizieren.

Nehmen wir beispielsweise an, dass Anfragen für `locations.html` ein `country`-Cookie enthalten, das entweder den Wert `uk` oder `fr` hat. Wenn Sie so konfigurieren CloudFront , dass Ihre Objekte basierend auf dem Wert des Cookies zwischengespeichert werden, CloudFront leitet es Anfragen an den Ursprung weiter und schließt das `country` Cookie und seinen Wert ein. `locations.html` `country` Ihr Ursprung kehrt zurück und CloudFront speichert das Objekt einmal für Anfragen`locations.html`, in denen der Wert des `country` Cookies enthalten ist, `uk` und einmal für Anfragen, in denen der Wert ist. `fr`

> **Important**
>
> Amazon S3 und einige HTTP-Server verarbeiten keine Cookies. Konfiguriere nicht so CloudFront , dass Cookies an einen Ursprung weitergeleitet werden, der keine Cookies verarbeitet oder seine Antwort nicht anhand von Cookies variiert. Das kann CloudFront dazu führen, dass mehr Anfragen für dasselbe Objekt an den Ursprung weitergeleitet werden, was die Leistung verlangsamt und die Belastung des Ursprungs erhöht. Wenn in Anbetracht des vorherigen Beispiels Ihr Ursprung das `country` Cookie nicht verarbeitet oder CloudFront unabhängig vom Wert des Cookies immer dieselbe Version von `locations.html` an zurückgibt, konfigurieren Sie es nicht so, dass dieses Cookie CloudFront weitergeleitet wird. `country`
> Umgekehrt gilt: Wenn Ihr benutzerdefinierter Ursprung von einem bestimmten Cookie abhängt oder basierend auf einem Cookie unterschiedliche Antworten sendet, stellen Sie sicher, dass Sie das Cookie so konfigurieren CloudFront , dass es an den Ursprung weitergeleitet wird. Andernfalls CloudFront entfernt das Cookie, bevor die Anfrage an Ihren Ursprung weitergeleitet wird.

Zum Konfigurieren der Cookie-Weiterleitung aktualisieren Sie das Cache-Verhalten Ihrer Verteilung. Weitere Informationen über Cache-Verhalten finden Sie unter Einstellungen für das Cache-Verhalten, insbesondere in den Abschnitten Cookies weiterleiten und Zulassungslisten-Cookies.

Sie können jedes Cache-Verhalten so konfigurieren, dass eine der folgenden Aktionen ausgeführt wird:

- Leiten Sie alle Cookies an Ihren Ursprung weiter — CloudFront einschließlich aller Cookies, die der Betrachter bei der Weiterleitung von Anfragen an den Ursprung sendet. Wenn Ihr Absender eine Antwort zurückgibt, CloudFront speichert er die Antwort unter Verwendung der Cookie-Namen und -Werte in der Viewer-Anfrage im Cache. Wenn die ursprüngliche Antwort `Set-Cookie` Header enthält, werden diese zusammen mit dem angeforderten Objekt an den Viewer CloudFront zurückgegeben. CloudFront speichert auch die `Set-Cookie` Header mit dem vom Ursprung zurückgegebenen Objekt im Cache und sendet diese `Set-Cookie` Header bei allen Cache-Treffern an die Viewer.
- Eine Gruppe von Cookies weiterleiten, die Sie angeben — CloudFront entfernt alle Cookies, die der Betrachter sendet und die nicht auf der Zulassungsliste stehen, bevor eine Anfrage an den Ursprung weitergeleitet wird. CloudFront speichert die Antwort unter Verwendung der in der Viewer-Anfrage aufgeführten Cookie-Namen und -Werte im Cache. Wenn die ursprüngliche Antwort `Set-Cookie` Header enthält, werden diese zusammen mit dem angeforderten Objekt an den Viewer CloudFront zurückgegeben. CloudFront speichert auch die `Set-Cookie` Header mit dem vom Ursprung zurückgegebenen Objekt im Cache und sendet diese `Set-Cookie` Header bei allen Cache-Treffern an die Viewer.

  Weitere Informationen zum Angeben von Platzhaltern in Cookie-Namen finden Sie unter Zulassungslisten-Cookies.

  Informationen zum aktuellen Kontingent für die Anzahl von Cookie-Namen, die Sie für jedes Cache-Verhalten weiterleiten können, oder zum Anfordern eines höheren Kontingents finden Sie unter Kontingente für Abfragezeichenfolgen (Legacy-Cache-Einstellungen).
- Leiten Sie keine Cookies an Ihren Ursprung weiter — Ihre Objekte werden CloudFront nicht auf der Grundlage von Cookies zwischengespeichert, die vom Betrachter gesendet wurden. CloudFront Löscht außerdem Cookies, bevor Anfragen an deinen Absender weitergeleitet werden, und entfernt `Set-Cookie` Header aus Antworten, bevor Antworten an deine Zuschauer zurückgegeben werden. Da Ihre Ursprungsressourcen auf diese Weise nicht optimal genutzt werden, sollten Sie bei der Auswahl dieses Cacheverhaltens sicherstellen, dass Ihr Ursprung standardmäßig keine Cookies in Ursprungsantworten enthält.

Beachten Sie die folgenden Informationen zur Angabe des Cookies, das Sie weiterleiten möchten:

Zugriffsprotokolle

Wenn Sie so konfigurieren, CloudFront dass Anfragen protokolliert und Cookies CloudFront protokolliert werden, werden alle Cookies und alle Cookie-Attribute protokolliert, auch wenn

Sie so konfigurieren, dass Cookies CloudFront nicht an Ihren Ursprung weitergeleitet werden, oder wenn Sie so konfigurieren CloudFront , dass nur bestimmte Cookies weitergeleitet werden. Weitere Informationen zur CloudFront Protokollierung finden Sie unterZugriffsprotokolle (Standardprotokolle).

Groß-/Kleinschreibung

Bei Cookie-Namen und -Werten muss die Groß-/Kleinschreibung beachtet werden. Wenn beispielsweise so konfiguriert CloudFront ist, dass alle Cookies weitergeleitet werden und zwei Viewer-Anfragen für dasselbe Objekt Cookies enthalten, die bis auf die Groß- und Kleinschreibung identisch sind, wird das Objekt zweimal CloudFront zwischengespeichert.

CloudFront sortiert Cookies

Wenn so konfiguriert CloudFront ist, dass Cookies (alle oder eine Teilmenge) weitergeleitet werden, CloudFront sortiert die Cookies in natürlicher Reihenfolge nach dem Cookie-Namen, bevor die Anfrage an Ihren Ursprung weitergeleitet wird.

> **Note**
>
> Cookie-Namen, die mit dem $ Zeichen beginnen, werden nicht unterstützt. CloudFront entfernt das Cookie, bevor die Anfrage an den Ursprung weitergeleitet wird. Sie können das Zeichen $ entfernen oder ein anderes Zeichen am Anfang des Cookie-Namens angeben.

`If-Modified-Since` und **`If-None-Match`**

`If-Modified-Since`und `If-None-Match` bedingte Anfragen werden nicht unterstützt, wenn so konfiguriert CloudFront ist, dass Cookies weitergeleitet werden (alle oder eine Teilmenge).

Standard-Name-Wert-Paar-Format erforderlich

CloudFront leitet einen Cookie-Header nur weiter, wenn der Wert dem Standardformat für Name-Wert-Paare entspricht, zum Beispiel: `"Cookie: cookie1=value1; cookie2=value2"`

Deaktivieren der Zwischenspeicherung von **`Set-Cookie`**-Headern

Wenn CloudFront es so konfiguriert ist, dass Cookies an den Ursprung weitergeleitet werden (ob alle oder bestimmte Cookies), werden auch die in der Quellantwort empfangenen `Set-Cookie` Header zwischengespeichert. CloudFront bezieht diese `Set-Cookie` Header in die Antwort an

den ursprünglichen Viewer ein und schließt sie auch in nachfolgende Antworten ein, die aus dem Cache bereitgestellt werden. CloudFront

Wenn Sie Cookies an Ihrem Ursprung erhalten möchten, aber die `Set-Cookie` Header in den Antworten Ihres Absenders nicht zwischenspeichern CloudFront möchten, konfigurieren Sie Ihren Ursprung so, dass ein `Cache-Control` Header mit einer `no-cache` Anweisung hinzugefügt wird, die `Set-Cookie` als Feldname angegeben wird. Beispiel: `Cache-Control: no-cache="Set-Cookie"`. Weitere Informationen finden Sie unter Cache-Control Antwortanweisungen im Hypertext Transfer Protocol (HTTP/1.1): Caching-Standard.

Maximallänge von Cookie-Namen

Wenn Sie so konfigurieren CloudFront , dass bestimmte Cookies an Ihren Ursprung weitergeleitet werden, darf die Gesamtzahl der Byte in allen Cookie-Namen, die Sie für die Weiterleitung konfigurieren CloudFront , 512 abzüglich der Anzahl der Cookies, die Sie weiterleiten, nicht überschreiten. Wenn Sie beispielsweise so konfigurieren, CloudFront dass 10 Cookies an Ihren Ursprung weitergeleitet werden, darf die kombinierte Länge der Namen der 10 Cookies 502 Byte (512 — 10) nicht überschreiten.

Wenn Sie so konfigurieren CloudFront , dass alle Cookies an Ihren Ursprung weitergeleitet werden, spielt die Länge der Cookie-Namen keine Rolle.

Informationen zur Verwendung der CloudFront Konsole zum Aktualisieren einer Distribution, sodass Cookies an den Ursprung CloudFront weitergeleitet werden, finden Sie unterEine Verteilung aktualisieren. Informationen zur Verwendung der CloudFront API zur Aktualisierung einer Distribution finden Sie UpdateDistribution in der CloudFront Amazon-API-Referenz.

Zwischenspeichern von Inhalten auf der Grundlage von Anforderungsheadern

Mit können Sie wählen CloudFront, ob Sie Header CloudFront an Ihren Ursprung weiterleiten und separate Versionen eines bestimmten Objekts auf der Grundlage der Header-Werte in Viewer-Anfragen zwischenspeichern möchten. Auf diese Weise können Sie verschiedene Versionen Ihrer Inhalte bereitstellen, basierend auf dem Gerät, das der Benutzer verwendet, dem Standort des Betrachters, der Sprache, die der Zuschauer verwendet, und einer Vielzahl anderer Kriterien.

Themen

- Header und Verteilungen – Übersicht

- Auswählen der Header, auf denen die Zwischenspeicherung basieren soll
- Konfigurieren Sie so CloudFront , dass die CORS-Einstellungen berücksichtigt werden
- Konfigurieren der Zwischenspeicherung basierend auf dem Gerätetyp
- Konfigurieren der Zwischenspeicherung basierend auf der Sprache des Viewers
- Konfigurieren der Zwischenspeicherung basierend auf dem Standort des Viewers
- Konfigurieren der Zwischenspeicherung basierend auf dem Protokoll der Anforderung
- Konfigurieren der Zwischenspeicherung für komprimierte Dateien
- Auswirkungen der Zwischenspeicherung auf der Grundlage von Headern auf die Leistung
- Auswirkungen der Schreibweise von Headern und Header-Werten auf die Zwischenspeicherung
- Header, die zum CloudFront Viewer zurückkehren

Header und Verteilungen – Übersicht

Überschriften werden standardmäßig CloudFront nicht berücksichtigt, wenn Sie Ihre Objekte an Randpositionen zwischenspeichern. Wenn Ihr Ursprung zwei Objekte zurückgibt und sie sich nur durch die Werte in den Anforderungsheadern unterscheiden, wird nur eine Version des Objekts CloudFront zwischengespeichert.

Sie können so konfigurieren CloudFront , dass Header an den Ursprung weitergeleitet werden, was CloudFront dazu führt, dass mehrere Versionen eines Objekts auf der Grundlage der Werte in einem oder mehreren Anforderungsheadern zwischengespeichert werden. Um CloudFront zu konfigurieren, dass Objekte auf der Grundlage der Werte bestimmter Header zwischengespeichert werden, geben Sie Einstellungen für das Cache-Verhalten für Ihre Distribution an. Weitere Informationen finden Sie unter Cache-Speicherung basierend auf ausgewählten Anforderungs-Headern.

Nehmen wir beispielsweise an, dass Viewer-Anforderungen für `logo.jpg` einen benutzerdefinierten `Product`-Header enthalten, der entweder den Wert `Acme` oder `Apex` hat. Wenn Sie so konfigurieren CloudFront , dass Ihre Objekte basierend auf dem Wert des `Product` Headers zwischengespeichert werden, CloudFront leitet Anfragen `logo.jpg` an den Ursprung weiter und schließt den Header und die `Product` Header-Werte ein. CloudFront speichert `logo.jpg` einmal für Anfragen, in denen der Wert des `Product` Headers enthalten ist, `Acme` und einmal für Anfragen, in denen der Wert enthalten ist. `Apex`

Sie können jedes Cache-Verhalten in einer Verteilung so konfigurieren, dass eine der folgenden Aktionen ausgeführt wird:

- Weiterleiten aller Header an Ihren Ursprung

> **Note**
>
> Für ältere Cache-Einstellungen — Wenn Sie so konfigurieren CloudFront , dass alle Header an Ihren Ursprung weitergeleitet werden, werden die Objekte, die mit diesem Cache-Verhalten verknüpft sind, CloudFront nicht zwischengespeichert. Stattdessen werden alle Anfragen an den Ursprung gesendet.

- Leitet eine Liste von Headern weiter, die Sie angeben. CloudFront speichert Ihre Objekte auf der Grundlage der Werte in allen angegebenen Headern im Cache. CloudFront leitet auch die Header weiter, die es standardmäßig weiterleitet, aber Ihre Objekte werden nur auf der Grundlage der von Ihnen angegebenen Header zwischengespeichert.
- Weiterleiten nur der Standard-Header. In dieser Konfiguration werden Ihre Objekte CloudFront nicht auf der Grundlage der Werte in den Anforderungsheadern zwischengespeichert.

Informationen zum aktuellen Kontingent für die Anzahl von Headern, die Sie für jedes Cache-Verhalten weiterleiten können, oder zum Anfordern eines höheren Kontingents finden Sie unter Kontingente für Header.

Hinweise zur Verwendung der CloudFront Konsole zum Aktualisieren einer Distribution, sodass Header an den Ursprung CloudFront weitergeleitet werden, finden Sie unter. Eine Verteilung aktualisieren Informationen zur Verwendung der CloudFront API zur Aktualisierung einer vorhandenen Distribution finden Sie unter Update Distribution in der CloudFront Amazon-API-Referenz.

Auswählen der Header, auf denen die Zwischenspeicherung basieren soll

Die Header, die Sie an den Ursprung weiterleiten können und auf denen das Caching CloudFront basiert, hängen davon ab, ob es sich bei Ihrem Ursprung um einen Amazon S3-Bucket oder einen benutzerdefinierten Ursprung handelt.

- Amazon S3 — Sie können so konfigurieren CloudFront , dass Ihre Objekte auf der Grundlage einer Reihe bestimmter Header weitergeleitet und zwischengespeichert werden (siehe die folgende Liste von Ausnahmen). Allerdings empfehlen wir, möglichst keine Header mit einem Amazon-S3-Ursprung weiterzuleiten, es sei denn, Sie implementieren Cross-Origin Resource Sharing (CORS, ursprungsübergreifende gemeinsame Nutzung von Ressourcen) oder möchten Inhalte mit Lambda@Edge in Ereignissen auf Ursprungsseite personalisieren.

  - Um CORS zu konfigurieren, müssen Sie Header weiterleiten, die es ermöglichen, Inhalte für Websites CloudFront zu verteilen, die für Cross-Origin Resource Sharing (CORS) aktiviert sind. Weitere Informationen finden Sie unter Konfigurieren Sie so CloudFront , dass die CORS-Einstellungen berücksichtigt werden.
  - Um Inhalte mithilfe von Headern zu personalisieren, die Sie an Ihren Amazon S3-Ursprung weiterleiten, schreiben Sie Lambda @Edge -Funktionen, fügen diese hinzu und verknüpfen sie mit Ihrer CloudFront Verteilung, sodass sie durch ein Ereignis ausgelöst werden, das mit dem Ursprung verbunden ist. Weitere Informationen zum Verwenden von Headern, um Inhalte zu personalisieren, finden Sie unter Personalisieren von Inhalten nach Land oder Gerätetyp-Header – Beispiele.

    Wir empfehlen, möglichst keine Header weiterzuleiten, die Sie nicht zum Personalisieren von Inhalten verwenden, da das Weiterleiten zusätzlicher Header zu einer Verringerung der Cache-Trefferrate führen kann. Das heißt, es CloudFront können nicht so viele Anfragen von Edge-Caches bedient werden, wie der Anteil aller Anfragen.

- Benutzerdefinierter Ursprung — Sie können CloudFront den Cache auf der Grundlage des Werts eines beliebigen Anforderungsheaders konfigurieren, mit Ausnahme der folgenden:
  - `Connection`
  - `Cookie` – Wenn die Weiterleitung und Zwischenspeicherung auf der Grundlage von Cookies erfolgen soll, verwenden Sie eine separate Einstellung in Ihrer Verteilung. Weitere Informationen finden Sie unter Zwischenspeichern von Inhalten auf der Grundlage von Cookies.
  - `Host (for Amazon S3 origins)`
  - `Proxy-Authorization`
  - `TE`
  - `Upgrade`

  Sie können so konfigurieren CloudFront , dass Objekte auf der Grundlage von Werten in den `User-Agent` Headern `Date` und zwischengespeichert werden, wir empfehlen dies jedoch nicht. Diese Header haben zahlreiche mögliche Werte, und das Caching auf der Grundlage ihrer Werte könnte dazu führen, dass deutlich mehr Anfragen CloudFront an Ihren Ursprung weitergeleitet werden.

Eine vollständige Liste der HTTP-Anforderungsheader und deren Verarbeitung finden Sie CloudFront unter. Header und CloudFront Verhalten von HTTP-Anfragen (benutzerdefiniert und Amazon S3-Ursprünge)

Konfigurieren Sie so CloudFront , dass die CORS-Einstellungen berücksichtigt werden

Wenn Sie Cross-Origin Resource Sharing (CORS) auf einem Amazon S3-Bucket oder einem benutzerdefinierten Ursprung aktiviert haben, müssen Sie bestimmte Header zum Weiterleiten auswählen, um die CORS-Einstellungen zu berücksichtigen. Die Header, die Sie weiterleiten müssen, sind abhängig vom Ursprung (Amazon S3 oder benutzerdefiniert) und von der Tatsache, ob Sie `OPTIONS`-Antworten zwischenspeichern möchten, unterschiedlich.

Amazon S3

- Wenn Sie möchten, dass `OPTIONS`-Antworten im Cache gespeichert werden, führen Sie die folgenden Schritte aus:
  - Wählen Sie die Optionen für die Standardeinstellungen für das Cache-Verhalten aus, mit denen das Zwischenspeichern von `OPTIONS`-Antworten aktiviert wird.
  - Konfigurieren CloudFront Sie so, dass die folgenden Header weitergeleitet werden: `OriginAccess-Control-Request-Headers`, und. `Access-Control-Request-Method`
- Wenn Sie nicht möchten, dass `OPTIONS`-Antworten im Cache gespeichert werden, konfigurieren Sie CloudFront so, dass der `Origin`-Header zusammen mit allen Headern, die für Ihren Ursprung erforderlich sind, weitergeleitet wird (z. B. `Access-Control-Request-Headers`, `Access-Control-Request-Method` oder andere).

Benutzerdefinierte Ursprünge – Leiten Sie den `Origin`-Header zusammen mit anderen Headern weiter, die für Ihren Ursprung erforderlich sind.

Um CloudFront zu konfigurieren, dass Antworten auf der Grundlage von CORS zwischengespeichert werden, müssen Sie die Weiterleitung von Headern mithilfe einer Cache-Richtlinie konfigurieren CloudFront . Weitere Informationen finden Sie unter Steuern des Cache-Schlüssels mit einer Richtlinie.

Weitere Informationen zu CORS und Amazon S3 finden Sie unter Cross-Origin Resource Sharing (CORS) verwenden im Benutzerhandbuch zu Amazon Simple Storage Service.

Konfigurieren der Zwischenspeicherung basierend auf dem Gerätetyp

Wenn Sie je CloudFront nach Gerät, das ein Benutzer zum Anzeigen Ihrer Inhalte verwendet, verschiedene Versionen Ihrer Objekte zwischenspeichern möchten, konfigurieren Sie

die entsprechenden Header so, dass die entsprechenden Header CloudFront an Ihren benutzerdefinierten Ursprung weitergeleitet werden:

- `CloudFront-Is-Desktop-Viewer`
- `CloudFront-Is-Mobile-Viewer`
- `CloudFront-Is-SmartTV-Viewer`
- `CloudFront-Is-Tablet-Viewer`

CloudFront Legt basierend auf dem Wert des `User-Agent` Headers den Wert dieser Header auf `true` oder `false` vor der Weiterleitung der Anfrage an Ihren Ursprung fest. Wenn ein Gerät in mehr als eine Kategorie fällt, können mehrere Werte sei `true`. Bei einigen Tablet-Geräten CloudFront können Sie z. B. `CloudFront-Is-Mobile-Viewer` sowohl als auch `CloudFront-Is-Tablet-Viewer` auf `true` setzen.

Konfigurieren der Zwischenspeicherung basierend auf der Sprache des Viewers

Wenn Sie verschiedene Versionen Ihrer Objekte auf der Grundlage der in der Anfrage angegebenen Sprache zwischenspeichern möchten CloudFront , konfigurieren Sie so, CloudFront dass der `Accept-Language` Header an Ihren Ursprung weitergeleitet wird.

Konfigurieren der Zwischenspeicherung basierend auf dem Standort des Viewers

Wenn Sie je CloudFront nach Land, aus dem die Anfrage stammt, verschiedene Versionen Ihrer Objekte zwischenspeichern möchten, konfigurieren Sie CloudFront , dass der `CloudFront-Viewer-Country` Header an Ihren Ursprung weitergeleitet wird. CloudFront wandelt die IP-Adresse, von der die Anfrage kam, automatisch in eine aus zwei Buchstaben bestehende Landeskennzahl um. Eine benutzerfreundliche Liste der Ländercodes, die nach Code und Ländername sortiert werden kann, finden Sie im Wikipedia-Eintrag ISO 3166-1 alpha-2.

Konfigurieren der Zwischenspeicherung basierend auf dem Protokoll der Anforderung

Wenn Sie verschiedene Versionen Ihrer Objekte basierend auf dem Protokoll der Anfrage, HTTP oder HTTPS, zwischenspeichern möchten CloudFront , konfigurieren Sie, dass der `CloudFront-Forwarded-Proto` Header CloudFront an Ihren Ursprung weitergeleitet wird.

Konfigurieren der Zwischenspeicherung für komprimierte Dateien

Wenn Ihr Ursprung die Brotli-Komprimierung unterstützt, können Sie die Zwischenspeicherung basierend auf dem `Accept-Encoding`-Header durchführen. Konfigurieren Sie die Zwischenspeicherung nur dann basierend auf `Accept-Encoding`, wenn Ihr Ursprungs-Server abhängig vom Header unterschiedliche Inhalte bereitstellt.

Auswirkungen der Zwischenspeicherung auf der Grundlage von Headern auf die Leistung

Wenn Sie CloudFront den Cache auf der Grundlage von einem oder mehreren Headern konfigurieren und die Header mehr als einen möglichen Wert haben, werden mehrere Anfragen für dasselbe Objekt an Ihren Ursprungsserver CloudFront weitergeleitet. Dadurch wird die Leistung verringert und die Last auf Ihrem Ursprungs-Server erhöht. Wenn Ihr Ursprungsserver unabhängig vom Wert eines bestimmten Headers dasselbe Objekt zurückgibt, empfehlen wir Ihnen, das Cachen nicht auf der Grundlage dieses Headers CloudFront zu konfigurieren.

Wenn Sie so konfigurieren, CloudFront dass mehr als ein Header weitergeleitet wird, wirkt sich die Reihenfolge der Header in Viewer-Anfragen nicht auf das Caching aus, solange die Werte identisch sind. Wenn beispielsweise eine Anfrage die Header enthält B:2 und eine andere Anfrage A:1,, CloudFront speichert nur eine Kopie des Objekts im Cache. B:2 A:1

Auswirkungen der Schreibweise von Headern und Header-Werten auf die Zwischenspeicherung

Beim CloudFront Zwischenspeichern auf der Grundlage von Header-Werten wird die Groß- und Kleinschreibung des Header-Namens nicht berücksichtigt, wohl aber die Groß- und Kleinschreibung des Header-Werts:

- Wenn Viewer-Anfragen `Product:Acme` sowohl als auch enthaltenproduct:Acme, wird ein Objekt nur einmal CloudFront zwischengespeichert. Der einzige Unterschied zwischen ihnen ist die Schreibweise des Header-Namens, die sich nicht auf die Zwischenspeicherung auswirkt.
- Wenn Viewer-Anfragen `Product:Acme` sowohl als auch enthaltenProduct:acme, wird ein Objekt zweimal CloudFront zwischengespeichert, da der Wert `Acme` in einigen Anfragen und `acme` in anderen enthalten ist.

Header, die zum CloudFront Viewer zurückkehren

Die Konfiguration CloudFront für das Weiterleiten und Zwischenspeichern von Headern hat keinen Einfluss darauf, welche Header an den Viewer CloudFront zurückgegeben werden. CloudFront gibt alle Header zurück, die es vom Ursprung erhält, mit wenigen Ausnahmen. Weitere Informationen finden Sie im entsprechenden Thema:

- Amazon S3-Ursprünge – siehe HTTP-Antwort-Header, die CloudFront entfernt oder aktualisiert werden.
- Benutzerdefinierte Ursprünge – Siehe HTTP-Antwort-Header, die CloudFront entfernt oder ersetzt werden.

Steuern des Cache-Schlüssels mit einer Richtlinie

Mit einer CloudFront Cache-Richtlinie können Sie die HTTP-Header, Cookies und Abfragezeichenfolgen angeben, die im Cache-Schlüssel für Objekte CloudFront enthalten sind, die an CloudFront Edge-Standorten zwischengespeichert werden. Der Cache-Schlüssel ist der eindeutige Bezeichner für jedes Objekt im Cache und bestimmt, ob die HTTP-Anforderung eines Viewers zu einem Cache-Treffer führt.

Zu einem Cache-Treffer kommt es, wenn eine Viewer-Anforderung denselben Cache-Schlüssel wie eine vorherige Anforderung generiert und das Objekt für diesen Cache-Schlüssel im Cache des Edge-Standorts vorhanden und gültig ist. Bei einem Cache-Treffer wird das Objekt dem Betrachter von einem CloudFront Edge-Standort aus bereitgestellt, was die folgenden Vorteile bietet:

- Geringere Auslastung Ihres Ursprungs-Servers
- Reduzierte Latenz für den Viewer

Weniger Werte im Cache-Schlüssel erhöhen die Wahrscheinlichkeit eines Cache-Treffers. So können Sie eine bessere Leistung für Ihre Website oder Anwendung erzielen, da dies die Cache-Trefferquote erhöht (ein höherer Anteil an Viewer-Anforderungen, die zu einem Cache-Treffer führen). Weitere Informationen finden Sie unter Grundlegendes zum Cache-Schlüssel.

Um den Cache-Schlüssel zu steuern, verwenden Sie eine CloudFront Cache-Richtlinie. Sie fügen eine Cache-Richtlinie einem oder mehreren Cache-Verhalten in einer CloudFront Verteilung zu.

Sie können die Cache-Richtlinie auch verwenden, um TTL-Einstellungen (Time to Live) für Objekte im CloudFront Cache festzulegen und komprimierte Objekte CloudFront anzufordern und zwischenzuspeichern.

> **Note**
>
> Cache-Einstellungen haben keine Auswirkungen auf gRPC-Anforderungen, da gRPC-Datenverkehr nicht zwischengespeichert werden kann. Weitere Informationen finden Sie unter gRPC mit CloudFront Distributionen verwenden.

Themen

- Grundlegendes zu Cache-Richtlinien

- Erstellen von Cache-Richtlinien
- Verwenden verwalteter Cache-Richtlinien
- Grundlegendes zum Cache-Schlüssel

Grundlegendes zu Cache-Richtlinien

Sie können eine Cache-Richtlinie verwenden, um das Cache-Trefferverhältnis zu verbessern, indem Sie die Werte (URL-Abfragezeichenfolgen, HTTP-Header und Cookies) kontrollieren, die im Cache-Schlüssel enthalten sind. CloudFront enthält einige vordefinierte Cache-Richtlinien für häufige Anwendungsfälle, die als verwaltete Richtlinien bezeichnet werden. Sie können diese verwalteten Richtlinien verwenden oder eine eigene Cache-Richtlinie erstellen, die speziell auf Ihre Anforderungen zugeschnitten ist. Weitere Informationen zu verwalteten Richtlinien finden Sie unter Verwenden verwalteter Cache-Richtlinien.

Eine Cache-Richtlinie enthält die folgenden Einstellungen, die in Richtlinieninformationen, Einstellungen für Time to Live (TTL, Gültigkeitsdauer der Verbindung) und Cache-Schlüssel-Einstellungen unterteilt sind.

Richtlinieninformationen

Name

Ein Name zur Identifizierung der Cache-Richtlinie. Verwenden Sie den Namen in der Konsole, um die Cache-Richtlinie einem Cache-Verhalten anzufügen.

Beschreibung

Ein Kommentar zur Beschreibung der Cache-Richtlinie. Dies ist optional, kann Ihnen jedoch helfen, den Zweck der Cache-Richtlinie zu identifizieren.

Einstellungen für Time to Live (TTL, Gültigkeitsdauer der Verbindung)

Die Einstellungen für die Gültigkeitsdauer (TTL) bestimmen zusammen mit den Headern `Cache-Control` und den `Expires` HTTP-Headern (sofern sie in der ursprünglichen Antwort enthalten sind), wie lange Objekte im CloudFront Cache gültig bleiben.

Mindest-TTL

Die Mindestdauer in Sekunden, für die Objekte im CloudFront Cache verbleiben sollen, bevor beim Ursprung CloudFront überprüft wird, ob das Objekt aktualisiert wurde. Weitere Informationen finden Sie unter Verwalten der Dauer, die Inhalte im Cache bleiben (Ablauf).

> **Warning**
>
> Wenn Ihre Mindest-TTL größer als 0 ist, CloudFront wird Inhalt mindestens für die in der Mindest-TTL der Cache-Richtlinie angegebene Dauer zwischengespeichert, auch wenn die `private` Direktiven `Cache-Control: no-cacheno-store`, oder in den Origin-Headern vorhanden sind.

Höchst-TTL

Die maximale Zeit in Sekunden, für die Objekte im CloudFront Cache verbleiben, bevor beim Ursprung CloudFront überprüft wird, ob das Objekt aktualisiert wurde. CloudFront verwendet diese Einstellung nur, wenn der Ursprung das Objekt sendet `Cache-Control` oder den `Expires` Header mit dem Objekt teilt. Weitere Informationen finden Sie unter Verwalten der Dauer, die Inhalte im Cache bleiben (Ablauf).

Standard-TTL

Die Standarddauer in Sekunden, für die Objekte im CloudFront Cache bleiben sollen, bevor beim Ursprung CloudFront überprüft wird, ob das Objekt aktualisiert wurde. CloudFront verwendet den Wert dieser Einstellung nur dann als TTL des Objekts, wenn der Ursprung keine `Expires` Kopfzeilen mit dem Objekt sendet`Cache-Control`. Weitere Informationen finden Sie unter Verwalten der Dauer, die Inhalte im Cache bleiben (Ablauf).

> **Note**
>
> Wenn die Einstellungen Minimale TTL, Maximale TTL und Standard-TTL alle auf 0 gesetzt sind, wird das Caching deaktiviert. CloudFront

Cache-Schlüssel-Einstellungen

Die Cache-Schlüsseleinstellungen geben die Werte in Viewer-Anfragen an, die im Cache-Schlüssel CloudFront enthalten sind. Bei den Werten kann es sich um URL-Abfragezeichenfolgen, HTTP-Header und Cookies handeln. Die Werte, die Sie in den Cache-Schlüssel einschließen, werden automatisch in Anforderungen aufgenommen, die CloudFront an den Ursprung sendet. Diese werden als Ursprungsanforderungen bezeichnet. Informationen zum Kontrollieren von Ursprungsanforderungen ohne Auswirkungen auf den Cache-Schlüssel finden Sie unter Steuern von Ursprungsanforderungen anhand einer Richtlinie.

Zu den Cache-Schlüssel-Einstellungen gehören:

- Header
- Cookies
- Abfragezeichenfolgen
- Komprimierungsunterstützung

Header

Die HTTP-Header in Viewer-Anfragen, die im Cache-Schlüssel CloudFront enthalten sind, und in Ursprungsanfragen. Sie können für Header eine der folgenden Einstellungen auswählen:

- None (Keine) – Die HTTP-Header in Viewer-Anforderungen sind nicht im Cache-Schlüssel enthalten und werden nicht automatisch in Ursprungsanforderungen eingeschlossen.
- Folgende Header einschließen – Sie geben an, welche der HTTP-Header in Viewer-Anforderungen im Cache-Schlüssel enthalten sind und automatisch in Ursprungsanforderungen eingeschlossen werden.

Wenn Sie die Einstellung Folgende Header einschließen verwenden, geben Sie HTTP-Header nach ihrem Namen und nicht nach ihrem Wert an. Betrachten Sie beispielsweise den folgenden HTTP-Header:

```
Accept-Language: en-US,en;q=0.5
```

In diesem Fall geben Sie den Header als `Accept-Language` an, nicht als `Accept-Language: en-US,en;q=0.5` an. CloudFront enthält jedoch den vollständigen Header, einschließlich seines Wertes, im Cache-Schlüssel und in Ursprungsanforderungen.

Sie können auch bestimmte Header, die von generiert wurden, CloudFront in den Cache-Schlüssel aufnehmen. Weitere Informationen finden Sie unter the section called “CloudFront Anforderungsheader hinzufügen”.

Cookies

Die Cookies in Viewer-Anfragen, die im Cache-Schlüssel CloudFront enthalten sind, und in Ursprungsanfragen. Für Cookies können Sie eine der folgenden Einstellungen auswählen:

- None (Keine) – Die Cookies in Viewer-Anforderungen sind nicht im Cache-Schlüssel enthalten und werden nicht automatisch in Ursprungsanforderungen aufgenommen.
- All (Alle) – Alle Cookies in Viewer-Anforderungen sind im Cache-Schlüssel enthalten und werden automatisch in Ursprungsanforderungen aufgenommen.
- Angegebene Cookies einschließen – Sie geben an, welche der Cookies in Viewer-Anforderungen im Cache-Schlüssel enthalten sind und automatisch in Ursprungsanforderungen eingeschlossen werden.
- Alle Cookies einschließen außer – Sie geben an, welche der Cookies in Viewer-Anforderungen nicht im Cache-Schlüssel enthalten sind und nicht automatisch in Ursprungsanforderungen enthalten sind. Alle anderen Cookies, mit Ausnahme der von ihnen angegebenen, sind im Cache-Schlüssel und automatisch in Ursprungsanforderungen enthalten.

Wenn Sie die Einstellung Angegebene Cookies einschließen oder Alle Cookies einschließen außer verwenden verwenden, geben Sie Cookies nach ihrem Namen und nicht nach ihrem Wert an. Betrachten Sie beispielsweise den folgenden `Cookie`-Header:

```
Cookie: session_ID=abcd1234
```

In diesem Fall geben Sie das Cookie als `session_ID`, nicht als `session_ID=abcd1234` an. CloudFront Schließt jedoch das vollständige Cookie, einschließlich seines Werts, in den Cache-Schlüssel und in Ursprungsanfragen ein.

Abfragezeichenfolgen

Die URL-Abfragezeichenfolgen in Viewer-Anfragen, die im Cache-Schlüssel CloudFront enthalten sind, und in ursprünglichen Anfragen. Für Abfragezeichenfolgen können Sie eine der folgenden Einstellungen auswählen:

- None (Keine) – Abfragezeichenfolgen in Viewer-Anforderungen sind nicht im Cache-Schlüssel enthalten und sind nicht automatisch in Ursprungsanforderungen enthalten.
- All (Alle) – Alle Abfragezeichenfolgen in Viewer-Anforderungen sind im Cache-Schlüssel enthalten und sind auch automatisch in Ursprungsanforderungen enthalten.

- Angegebene Abfragezeichenfolgen einschließne – Sie geben an, welche der Abfragezeichenfolgen in Viewer-Anforderungen im Cache-Schlüssel enthalten sind und automatisch in Ursprungsanforderungen eingeschlossen werden.
- Alle Abfragezeichenfolgen einschließen außer – Sie geben an, welche der Abfragezeichenfolgen in Viewer-Anforderungen nicht im Cache-Schlüssel enthalten sind und nicht automatisch in Ursprungsanforderungen eingeschlossen werden. Alle anderen Abfragezeichenfolgen außer denen, die Sie angegeben haben, sind im Cache-Schlüssel und automatisch in Ursprungsanforderungen enthalten.

Wenn Sie die Einstellung Angegebene Abfragezeichenfolgen einschließen oder Alle Abfragezeichenfolgen einschließen außer verwenden, geben Sie Abfragezeichenfolgen nach ihrem Namen und nicht nach ihrem Wert an. Betrachten Sie beispielsweise den folgenden URL-Pfad:

```
/content/stories/example-story.html?split-pages=false
```

In diesem Fall geben Sie die Abfragezeichenfolge als `split-pages`, nicht als `split-pages=false` an. CloudFront Schließt jedoch die vollständige Abfragezeichenfolge, einschließlich ihres Werts, in den Cache-Schlüssel und in Ursprungsanfragen ein.

> **Note**
>
> CloudFront Behandelt bei den Einstellungen für den Cacheschlüssel das Sternchen (*) für die Header, Abfragezeichenfolgen und Cookies als wörtliche Zeichenfolge und nicht als Platzhalter.

Komprimierungsunterstützung

Diese Einstellungen ermöglichen CloudFront das Anfordern und Zwischenspeichern von Objekten, die im Gzip- oder Brotli-Komprimierungsformat komprimiert sind, sofern der Viewer dies unterstützt. Diese Einstellungen ermöglichen auch die Funktion der CloudFront -Komprimierung . Viewer geben ihre Unterstützung für diese Komprimierungsformate mit dem `Accept-Encoding`-HTTP-Header an.

> **Note**
>
> Die Webbrowser Chrome und Firefox unterstützen die Brotli-Komprimierung nur, wenn die Anforderung über HTTPS gesendet wird. Diese Browser unterstützen Brotli mit HTTP-Anforderungen nicht.

Aktivieren Sie diese Einstellungen, wenn eine der folgenden Bedingungen zutrifft:

- Ihr Ursprung gibt mit Gzip komprimierte Objekte zurück, wenn Viewer diese unterstützen (Anforderungen enthalten den `Accept-Encoding`-HTTP-Header mit `gzip` als Wert). Verwenden Sie in diesem Fall die Gzip-Einstellung aktiviert (`true`in der CloudFront API `EnableAcceptEncodingGzip` auf, AWS SDKs, AWS CLI oder gesetzt). CloudFormation
- Ihr Ursprung gibt mit Brotli komprimierte Objekte zurück, wenn Viewer diese unterstützen (Anforderungen enthalten den `Accept-Encoding`-HTTP-Header mit `br` als Wert). Verwenden Sie in diesem Fall die Einstellung Brotli enabled (`true`in der CloudFront API `EnableAcceptEncodingBrotli` auf, AWS SDKs AWS CLI, oder gesetzt). CloudFormation
- Die Cache-Verhaltensweise, der diese Cache-Richtlinie angefügt ist, ist mit CloudFront-Komprimierung konfiguriert. In diesem Fall können Sie die Zwischenspeicherung für Gzip oder Brotli oder beides aktivieren. Wenn die CloudFront Komprimierung aktiviert ist, kann die Aktivierung des Zwischenspeichers für beide Formate dazu beitragen, Ihre Kosten für die Datenübertragung ins Internet zu senken.

> **Note**
>
> Wenn Sie das Caching für eines oder beide dieser Komprimierungsformate aktivieren, sollten Sie den `Accept-Encoding` Header nicht in eine Quellanforderungsrichtlinie aufnehmen, die demselben Cache-Verhalten zugeordnet ist. CloudFront bezieht diesen Header immer in ursprüngliche Anfragen ein, wenn das Caching für eines dieser Formate aktiviert ist, sodass die Aufnahme `Accept-Encoding` in eine Richtlinie für ursprüngliche Anfragen keine Auswirkung hat.

Wenn Ihr Ursprungsserver keine mit Gzip oder Brotli komprimierten Objekte zurückgibt oder das Cache-Verhalten nicht mit CloudFront Komprimierung konfiguriert ist, aktivieren Sie das Caching für komprimierte Objekte nicht. Wenn Sie dies dennoch tun, kann dies zu einer Verringerung der Cache-Trefferquote führen.

Im Folgenden wird erklärt, wie sich diese Einstellungen auf eine Verteilung auswirken. CloudFront In allen folgenden Szenarien wird davon ausgegangen, dass die Viewer-Anforderung den `Accept-Encoding`-Header enthält. Wenn die Viewer-Anfrage den `Accept-Encoding` Header nicht enthält, CloudFront nimmt sie diesen Header nicht in den Cache-Schlüssel auf und nimmt ihn nicht in die entsprechende ursprüngliche Anfrage auf.

Wenn das Zwischenspeichern komprimierter Objekte für beide Komprimierungsformate aktiviert ist

Wenn der Viewer sowohl Gzip als auch Brotli unterstützt — das heißt, wenn sich die `br` Werte `gzip` und beide im `Accept-Encoding` Header der Viewer-Anfrage befinden — wird wie folgt vorgegangen: CloudFront

- Sie normalisiert den Header zu `Accept-Encoding: br,gzip` und schließt den normalisierten Header in den Cache-Schlüssel ein. Der Cache-Schlüssel enthält keine anderen Werte, die sich in dem vom Viewer gesendeten `Accept-Encoding`-Header befanden.
- Wenn der Edge-Standort ein mit Brotli oder Gzip komprimiertes Objekt im Cache enthält, das der Anforderung entspricht und nicht abgelaufen ist, gibt der Edge-Standort das Objekt an den Viewer zurück.
- Wenn der Edge-Standort kein komprimiertes Brotli- oder Gzip-Objekt im Cache hat, das der Anfrage entspricht und nicht abgelaufen ist, nimmt er den normalisierten Header () in die CloudFront entsprechende ursprüngliche Anfrage auf. `Accept-Encoding: br,gzip` Die Ursprungsanforderung enthält keine anderen Werte, die sich in dem vom Viewer gesendeten `Accept-Encoding`-Header befanden.

Wenn der Viewer ein Komprimierungsformat unterstützt, das andere aber nicht — zum Beispiel, wenn es sich um einen Wert im `Accept-Encoding` Header der Viewer-Anfrage `gzip` handelt, dies aber nicht `br` ist —, wird wie folgt vorgegangen: CloudFront

- Sie normalisiert den Header zu `Accept-Encoding: gzip` und schließt den normalisierten Header in den Cache-Schlüssel ein. Der Cache-Schlüssel enthält keine anderen Werte, die sich in dem vom Viewer gesendeten `Accept-Encoding`-Header befanden.
- Wenn der Edge-Standort ein mit Gzip komprimiertes Objekt im Cache enthält, das der Anforderung entspricht und nicht abgelaufen ist, gibt der Edge-Standort das Objekt an den Viewer zurück.
- Wenn der Edge-Standort kein Gzip-komprimiertes Objekt im Cache hat, das der Anfrage entspricht und nicht abgelaufen ist, CloudFront nimmt er den normalisierten Header (`Accept-Encoding: gzip`) in die entsprechende Ursprungsanforderung auf. Die

Ursprungsanforderung enthält keine anderen Werte, die sich in dem vom Viewer gesendeten `Accept-Encoding`-Header befanden.

Um zu verstehen, was CloudFront passiert, wenn der Viewer Brotli, aber nicht Gzip unterstützt, ersetzen Sie die beiden Komprimierungsformate im vorherigen Beispiel miteinander.

Falls der Viewer Brotli oder GZip nicht unterstützt — das heißt, der `Accept-Encoding` Header in der Viewer-Anfrage enthält keine Werte oder als Werte —: `br` `gzip` CloudFront

- Schließt den `Accept-Encoding`-Header nicht in den Cache-Schlüssel ein.
- Schließt `Accept-Encoding: identity` in die entsprechende Ursprungsanforderung ein. Die Ursprungsanforderung enthält keine anderen Werte, die sich in dem vom Viewer gesendeten `Accept-Encoding`-Header befanden.

Wenn die Zwischenspeicherung komprimierter Objekte für ein Komprimierungsformat aktiviert ist, aber nicht für das andere

Wenn der Viewer das Format unterstützt, für das das Caching aktiviert ist — zum Beispiel, wenn das Zwischenspeichern komprimierter Objekte für Gzip aktiviert ist und der Viewer Gzip unterstützt (`gzip`ist einer der Werte im Header der Viewer-Anfrage) — geht Folgendes vor: `Accept-Encoding` CloudFront

- Sie normalisiert den Header zu `Accept-Encoding: gzip` und schließt den normalisierten Header in den Cache-Schlüssel ein.
- Wenn der Edge-Standort ein mit Gzip komprimiertes Objekt im Cache enthält, das der Anforderung entspricht und nicht abgelaufen ist, gibt der Edge-Standort das Objekt an den Viewer zurück.
- Wenn der Edge-Standort kein Gzip-komprimiertes Objekt im Cache hat, das der Anfrage entspricht und nicht abgelaufen ist, wird der normalisierte Header () in die entsprechende CloudFront Ursprungsanforderung aufgenommen. `Accept-Encoding: gzip` Die Ursprungsanforderung enthält keine anderen Werte, die sich in dem vom Viewer gesendeten `Accept-Encoding`-Header befanden.

Dieses Verhalten ist identisch, wenn der Viewer sowohl Gzip als auch Brotli unterstützt (der `Accept-Encoding`-Header in der Viewer-Anforderung enthält `gzip` und `br` als Werte), da in diesem Szenario das Zwischenspeichern komprimierter Objekte für Brotli nicht aktiviert ist.

Um zu verstehen, was CloudFront passiert, wenn das Zwischenspeichern komprimierter Objekte für Brotli, aber nicht für Gzip aktiviert ist, ersetzen Sie die beiden Komprimierungsformate im vorherigen Beispiel durch einander.

Falls der Viewer das Komprimierungsformat, für das das Caching aktiviert ist, nicht unterstützt (der `Accept-Encoding` Header in der Viewer-Anfrage enthält nicht den Wert für dieses Format), gilt Folgendes: CloudFront

- Schließt den `Accept-Encoding`-Header nicht in den Cache-Schlüssel ein.
- Schließt `Accept-Encoding: identity` in die entsprechende Ursprungsanforderung ein. Die Ursprungsanforderung enthält keine anderen Werte, die sich in dem vom Viewer gesendeten `Accept-Encoding`-Header befanden.

Wenn das Zwischenspeichern komprimierter Objekte für beide Komprimierungsformate deaktiviert ist

Wenn das Zwischenspeichern komprimierter Objekte für beide Komprimierungsformate deaktiviert ist, wird der `Accept-Encoding` Header genauso CloudFront behandelt wie jeder andere HTTP-Header in der Viewer-Anforderung. Standardmäßig ist dieser nicht im Cache-Schlüssel und nicht in Ursprungsanforderungen enthalten. Sie können diesen wie jeden anderen HTTP-Header in eine Cache-Richtlinie oder eine Ursprungsanforderungsrichtlinie in die Header-Liste einfügen.

Erstellen von Cache-Richtlinien

Sie können eine Cache-Richtlinie verwenden, um das Cache-Trefferverhältnis zu verbessern, indem Sie die Werte (URL-Abfragezeichenfolgen, HTTP-Header und Cookies) steuern, die im Cache-Schlüssel enthalten sind. Sie können eine Cache-Richtlinie in der CloudFront Konsole, mit der AWS Command Line Interface (AWS CLI) oder mit der CloudFront API erstellen.

Nachdem Sie eine Cache-Richtlinie erstellt haben, fügen Sie sie einer oder mehreren Cache-Verhaltensweisen in einer CloudFront -Verteilung an.

Console

So erstellen Sie eine Cache-Richtlinie (Konsole):

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Seite Richtlinien in der CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home?#/policies.
2. Wählen Sie Create cache policy (Cache-Richtlinie erstellen).
3. Wählen Sie die gewünschte Einstellung für diese Cache-Richtlinie aus. Weitere Informationen finden Sie unter Grundlegendes zu Cache-Richtlinien.
4. Wenn Sie fertig sind, wählen Sie Erstellen.

Nachdem Sie eine Cache-Richtlinie erstellt haben, können Sie sie an eine Cache-Verhaltensweise anfügen.

So fügen Sie eine Cache-Richtlinie an eine vorhandene Verteilung an (Konsole):

1. Öffnen Sie die Seite Distributions (Verteilungen) in der CloudFront-Konsole unter https://console.aws.amazon.com/cloudfront/v4/home#/distributions.
2. Wählen Sie die Verteilung aus, die Sie aktualisieren möchten, und anschließend die Registerkarte Verhaltensweisen aus.
3. Wählen Sie das Cacheverhalten, das Sie aktualisieren möchten, und anschließend Bearbeiten aus.

   Um ein neues Cacheverhalten zu erstellen, wählen Sie Verhalten erstellen aus.
4. Stellen Sie im Abschnitt Cache-Schlüssel- und Ursprungsanforderungen sicher, dass Cache-Richtlinie und Ursprungsanforderungsrichtlinie ausgewählt sind.
5. Wählen Sie für Cache Policy (Cache-Richtlinie) die Cache-Richtlinie aus, die diesem Cache-Verhalten angefügt werden soll.
6. Wählen Sie unten auf der Seite die Option Änderungen speichern aus.

So fügen Sie eine Cache-Richtlinie an eine neue Verteilung an (Konsole):

1. Öffnen Sie die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home.
2. Wählen Sie Verteilung erstellen.
3. Stellen Sie im Abschnitt Cache-Schlüssel- und Ursprungsanforderungen sicher, dass Cache-Richtlinie und Ursprungsanforderungsrichtlinie ausgewählt sind.
4. Wählen Sie unter Cache policy (Cache-Richtlinie) die Cache-Richtlinie aus, die an das Standard-Cacheverhalten dieser Verteilung angefügt werden soll.
5. Wählen Sie die gewünschten Einstellungen für den Ursprung, das Standard-Cacheverhalten und andere Verteilungseinstellungen aus. Weitere Informationen finden Sie unter Referenz für alle Distributionseinstellungen.
6. Wenn Sie fertig sind, wählen Sie Verteilung erstellen aus.

CLI

Verwenden Sie den aws cloudfront create-cache-policy Befehl, um eine Cache-Richtlinie mit dem AWS Command Line Interface (AWS CLI) zu erstellen. Sie können die Eingabeparameter des Befehls in einer Eingabedatei bereitstellen, anstatt jeden einzelnen Parameter als Befehlszeileneingabe anzugeben.

So erstellen Sie eine Cache-Richtlinie (CLI mit Eingabedatei):

1. Verwenden Sie den folgenden Befehl, um eine Datei mit dem Namen `cache-policy.yaml` zu erstellen, die alle Eingabeparameter für den create-cache-policy-Befehl enthält.

   ```
   aws cloudfront create-cache-policy --generate-cli-skeleton yaml-input > cache-policy.yaml
   ```

2. Öffnen Sie die Datei mit dem Namen `cache-policy.yaml`, die Sie gerade erstellt haben. Bearbeiten Sie die Datei, um die gewünschten Cache-Richtlinieneinstellungen anzugeben, und speichern Sie die Datei. Sie können optionale Felder aus der Datei entfernen, erforderliche Felder dürfen jedoch nicht entfernt werden.

   Weitere Informationen zu den Cache-Richtlinieneinstellungen finden Sie unter Grundlegendes zu Cache-Richtlinien.

3. Verwenden Sie den folgenden Befehl, um die Cache-Richtlinie mit Eingabeparametern aus der `cache-policy.yaml`-Datei zu erstellen.

   ```
   aws cloudfront create-cache-policy --cli-input-yaml file://cache-policy.yaml
   ```

   Notieren Sie den `Id`-Wert in der Ausgabe des Befehls. Dies ist die Cache-Richtlinien-ID, und Sie benötigen sie, um die Cache-Richtlinie an das Cache-Verhalten einer CloudFront Distribution anzuhängen.

So fügen Sie eine Cache-Richtlinie an eine vorhandene Verteilung an (CLI mit Eingabedatei):

1. Verwenden Sie den folgenden Befehl, um die Verteilungskonfiguration für die CloudFront Distribution zu speichern, die Sie aktualisieren möchten. Ersetzen Sie *`distribution_ID`* durch die ID der Verteilung.

```
aws cloudfront get-distribution-config --id distribution_ID --output yaml >
 dist-config.yaml
```

2. Öffnen Sie die Datei mit dem Namen `dist-config.yaml`, die Sie gerade erstellt haben. Bearbeiten Sie die Datei, indem Sie die folgenden Änderungen an jeder Cache-Verhaltensweise vornehmen, die Sie aktualisieren, um eine Cache-Richtlinie zu verwenden.

   - Fügen Sie in der Cache-Verhaltensweise ein Feld mit dem Namen `CachePolicyId` hinzu. Verwenden Sie für den Wert des Feldes die Cache-Richtlinien-ID, die Sie nach dem Erstellen der Richtlinie notiert haben.
   - Entfernen Sie die Felder `MinTTL`, `MaxTTL`, `DefaultTTL` und `ForwardedValues` aus der Cache-Verhaltensweise. Diese Einstellungen werden in der Cache-Richtlinie angegeben, somit können Sie diese Felder und eine Cache-Richtlinie nicht in dasselbe Cache-Verhalten einschließen.
   - Benennen Sie das Feld `ETag` in `IfMatch` um, ändern Sie jedoch nicht den Wert des Feldes.

   Speichern Sie die Datei, wenn Sie fertig sind.

3. Verwenden Sie den folgenden Befehl, um die Verteilung so zu aktualisieren, dass die Cache-Richtlinie verwendet wird. Ersetzen Sie *`distribution_ID`* durch die ID der Verteilung.

```
aws cloudfront update-distribution --id distribution_ID --cli-input-yaml file://
dist-config.yaml
```

So fügen Sie eine Cache-Richtlinie an eine neue Verteilung an (CLI mit Eingabedatei):

1. Verwenden Sie den folgenden Befehl, um eine Datei mit dem Namen `distribution.yaml` zu erstellen, die alle Eingabeparameter für den create-distribution-Befehl enthält.

```
aws cloudfront create-distribution --generate-cli-skeleton yaml-input >
 distribution.yaml
```

2. Öffnen Sie die Datei mit dem Namen `distribution.yaml`, die Sie gerade erstellt haben. Geben Sie im Standard-Cacheverhalten in das `CachePolicyId`-Feld die Cache-Richtlinien-ID ein, die Sie nach dem Erstellen der Richtlinie notiert haben. Fahren Sie mit der Bearbeitung der Datei fort, um die gewünschten Verteilungseinstellungen anzugeben, und speichern Sie die Datei, wenn Sie fertig sind.

   Weitere Informationen zu den Verteilungseinstellungen finden Sie unter Referenz für alle Distributionseinstellungen.

3. Verwenden Sie den folgenden Befehl, um die Verteilung mit Eingabeparametern aus der Datei `distribution.yaml` zu erstellen.

```
aws cloudfront create-distribution --cli-input-yaml file://distribution.yaml
```

API

Um eine Cache-Richtlinie mit der CloudFront API zu erstellen, verwenden Sie CreateCachePolicy. Weitere Informationen zu den Feldern, die Sie in diesem API-Aufruf angeben, finden Sie in Grundlegendes zu Cache-Richtlinien und in der API-Referenzdokumentation für Ihr AWS SDK oder einen anderen API-Client.

Nachdem Sie eine Cache-Richtlinie erstellt haben, können Sie sie mit einem der folgenden API-Aufrufe an eine Cache-Verhaltensweise anfügen:

- Um es an ein Cache-Verhalten in einer vorhandenen Distribution anzuhängen, verwenden Sie UpdateDistribution.
- Um es an ein Cache-Verhalten in einer neuen Distribution anzuhängen, verwenden Sie CreateDistribution.

Geben Sie für beide API-Aufrufe die ID der Cache-Richtlinie im Feld `CachePolicyId` innerhalb eines Cache-Verhaltens an. Weitere Informationen zu den anderen Feldern, die Sie in diesen API-Aufrufen angeben, finden Referenz für alle Distributionseinstellungen Sie in der API-Referenzdokumentation für Ihr AWS SDK oder einen anderen API-Client.

Verwenden verwalteter Cache-Richtlinien

CloudFront bietet eine Reihe verwalteter Cache-Richtlinien, die Sie an jedes Cache-Verhalten Ihrer Distribution anhängen können. Bei Verwendung einer verwalteten Cache-Richtlinie müssen Sie keine eigene Cache-Richtlinie schreiben oder verwalten. Die verwalteten Richtlinien verwenden Einstellungen, die für bestimmte Anwendungsfälle optimiert sind.

Um eine verwaltete Cache-Richtlinie zu verwenden, fügen Sie sie einem Cache-Verhalten in Ihrer Distribution an. Der Prozess ist der gleiche wie beim Erstellen einer Cache-Richtlinie. Anstatt jedoch eine neue Cache-Richtlinie zu erstellen, fügen Sie einfach eine verwaltete Cache-Richtlinie an. Sie fügen die Richtlinie entweder nach Namen (mit der Konsole) oder nach ID (mit AWS CLI oder SDKs) hinzu. Die Namen und IDs sind im folgenden Abschnitt aufgeführt.

Weitere Informationen finden Sie unter Erstellen von Cache-Richtlinien.

In den folgenden Themen werden die verwalteten Richtlinien beschrieben, die Sie verwenden können.

Themen

- Amplify
- CachingDisabled
- CachingOptimized
- CachingOptimizedForUncompressedObjects
- Elementar- MediaPackage
- UseOriginCacheControlHeaders
- UseOriginCacheControlHeaders-QueryStrings

Amplify

Diese Richtlinie in der CloudFront Konsole anzeigen

Diese Richtlinie wurde für die Verwendung mit einem Ursprung entwickelt, bei dem es sich um eine AWS Amplify-Web-App handelt.

Wenn Sie CloudFormation die AWS CLI oder die CloudFront API verwenden, lautet die ID für diese Richtlinie:

```
2e54312d-136d-493c-8eb9-b001f22f67d2
```

Diese Richtlinie hat folgende Einstellungen:

- Mindest-TTL: 2 Sekunden
- Höchst-TTL: 600 Sekunden (10 Minuten)
- Standard-TTL: 2 Sekunden
- Im Cache-Schlüssel enthaltene Header:
  - `Authorization`
  - `CloudFront-Viewer-Country`
  - `Host`

  Der normalisierte Header `Accept-Encoding` ist auch enthalten, da die Einstellung für komprimierte Cache-Objekte aktiviert ist. Weitere Informationen finden Sie unter Komprimierungsunterstützung.
- Im Cache-Schlüssel enthaltene Cookies: Alle Cookies sind enthalten.
- Im Cache-Schlüssel enthaltene Abfragezeichenfolgen: Alle Abfragezeichenfolgen sind enthalten.
- Einstellung für komprimierte Cache-Objekte: Aktiviert. Weitere Informationen finden Sie unter Komprimierungsunterstützung.

> **Warning**
>
> Da diese Richtlinie eine Mindest-TTL von mehr als 0 hat, CloudFront wird Inhalt mindestens für die in der Mindest-TTL der Cache-Richtlinie angegebene Dauer zwischengespeichert, auch wenn die `private` Direktiven `Cache-Control: no-cacheno-store`, oder in den ursprünglichen Headern vorhanden sind.

AWS Amplify Cache-Richtlinien hosten

Amplify verwendet die folgenden verwalteten Cache-Richtlinien, um die Standard-Cache-Konfiguration für Kundenanwendungen zu optimieren:

- Amplify-Default
- Amplify- DefaultNoCookies
- Amplify- ImageOptimization
- Amplify- StaticContent

> **Note**
>
> Diese Richtlinien werden nur von Amplify verwendet. Es ist nicht zu empfehlen, diese Richtlinien für Ihre Distributionen zu verwenden.

Weitere Informationen zur Verwaltung der Cache-Konfiguration für Ihre von Amplify gehostete Anwendung finden Sie unter Verwaltung der Cache-Konfiguration im Benutzerhandbuch für Amplify Hosting.

CachingDisabled

Diese Richtlinie in der CloudFront Konsole anzeigen

Diese Richtlinie deaktiviert die Zwischenspeicherung. Diese Richtlinie ist für dynamische Inhalte und für Anforderungen nützlich, die nicht zwischengespeichert werden können.

Wenn Sie CloudFormation die AWS CLI oder die CloudFront API verwenden, lautet die ID für diese Richtlinie:

```
4135ea2d-6df8-44a3-9df3-4b5a84be39ad
```

Diese Richtlinie hat folgende Einstellungen:

- Mindest-TTL: 0 Sekunden
- Höchst-TTL: 0 Sekunden
- Standard-TTL: 0 Sekunden
- Im Cache-Schlüssel enthaltene Header: Keine
- Im Cache-Schlüssel enthaltene Cookies: Keine
- Im Cache-Schlüssel enthaltene Abfragezeichenfolgen: Keine
- Einstellung für komprimierte Cache-Objekte: Deaktiviert

CachingOptimized

Diese Richtlinie in der CloudFront Konsole anzeigen

Diese Richtlinie wurde entwickelt, um die Cache-Effizienz zu optimieren, indem die Werte, die im Cache-Schlüssel CloudFront enthalten sind, minimiert werden. CloudFront enthält

keine Abfragezeichenfolgen oder Cookies im Cache-Schlüssel und bezieht nur den normalisierten `Accept-Encoding` Header ein. Dadurch können CloudFront Objekte in den Komprimierungsformaten Gzip und Brotli separat zwischengespeichert werden, wenn der Ursprung sie zurückgibt oder wenn die Kantenkomprimierung aktiviert ist. CloudFront

Wenn Sie die oder die CloudFront API verwenden CloudFormation AWS CLI, lautet die ID für diese Richtlinie:

`658327ea-f89d-4fab-a63d-7e88639e58f6`

Diese Richtlinie hat folgende Einstellungen:

- Mindest-TTL: 1 Sekunde
- Höchst-TTL: 31.536.000 Sekunden (365 Tage).
- Standard-TTL: 86.400 Sekunden (24 Stunden).
- Im Cache-Schlüssel enthaltene Header: Es werden keine Header explizit eingefügt. Der normalisierte Header `Accept-Encoding` ist enthalten, da die Einstellung für komprimierte Cache-Objekte aktiviert ist. Weitere Informationen finden Sie unter Komprimierungsunterstützung.
- Im Cache-Schlüssel enthaltene Cookies: Keine.
- Im Cache-Schlüssel enthaltene Abfragezeichenfolgen: Keine.
- Einstellung für komprimierte Cache-Objekte: Aktiviert. Weitere Informationen finden Sie unter Komprimierungsunterstützung.

> **Warning**
>
> Da diese Richtlinie eine Mindest-TTL von mehr als 0 hat, CloudFront wird Inhalt mindestens für die in der Mindest-TTL der Cache-Richtlinie angegebene Dauer zwischengespeichert, auch wenn die `private` Direktiven `Cache-Control: no-cacheno-store`, oder in den ursprünglichen Headern vorhanden sind.

CachingOptimizedForUncompressedObjects

Diese Richtlinie in der Konsole anzeigen CloudFront

Diese Richtlinie wurde zur Optimierung der Cache-Effizienz entwickelt, indem die in den Cache-Schlüssel eingefügten Werte minimiert werden. Es sind keine Abfragezeichenfolgen, Header oder

Cookies enthalten. Diese Richtlinie ist identisch mit der vorherigen Richtlinie; die Einstellung für komprimierte Cache-Objekte ist jedoch deaktiviert.

Wenn Sie CloudFormation die AWS CLI oder die CloudFront API verwenden, lautet die ID für diese Richtlinie:

`b2884449-e4de-46a7-ac36-70bc7f1ddd6d`

Diese Richtlinie hat folgende Einstellungen:

- Mindest-TTL: 1 Sekunde
- Höchst-TTL: 31.536.000 Sekunden (365 Tage)
- Standard-TTL: 86.400 Sekunden (24 Stunden)
- Im Cache-Schlüssel enthaltene Header: Keine
- Im Cache-Schlüssel enthaltene Cookies: Keine
- Im Cache-Schlüssel enthaltene Abfragezeichenfolgen: Keine
- Einstellung für komprimierte Cache-Objekte: Deaktiviert

> **Warning**
>
> Da diese Richtlinie eine Mindest-TTL von mehr als 0 hat, CloudFront wird Inhalt mindestens für die in der Mindest-TTL der Cache-Richtlinie angegebene Dauer zwischengespeichert, auch wenn die `private` Direktiven `Cache-Control: no-cache` `no-store`, oder in den ursprünglichen Headern vorhanden sind.

Elementar- MediaPackage

Diese Richtlinie in der CloudFront Konsole anzeigen

Diese Richtlinie wurde für die Verwendung mit einem Ursprung entwickelt, bei dem es sich um einen AWS Elemental MediaPackage -Endpunkt handelt.

Wenn Sie CloudFormation die AWS CLI oder die CloudFront API verwenden, lautet die ID für diese Richtlinie:

`08627262-05a9-4f76-9ded-b50ca2e3a84f`

Diese Richtlinie hat folgende Einstellungen:

- Mindest-TTL: 0 Sekunden
- Höchst-TTL: 31.536.000 Sekunden (365 Tage)
- Standard-TTL: 86.400 Sekunden (24 Stunden)
- Im Cache-Schlüssel enthaltene Header:
  - `Origin`

  Der normalisierte Header `Accept-Encoding` ist enthalten, da die Einstellung für komprimierte Cache-Objekte für Gzip aktiviert ist. Weitere Informationen finden Sie unter Komprimierungsunterstützung.
- Im Cache-Schlüssel enthaltene Cookies: Keine
- Im Cache-Schlüssel enthaltene Abfragezeichenfolgen:
  - `aws.manifestfilter`
  - `start`
  - `end`
  - `m`
- Einstellung für komprimierte Cache-Objekte: Für Gzip aktiviert. Weitere Informationen finden Sie unter Komprimierungsunterstützung.

UseOriginCacheControlHeaders

Diese Richtlinie in der CloudFront Konsole anzeigen

Diese Richtlinie ist für die Verwendung mit einem Ursprung konzipiert, der die HTTP-Antwortheader `Cache-Control` zurückgibt und keine unterschiedlichen Inhalte bereitstellt, die auf den in der Abfragezeichenfolge vorhandenen Werten basieren. Wenn Ihr Ursprung abhängig von den Werten in der Abfragezeichenfolge unterschiedliche Inhalte bereitstellt, sollten Sie die Verwendung von UseOriginCacheControlHeaders-QueryStrings in Betracht ziehen.

Wenn Sie CloudFormation die AWS CLI oder die CloudFront API verwenden, lautet die ID für diese Richtlinie:

`83da9c7e-98b4-4e11-a168-04f0df8e2c65`

Diese Richtlinie hat folgende Einstellungen:

- Mindest-TTL: 0 Sekunden

- Höchst-TTL: 31.536.000 Sekunden (365 Tage)
- Standard-TTL: 0 Sekunden
- Im Cache-Schlüssel enthaltene Header:
  - `Host`
  - `Origin`
  - `X-HTTP-Method-Override`
  - `X-HTTP-Method`
  - `X-Method-Override`

  Der normalisierte Header `Accept-Encoding` ist auch enthalten, da die Einstellung für komprimierte Cache-Objekte aktiviert ist. Weitere Informationen finden Sie unter Komprimierungsunterstützung.
- Im Cache-Schlüssel enthaltene Cookies: Alle Cookies sind enthalten.
- Im Cache-Schlüssel enthaltene Abfragezeichenfolgen: Keine.
- Einstellung für komprimierte Cache-Objekte: Aktiviert. Weitere Informationen finden Sie unter Komprimierungsunterstützung.

UseOriginCacheControlHeaders-QueryStrings

Diese Richtlinie in der CloudFront Konsole anzeigen

Diese Richtlinie ist für die Verwendung mit einem Ursprung konzipiert, der die HTTP-Antwortheader `Cache-Control` zurückgibt und unterschiedliche Inhalte bereitstellt, die auf den in der Abfragezeichenfolge vorhandenen Werten basieren. Wenn Ihr Ursprung abhängig von den Werten in der Abfragezeichenfolge keine unterschiedlichen Inhalte bereitstellt, sollten Sie die Verwendung von UseOriginCacheControlHeaders in Betracht ziehen.

Wenn Sie CloudFormation die AWS CLI oder die CloudFront API verwenden, lautet die ID für diese Richtlinie:

`4cc15a8a-d715-48a4-82b8-cc0b614638fe`

Diese Richtlinie hat folgende Einstellungen:

- Mindest-TTL: 0 Sekunden
- Höchst-TTL: 31.536.000 Sekunden (365 Tage)

- Standard-TTL: 0 Sekunden
- Im Cache-Schlüssel enthaltene Header:
  - `Host`
  - `Origin`
  - `X-HTTP-Method-Override`
  - `X-HTTP-Method`
  - `X-Method-Override`

  Der normalisierte Header `Accept-Encoding` ist auch enthalten, da die Einstellung für komprimierte Cache-Objekte aktiviert ist. Weitere Informationen finden Sie unter Komprimierungsunterstützung.
- Im Cache-Schlüssel enthaltene Cookies: Alle Cookies sind enthalten.
- Im Cache-Schlüssel enthaltene Abfragezeichenfolgen: Alle Abfragezeichenfolgen sind enthalten.
- Einstellung für komprimierte Cache-Objekte: Aktiviert. Weitere Informationen finden Sie unter Komprimierungsunterstützung.

Grundlegendes zum Cache-Schlüssel

Der *Cache-Schlüssel* bestimmt, ob eine Viewer-Anfrage an einen CloudFront Edge-Standort zu einem *Cache-Treffer* führt. Der Cache-Schlüssel ist der eindeutige Bezeichner für ein Objekt im Cache. Jedes Objekt im Cache verfügt über einen eindeutigen Cache-Schlüssel.

Ein Cache-Treffer tritt auf, wenn eine Viewer-Anforderung denselben Cache-Schlüssel wie eine vorherige Anforderung generiert und sich das Objekt für diesen Cache-Schlüssel im Cache des Edge-Standorts befindet und gültig ist. Bei einem Cache-Treffer wird das angeforderte Objekt dem Betrachter von einer CloudFront Edge-Position aus zugestellt, was die folgenden Vorteile hat:

- Geringere Auslastung Ihres Ursprungs-Servers
- Reduzierte Latenz für den Viewer

Sie können eine bessere Leistung für Ihre Website oder Anwendung erzielen, wenn Sie eine höhere Cache-Trefferquote haben (ein höherer Anteil an Viewer-Anforderungen, die zu einem Cache-Treffer führen). Eine Möglichkeit, die Cache-Trefferquote zu verbessern, besteht darin, nur die minimal notwendigen Werte in den Cache-Schlüssel aufzunehmen. Weitere Informationen finden Sie in den folgenden Abschnitten.

Sie können die Werte (URL-Abfragezeichenfolgen, HTTP-Header und Cookies) im Cache-Schlüssel mithilfe einer Cache-Richtlinie ändern. (Sie können den Cache-Schlüssel auch ändern, indem Sie ein Lambda @Edgefunction oder eine CloudFront Funktion auf Viewer-Anfrage verwenden.) Bevor Sie den Cache-Schlüssel ändern, ist es wichtig zu verstehen, wie Ihre Anwendung entworfen ist und wann und wie sie verschiedene Antworten basierend auf den Eigenschaften der Viewer-Anforderung bereitstellen kann. Wenn ein Wert in der Viewer-Anforderung die Antwort bestimmt, die Ihr Ursprung zurückgibt, sollten Sie diesen Wert in den Cache-Schlüssel aufnehmen. Wenn Sie jedoch einen Wert in den Cache-Schlüssel einfügen, der sich nicht auf die Antwort auswirkt, die Ihr Ursprung zurückgibt, kann es passieren, dass Sie doppelte Objekte zwischenspeichern.

Standard-Cache-Schlüssel

Standardmäßig enthält der Cache-Schlüssel für eine CloudFront Distribution die folgenden Informationen:

- Der Domainname der CloudFront Distribution (z. B. d111111abcdef8.cloudfront.net)
- Der URL-Pfad des angeforderten Objekts (z. B, `/content/stories/example-story.html`)

> **Note**
>
> Die`OPTIONS`-Methode ist im Cache-Schlüssel für `OPTIONS`-Anforderungen enthalten. Dies bedeutet, dass Antworten auf `OPTIONS`-Anforderungen getrennt von Antworten auf `GET`- und `HEAD`-Anforderungen zwischengespeichert werden.

Andere Werte aus der Viewer-Anforderung sind standardmäßig nicht im Cache-Schlüssel enthalten. Betrachten Sie die folgende HTTP-Anforderung von einem Webbrowser.

```
           GET /content/stories/example-story.html?ref=0123abc&split-pages=false
HTTP/1.1
           Host: d111111abcdef8.cloudfront.net
           User-Agent: Mozilla/5.0 Gecko/20100101 Firefox/68.0
           Accept: text/html,*/*
           Accept-Language: en-US,en
           Cookie: session_id=01234abcd
           Referer: https://news.example.com/
```

Wenn eine Viewer-Anfrage wie in diesem Beispiel an einem CloudFront Edge-Standort eingeht, CloudFront verwendet der Cache-Schlüssel, um zu ermitteln, ob ein Cache-Treffer vorliegt. Standardmäßig sind nur die folgenden Komponenten der Anforderung im Cache-Schlüssel enthalten: `/content/stories/example-story.html` und `d111111abcdef8.cloudfront.net`. Wenn sich das angeforderte Objekt nicht im Cache befindet (ein Cache-Fehler), wird eine Anfrage an den Ursprung CloudFront gesendet, um das Objekt abzurufen. Nachdem das Objekt abgerufen wurde, wird es an den Viewer CloudFront zurückgegeben und im Cache des Edge-Standorts gespeichert.

Wenn eine weitere Anfrage für dasselbe Objekt CloudFront empfangen wird, wird das Objekt, wie durch den Cache-Schlüssel bestimmt, CloudFront sofort an den Betrachter weitergeleitet, ohne eine Anfrage an den Ursprung zu senden. Beachten Sie beispielsweise die folgende HTTP-Anforderung, die nach der vorherigen Anforderung eingeht.

```
            GET /content/stories/example-story.html?ref=xyz987&split-pages=true
HTTP/1.1
            Host: d111111abcdef8.cloudfront.net
            User-Agent: Mozilla/5.0 AppleWebKit/537.36 Chrome/83.0.4103.116
            Accept: text/html,*/*
            Accept-Language: en-US,en
            Cookie: session_id=wxyz9876
            Referer: https://rss.news.example.net/
```

Diese Anforderung gilt für dasselbe Objekt wie die vorherige Anforderung, unterscheidet sich jedoch von der vorherigen Anforderung. Sie hat eine andere URL-Abfragezeichenfolge, verschiedene `User-Agent`- und `Referer`-Header und ein anderes `session_id`- Cookie. Keiner dieser Werte ist jedoch standardmäßig Teil des Cache-Schlüssels, so dass diese zweite Anforderung zu einem Cache-Treffer führt.

Anpassen des Cache-Schlüssels

In einigen Fällen möchten Sie möglicherweise mehr Informationen in den Cache-Schlüssel aufnehmen, obwohl dies zu weniger Cache-Treffern führen kann. Sie können angeben, was in dem Cache-Schlüssel enthalten sein soll, indem Sie eine Cache-Richtlinie verwenden.

Wenn Ihr Ursprungs-Server beispielsweise den `Accept-Language`-HTTP-Header in Viewer-Anforderungen verwendet, um unterschiedliche Inhalte basierend auf der Sprache des Viewers zurückzugeben, sollten Sie diesen Header möglicherweise in den Cache-Schlüssel einfügen. Wenn Sie das tun, CloudFront verwendet diesen Header, um Cache-Treffer zu ermitteln, und schließt den

Header in Quellanfragen ein (Anfragen, die CloudFront an den Ursprung gesendet werden, wenn ein Cache-Fehler auftritt).

Eine mögliche Folge der Aufnahme zusätzlicher Werte in den Cache-Schlüssel ist, dass aufgrund der Variation, die bei den Viewer-Anfragen auftreten kann, CloudFront möglicherweise doppelte Objekte zwischengespeichert werden. Beispielsweise können Viewer einen der folgenden Werte für die `Accept-Language`-Header senden:

- `en-US,en`
- `en,en-US`
- `en-US, en`
- `en-US`

All diese unterschiedlichen Werte deuten darauf hin, dass die Sprache des Betrachters Englisch ist, aber die Variation kann CloudFront dazu führen, dass dasselbe Objekt mehrfach zwischengespeichert wird. Dies kann Cache-Treffer reduzieren und die Anzahl der Ursprungsanforderungen erhöhen. Sie können diese Dopplung vermeiden, indem Sie den `Accept-Language`-Header nicht in den Cache-Schlüssel einfügen und stattdessen Ihre Website oder Anwendung so konfigurieren, dass verschiedene URLs für Inhalte in verschiedenen Sprachen verwendet werden (z. B. `/en-US/content/stories/example-story.html`).

Bei jedem gegebenen Wert, den Sie in den Cache-Schlüssel einschließen möchten, sollten Sie sicherstellen, dass Sie verstehen, wie viele verschiedene Variationen dieses Werts in Viewer-Anforderungen angezeigt werden können. Bei bestimmten Anforderungswerten ist es selten sinnvoll, sie in den Cache-Schlüssel aufzunehmen. Beispielsweise kann der `User-Agent`-Header Tausende von eindeutigen Variationen haben, daher ist er im Allgemeinen kein guter Kandidat für die Aufnahme in den Cache-Schlüssel. Cookies, die benutzerspezifische oder sitzungsspezifische Werte aufweisen und über Tausende (oder sogar Millionen) von Anforderungen eindeutig sind, sind auch keine guten Kandidaten für die Aufnahme in Cache-Schlüssel. Wenn Sie diese Werte in den Cache-Schlüssel aufnehmen, führt jede eindeutige Variation zu einer weiteren Kopie des Objekts im Cache. Wenn diese Kopien des Objekts nicht eindeutig sind oder wenn Sie eine so großen Anzahl von leicht unterschiedlichen Objekten haben, dass jedes Objekt nur eine kleine Anzahl von Cache-Treffern erhält, sollten Sie einen anderen Ansatz in Betracht ziehen. Sie können diese hochvariablen Werte aus dem Cache-Schlüssel ausschließen oder Objekte als nicht zwischenspeicherbar markieren.

Seien Sie vorsichtig, wenn Sie den Cache-Schlüssel anpassen. Manchmal ist dies wünschenswert, es kann aber unbeabsichtigte Konsequenzen haben, wie z. B. das Zwischenspeichern von

doppelten Objekten, die Verringerung der Cache-Trefferquote und die Erhöhung der Anzahl der Ursprungsanforderungen. Manchmal benötigt Ihre ursprüngliche Website oder Anwendung bestimmte Werte aus Zuschaueranfragen für Analysen, Telemetrie oder andere Zwecke, aber diese Werte ändern nicht das Objekt, das der Ursprung zurückgibt. Verwenden Sie in diesem Fall eine Quellanforderungsrichtlinie, um diese Werte in Quellanfragen, aber nicht in den Cache-Schlüssel aufzunehmen.

Steuern von Ursprungsanforderungen anhand einer Richtlinie

Wenn eine Viewer-Anfrage zu einem Cache-Fehler CloudFront führt (das angeforderte Objekt wird am Edge-Standort nicht zwischengespeichert), CloudFront sendet er eine Anfrage an den Ursprung, um das Objekt abzurufen. Dies wird als Ursprungsanforderung bezeichnet. Die Ursprungsanforderung enthält stets die folgenden Informationen aus der Anforderung des Viewers:

- Den URL-Pfad (nur den Pfad, ohne URL-Abfragezeichenfolgen oder Domänennamen)
- Den Text der Anforderung (wenn vorhanden)
- Die HTTP-Header, die CloudFront automatisch in jeder ursprünglichen Anfrage enthalten sind, einschließlich`Host`, und `User-Agent` `X-Amz-Cf-Id`

Andere Informationen aus der Viewer-Anforderung, z. B. URL-Abfragezeichenfolgen, HTTP-Header und Cookies, sind standardmäßig nicht in der Ursprungsanforderung enthalten. (Ausnahme: Leitet bei älteren Cache-Einstellungen CloudFront die Header standardmäßig an Ihren Ursprung weiter.) Möglicherweise möchten Sie jedoch einige dieser anderen Informationen am Ursprung erhalten, um z. B. Daten für Analysen oder Telemetrie zu sammeln. Sie können eine Ursprungsanforderungsrichtlinie verwenden, um die Informationen zu steuern, die in einer Ursprungsanforderung enthalten sind.

Ursprungsanforderungsrichtlinien unterscheiden sich von Cache-Richtlinien. Diese steuern den Cache-Schlüssel. So können Sie zusätzliche Informationen am Ursprung erhalten und auch eine gute Cache-Trefferrate wahren (der Anteil der Viewer-Anforderungen, die zu einem Cache-Treffer führen). Hierzu steuern Sie separat, welche Informationen in Ursprungsanforderungen enthalten sind (über die Ursprungsanforderungsrichtlinie) und welche Informationen im Cache-Schlüssel enthalten sind (über die Cache-Richtlinie).

Auch wenn dies zwei getrennte Arten von Richtlinien sind, sind sie verwandt. Alle URL-Abfragezeichenfolgen, HTTP-Header und Cookies, die Sie in den Cache-Schlüssel (über eine Cache-Richtlinie) einfügen, werden automatisch auch in Ursprungsanforderungen eingefügt. Mithilfe der Ursprungsanforderungsrichtlinie können Sie die Informationen angeben, die Sie in Ursprungsanforderungen, jedoch nicht in den Cache-Schlüssel einfügen möchten. Genau wie bei einer Cache-Richtlinie fügen Sie eine Origin-Request-Richtlinie an ein oder mehrere Cache-Verhaltensweisen in einer CloudFront Distribution an.

Sie können eine Ursprungsanforderungsrichtlinie auch verwenden, um einer Ursprungsanforderung zusätzliche HTTP-Header hinzuzufügen, die nicht in der Viewer-Anforderung enthalten waren. Diese zusätzlichen Header werden CloudFront vor dem Senden der ursprünglichen Anfrage hinzugefügt, wobei die Header-Werte automatisch auf der Grundlage der Viewer-Anfrage bestimmt werden. Weitere Informationen finden Sie unter the section called “CloudFront Anforderungsheader hinzufügen”.

Themen

- Grundlegendes zu Ursprungsanforderungsrichtlinien
- Erstellen von Ursprungsanforderungsrichtlinien
- Verwenden verwalteter Ursprungsanforderungsrichtlinien
- CloudFront Anforderungsheader hinzufügen
- Erläuterung des Zusammenwirkens von Ursprungsanforderungsrichtlinien und Cache-Richtlinien

Grundlegendes zu Ursprungsanforderungsrichtlinien

CloudFront bietet einige vordefinierte Richtlinien für Anfragen mit Herkunft, die als verwaltete Richtlinien bezeichnet werden, für allgemeine Anwendungsfälle. Sie können diese verwalteten Richtlinien verwenden oder eine eigene Ursprungsanforderungsrichtlinie speziell für Ihre Anforderungen erstellen. Weitere Informationen zu verwalteten Richtlinien finden Sie unter Verwenden verwalteter Ursprungsanforderungsrichtlinien.

Eine Ursprungsanforderungsrichtlinie enthält die folgenden Einstellungen, die in Richtlinieninformationen und Ursprungsanforderungseinstellungen unterteilt werden.

Richtlinieninformationen

Name

Ein Name zur Identifizierung der Ursprungsanforderungsrichtlinie. Sie verwenden den Namen in der Konsole, um die Ursprungsanforderungsrichtlinie einem Cacheverhalten anzufügen.

Beschreibung

Ein Kommentar zur Beschreibung der Ursprungsanforderungsrichtlinie. Dieser Schritt ist optional.

Ursprungsanforderungseinstellungen

Die Einstellungen für Ursprungsanfragen geben die Werte in Viewer-Anfragen an, die in Anfragen enthalten sind, die an den Ursprung CloudFront gesendet werden (sogenannte Origin-Anfragen). Bei den Werten kann es sich um URL-Abfragezeichenfolgen, HTTP-Header und Cookies handeln. Die von Ihnen angegebenen Werte werden in Ursprungsanforderungen, jedoch nicht in den Cache-Schlüssel eingefügt. Informationen zum Steuern des Cache-Schlüssels finden Sie unter Steuern des Cache-Schlüssels mit einer Richtlinie.

Header

Die HTTP-Header in Viewer-Anfragen, die in CloudFront ursprünglichen Anfragen enthalten sind. Sie können für Header eine der folgenden Einstellungen auswählen:

- None (Keine) – Die HTTP-Header in Betrachteranfragen werden nicht in Ursprungsanforderungen eingefügt.
- All viewer headers (Alle Betrachter-Header) – Alle HTTP-Header in Betrachteranfragen werden in Ursprungsanforderungen eingefügt.
- Alle Viewer-Header und die folgenden CloudFront Header — Alle HTTP-Header in Viewer-Anfragen sind in den ursprünglichen Anfragen enthalten. Darüber hinaus geben Sie an, welche der CloudFront Header Sie zu den ursprünglichen Anfragen hinzufügen möchten. Weitere Hinweise zu den CloudFront Headern finden Sie unter. the section called “CloudFront Anforderungsheader hinzufügen”
- Die folgenden Header einschließen – Sie geben an, welche HTTP-Header in Ursprungsanforderungen eingeschlossen werden.

  > **Note**
  >
  > Geben Sie keinen Header an, der bereits in Ihren Einstellungen für benutzerdefinierte Ursprungs-Header enthalten ist. Weitere Informationen finden Sie unter Konfiguriere CloudFront , um benutzerdefinierte Header zu Quellanfragen hinzuzufügen.

- Alle Viewer-Header außer – Sie geben an, welche HTTP-Header nicht in Ursprungsanforderungen enthalten sind. Alle anderen HTTP-Header in Viewer-Anforderungen, mit Ausnahme der angegebenen, sind enthalten.

Wenn Sie die Einstellung Alle Viewer-Header und die folgenden CloudFront Header, Folgende Header einbeziehen oder Alle Viewer-Header außer verwenden, geben Sie HTTP-Header nur

anhand des Header-Namens an. CloudFront schließt den vollständigen Header, einschließlich seines Werts, in ursprüngliche Anfragen ein.

> **Note**
>
> Wenn Sie die Einstellung Alle Viewer-Header außer verwenden, um den `Host` Header des Viewers zu entfernen, wird der ursprünglichen Anfrage ein neuer `Host` Header mit dem Domainnamen des Ursprungs CloudFront hinzugefügt.

Cookies

Die Cookies in Viewer-Anfragen, CloudFront einschließlich in ursprünglichen Anfragen. Für Cookies können Sie eine der folgenden Einstellungen auswählen:

- Keine – Die Cookies in Viewer-Anforderungen werden nicht in Ursprungsanforderungen eingefügt.
- Alle – Alle Cookies in Viewer-Anforderungen werden in Ursprungsanforderungen eingefügt.
- Die folgenden Cookies einschließen – Sie geben an, welche Cookies in Viewer-Anforderungen in Ursprungsanforderungen eingefügt werden.
- Alle Cookies außer – Sie geben an, welche Cookies in Viewer-Anforderungen nicht in Ursprungsanforderungen eingefügt werden. Alle anderen Cookies in Viewer-Anfragen werden eingefügt.

Wenn Sie die Einstellung „Folgende Cookies einbeziehen“ oder „Alle Cookies außer“ verwenden, geben Sie Cookies nur anhand ihres Namens an. CloudFront schließt das vollständige Cookie, einschließlich seines Werts, in ursprüngliche Anfragen ein.

Abfragezeichenfolgen

Die URL-Abfragezeichenfolgen in Viewer-Anfragen, CloudFront einschließlich in ursprünglichen Anfragen. Für Abfragezeichenfolgen können Sie eine der folgenden Einstellungen auswählen:

- None (Keine) – Die Abfragezeichenfolgen in Betrachteranfragen werden nicht in Ursprungsanforderungen eingefügt.
- All (Alle) – Alle Abfragezeichenfolgen in Betrachteranfragen werden in Ursprungsanforderungen eingefügt.
- Die folgenden Abfragezeichenfolgen einschließen – Sie geben an, welche Abfragezeichenfolgen in Viewer-Anforderungen in Ursprungsanforderungen eingefügt werden.

- Alle Abfragezeichenfolgen außer – Sie geben an, welche Abfragezeichenfolgen in Viewer-Anforderungen nicht in Ursprungsanforderungen eingefügt werden. Alle anderen Abfragezeichenfolgen werden eingefügt.

Wenn Sie die Einstellung Folgende Abfragezeichenfolgen einbeziehen oder Alle Abfragezeichenfolgen außer verwenden, geben Sie Abfragezeichenfolgen nur anhand ihres Namens an. CloudFront schließt die vollständige Abfragezeichenfolge, einschließlich ihres Werts, in ursprüngliche Anfragen ein.

Erstellen von Ursprungsanforderungsrichtlinien

Sie können eine Richtlinie für ursprüngliche Anfragen verwenden, um die Werte (URL-Abfragezeichenfolgen, HTTP-Header und Cookies) zu steuern, die in Anfragen enthalten sind, die CloudFront an Ihren Absender gesendet werden. Sie können eine Richtlinie für Ursprungsanfragen in der CloudFront Konsole, mit der AWS Command Line Interface (AWS CLI) oder mit der CloudFront API erstellen.

Nachdem Sie eine Origin-Anforderungsrichtlinie erstellt haben, fügen Sie sie einem oder mehreren Cache-Verhalten in einer CloudFront Distribution hinzu.

Ursprungsanforderungsrichtlinien sind nicht erforderlich. Wenn einem Cacheverhalten keine Ursprungsanforderungsrichtlinie angefügt ist, enthält die Ursprungsanforderung alle Werte, die in der Cache-Richtlinie angegeben sind, jedoch keine weiteren Werte.

> **Note**
>
> Um eine Ursprungsanforderungsrichtlinie verwenden zu können, muss das Cacheverhalten auch eine Cache-Richtlinie verwenden. Sie können eine Ursprungsanforderungsrichtlinie in einem Cacheverhalten nicht ohne eine Cache-Richtlinie verwenden.

Console

So erstellen Sie eine Ursprungsanforderungsrichtlinie (Konsole)

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Seite Richtlinien in der CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home?#/policies.
2. Wählen Sie Ursprungsanforderung und anschließend Ursprungsanforderungsrichtlinie erstellen aus.

3. Wählen Sie die gewünschte Einstellung für diese Ursprungsanforderungsrichtlinie aus. Weitere Informationen finden Sie unter Grundlegendes zu Ursprungsanforderungsrichtlinien.
4. Wenn Sie fertig sind, wählen Sie Erstellen.

Nach der Erstellung einer Ursprungsanforderungsrichtlinie können Sie diese einem Cacheverhalten anfügen.

So fügen Sie eine Ursprungsanforderungsrichtlinie an eine vorhandene Verteilung an (Konsole)

1. Öffnen Sie die Seite Distributionen in der CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home#/distributions.
2. Wählen Sie die Verteilung aus, die Sie aktualisieren möchten, und anschließend die Registerkarte Verhaltensweisen aus.
3. Wählen Sie das Cacheverhalten, das Sie aktualisieren möchten, und anschließend Bearbeiten aus.

   Um ein neues Cacheverhalten zu erstellen, wählen Sie Verhalten erstellen aus.
4. Stellen Sie im Abschnitt Cache-Schlüssel- und Ursprungsanforderungen sicher, dass Cache-Richtlinie und Ursprungsanforderungsrichtlinie ausgewählt sind.
5. Wählen Sie unter Ursprungsanforderungsrichtlinie die Ursprungsanforderungsrichtlinie aus, die diesem Cacheverhalten angefügt werden soll.
6. Wählen Sie unten auf der Seite die Option Änderungen speichern aus.

So fügen Sie eine Ursprungsanforderungsrichtlinie an eine neue Verteilung an (Konsole)

1. Öffnen Sie die CloudFront Konsole unter. https://console.aws.amazon.com/cloudfront/v4/home
2. Wählen Sie Verteilung erstellen.
3. Stellen Sie im Abschnitt Cache-Schlüssel- und Ursprungsanforderungen sicher, dass Cache-Richtlinie und Ursprungsanforderungsrichtlinie ausgewählt sind.
4. Wählen Sie für Origin request policy (Ursprungsanforderungsrichtlinie) die Ursprungsanforderungsrichtlinie aus, die dem Standard-Cacheverhalten dieser Verteilung angefügt werden soll.

5. Wählen Sie die gewünschten Einstellungen für den Ursprung, das Standard-Cacheverhalten und andere Verteilungseinstellungen aus. Weitere Informationen finden Sie unter Referenz für alle Distributionseinstellungen.
6. Wenn Sie fertig sind, wählen Sie Verteilung erstellen aus.

CLI

Verwenden Sie den aws cloudfront create-origin-request-policy Befehl, um eine Origin-Request-Richtlinie mit dem AWS Command Line Interface (AWS CLI) zu erstellen. Sie können die Eingabeparameter des Befehls in einer Eingabedatei bereitstellen, anstatt jeden einzelnen Parameter als Befehlszeileneingabe anzugeben.

So erstellen Sie eine Ursprungsanforderungsrichtlinie (CLI mit Eingabedatei)

1. Verwenden Sie den folgenden Befehl, um eine Datei mit dem Namen `origin-request-policy.yaml` zu erstellen, die alle Eingabeparameter für den create-origin-request-policy-Befehl enthält.

```
aws cloudfront create-origin-request-policy --generate-cli-skeleton yaml-input >
 origin-request-policy.yaml
```

2. Öffnen Sie die Datei mit dem Namen `origin-request-policy.yaml`, die Sie gerade erstellt haben. Bearbeiten Sie die Datei, um die gewünschten Einstellungen für die Ursprungsanforderungsrichtlinie anzugeben, und speichern Sie die Datei. Sie können optionale Felder aus der Datei entfernen, erforderliche Felder dürfen jedoch nicht entfernt werden.

   Weitere Informationen zu den Einstellungen für die Ursprungsanforderungsrichtlinie finden Sie unter Grundlegendes zu Ursprungsanforderungsrichtlinien.
3. Verwenden Sie den folgenden Befehl, um die Ursprungsanforderungsrichtlinie mit Eingabeparametern aus der Datei `origin-request-policy.yaml` zu erstellen.

```
aws cloudfront create-origin-request-policy --cli-input-yaml file://origin-
request-policy.yaml
```

Notieren Sie den `Id`-Wert in der Ausgabe des Befehls. Dies ist die Richtlinien-ID für die ursprüngliche Anfrage. Sie benötigen sie, um die Richtlinie für die Ursprungsanforderung an das Cache-Verhalten einer CloudFront Distribution anzuhängen.

So fügen Sie eine Ursprungsanforderungsrichtlinie an eine vorhandene Verteilung an (CLI mit Eingabedatei)

1. Verwenden Sie den folgenden Befehl, um die Verteilungskonfiguration für die CloudFront Distribution zu speichern, die Sie aktualisieren möchten. Ersetzen Sie *`distribution_ID`* durch die ID der Verteilung.

```
aws cloudfront get-distribution-config --id distribution_ID --output yaml >
 dist-config.yaml
```

2. Öffnen Sie die Datei mit dem Namen `dist-config.yaml`, die Sie gerade erstellt haben. Bearbeiten Sie die Datei und nehmen Sie die folgenden Änderungen an jedem Cacheverhalten vor, das Sie aktualisieren, um eine Ursprungsanforderungsrichtlinie zu verwenden.

   - Fügen Sie in der Cache-Verhaltensweise ein Feld mit dem Namen `OriginRequestPolicyId` hinzu. Verwenden Sie als Wert des Feldes die Ursprungsanforderungsrichtlinien-ID, die Sie nach dem Erstellen der Richtlinie notiert haben.
   - Benennen Sie das Feld `ETag` in `IfMatch` um, ändern Sie jedoch nicht den Wert des Feldes.

   Speichern Sie die Datei, wenn Sie fertig sind.

3. Verwenden Sie den folgenden Befehl, um die Verteilung zur Verwendung der Ursprungsanforderungsrichtlinie zu aktualisieren. Ersetzen Sie *`distribution_ID`* durch die ID der Verteilung.

```
aws cloudfront update-distribution --id distribution_ID --cli-input-yaml file://
dist-config.yaml
```

So fügen Sie eine Ursprungsanforderungsrichtlinie an eine neu Verteilung an (CLI mit Eingabedatei)

1. Verwenden Sie den folgenden Befehl, um eine Datei mit dem Namen `distribution.yaml` zu erstellen, die alle Eingabeparameter für den create-distribution-Befehl enthält.

```
aws cloudfront create-distribution --generate-cli-skeleton yaml-input >
 distribution.yaml
```

2. Öffnen Sie die Datei mit dem Namen `distribution.yaml`, die Sie gerade erstellt haben. Geben Sie im Standard-Cacheverhalten in das Feld `OriginRequestPolicyId` die Ursprungsanforderungsrichtlinien-ID ein, die Sie nach dem Erstellen der Richtlinie notiert haben. Fahren Sie mit der Bearbeitung der Datei fort, um die gewünschten Verteilungseinstellungen anzugeben, und speichern Sie die Datei, wenn Sie fertig sind.

   Weitere Informationen zu den Verteilungseinstellungen finden Sie unter Referenz für alle Distributionseinstellungen.

3. Verwenden Sie den folgenden Befehl, um die Verteilung mit Eingabeparametern aus der Datei `distribution.yaml` zu erstellen.

```
aws cloudfront create-distribution --cli-input-yaml file://distribution.yaml
```

API

Um mit der CloudFront API eine Origin-Request-Richtlinie zu erstellen, verwenden Sie CreateOriginRequestPolicy. Weitere Informationen zu den Feldern, die Sie in diesem API-Aufruf angeben, finden Sie in Grundlegendes zu Ursprungsanforderungsrichtlinien und in der API-Referenzdokumentation für Ihr AWS SDK oder einen anderen API-Client.

Nach der Erstellung einer Ursprungsanforderungsrichtlinie können Sie diese mit einem der folgenden API-Aufrufe an ein Cacheverhalten anfügen:

- Um es an ein Cache-Verhalten in einer vorhandenen Distribution anzuhängen, verwenden Sie UpdateDistribution.
- Um es an ein Cache-Verhalten in einer neuen Distribution anzuhängen, verwenden Sie CreateDistribution.

Geben Sie für beide API-Aufrufe die ID der Ursprungsanforderungsrichtlinie im Feld `OriginRequestPolicyId` innerhalb eines Cache-Verhaltens an. Weitere Informationen zu den anderen Feldern, die Sie in diesen API-Aufrufen angeben, finden Sie in Referenz für alle Distributionseinstellungen und in der API-Referenzdokumentation für Ihr AWS SDK oder einen anderen API-Client.

Verwenden verwalteter Ursprungsanforderungsrichtlinien

CloudFront bietet eine Reihe von Richtlinien für verwaltete Herkunftsanfragen, die Sie an jedes Cache-Verhalten Ihrer Distribution anhängen können. Bei Verwendung einer verwalteten Ursprungsanforderungsrichtlinie müssen Sie keine eigene Ursprungsanforderungsrichtlinie schreiben oder verwalten. Die verwalteten Richtlinien verwenden Einstellungen, die für bestimmte Anwendungsfälle optimiert sind.

Wenn Sie eine verwaltete Ursprungsanforderungsrichtlinie verwenden möchten, fügen Sie sie einem Cache-Verhalten in Ihrer Distribution zu. Der Prozess ist der gleiche wie beim Erstellen einer Ursprungsanforderungsrichtlinie. Anstatt jedoch eine neue zu erstellen, fügen Sie einfach eine der verwalteten Ursprungsanforderungsrichtlinien an. Sie fügen die Richtlinie entweder nach Name (über die Konsole) oder nach ID an (über die AWS CLI oder SDKs). Die Namen und IDs sind im folgenden Abschnitt aufgeführt.

Weitere Informationen finden Sie unter Erstellen von Ursprungsanforderungsrichtlinien.

In den folgenden Themen werden die verwalteten Ursprungsanforderungsrichtlinien beschrieben, die Sie verwenden können.

Themen

- AllViewer
- AllViewerAndCloudFrontHeaders-2022-06
- AllViewerExceptHostHeader
- CORS-CustomOrigin
- CORS-S3Origin
- Elemental-MediaTailor-PersonalizedManifests
- HostHeaderOnly
- UserAgentRefererHeaders

AllViewer

Diese Richtlinie in der CloudFront Konsole anzeigen

Diese Richtlinie enthält alle Werte (Header, Cookies und Abfragezeichenfolgen) aus der Viewer-Anforderung.

Wenn Sie CloudFormation die AWS CLI oder die CloudFront API verwenden, lautet die ID für diese Richtlinie:

`216adef6-5c7f-47e4-b989-5492eafa07d3`

Diese Richtlinie hat folgende Einstellungen:

- Header, die in Ursprungsanfragen enthalten sind: Alle Header in der Betrachteranfrage
- Cookies, die in Ursprungsanfragen enthalten sind: Alle
- Abfragezeichenfolgen, die in Ursprungsanforderungen enthalten sind: Alle

AllViewerAndCloudFrontHeaders-2022-06

Diese Richtlinie in der CloudFront Konsole anzeigen

Diese Richtlinie umfasst alle Werte (Header, Cookies und Abfragezeichenfolgen) aus der Viewer-Anfrage sowie alle CloudFront Header, die bis Juni 2022 veröffentlicht wurden (CloudFront Header, die nach Juni 2022 veröffentlicht wurden, sind nicht enthalten).

Wenn Sie die oder die CloudFront API verwenden CloudFormation AWS CLI, lautet die ID für diese Richtlinie:

`33f36d7e-f396-46d9-90e0-52428a34d9dc`

Diese Richtlinie hat folgende Einstellungen:

- In ursprünglichen Anfragen enthaltene Header: Alle Header in der Viewer-Anfrage und die folgenden CloudFront Header:
  - `CloudFront-Forwarded-Proto`
  - `CloudFront-Is-Android-Viewer`
  - `CloudFront-Is-Desktop-Viewer`
  - `CloudFront-Is-IOS-Viewer`

- `CloudFront-Is-Mobile-Viewer`
- `CloudFront-Is-SmartTV-Viewer`
- `CloudFront-Is-Tablet-Viewer`
- `CloudFront-Viewer-Address`
- `CloudFront-Viewer-ASN`
- `CloudFront-Viewer-City`
- `CloudFront-Viewer-Country`
- `CloudFront-Viewer-Country-Name`
- `CloudFront-Viewer-Country-Region`
- `CloudFront-Viewer-Country-Region-Name`
- `CloudFront-Viewer-Http-Version`
- `CloudFront-Viewer-Latitude`
- `CloudFront-Viewer-Longitude`
- `CloudFront-Viewer-Metro-Code`
- `CloudFront-Viewer-Postal-Code`
- `CloudFront-Viewer-Time-Zone`
- `CloudFront-Viewer-TLS`

- Cookies, die in Ursprungsanfragen enthalten sind: Alle
- Abfragezeichenfolgen, die in Ursprungsanforderungen enthalten sind: Alle

AllViewerExceptHostHeader

Diese Richtlinie in der CloudFront Konsole anzeigen

Diese Richtlinie beinhaltet nicht den `Host`-Header der Viewer-Anforderung, aber alle anderen Werte (Header, Cookies und Abfragezeichenfolgen) aus der Viewer-Anforderung.

Diese Richtlinie umfasst auch zusätzliche CloudFront Anforderungsheader für das HTTP-Protokoll, die HTTP-Version, die TLS-Version und alle Header für Gerätetyp und Viewer-Standort.

Diese Richtlinie ist für die Verwendung mit Amazon API Gateway, AWS Lambda Funktions-URL und Lambda Web Function Origins vorgesehen. Bei diesen Ursprüngen wird davon `Host` ausgegangen, dass der Header den Ursprungs-Domainnamen und nicht den Domainnamen der CloudFront

Distribution enthält. Das Weiterleiten des `Host`-Headers von der Viewer-Anforderung an diese Ursprünge kann dazu führen, dass sie nicht funktionieren.

> **Note**
>
> Wenn Sie diese Richtlinie für verwaltete Herkunftsanfragen verwenden, um den `Host` Header des Betrachters zu entfernen, wird der ursprünglichen Anfrage ein neuer `Host` Header mit dem Domainnamen des Ursprungs CloudFront hinzugefügt.

Wenn Sie CloudFormation die AWS CLI oder die CloudFront API verwenden, lautet die ID für diese Richtlinie:

`b689b0a8-53d0-40ab-baf2-68738e2966ac`

Diese Richtlinie hat folgende Einstellungen:

- Header, die in Ursprungsanforderungen enthalten sind: Alle Header in der Viewer-Anforderung mit Ausnahme des `Host`-Headers
- Cookies, die in Ursprungsanfragen enthalten sind: Alle
- Abfragezeichenfolgen, die in Ursprungsanforderungen enthalten sind: Alle

CORS-CustomOrigin

Diese Richtlinie in der CloudFront Konsole anzeigen

Diese Richtlinie enthält den Header, der CORS-Anforderungen (Cross-Origin Resource Sharing) aktiviert, wenn der Ursprung ein benutzerdefinierter Ursprung ist.

Wenn Sie CloudFormation die AWS CLI oder die CloudFront API verwenden, lautet die ID für diese Richtlinie:

`59781a5b-3903-41f3-afcb-af62929ccde1`

Diese Richtlinie hat folgende Einstellungen:

- Header, die in Ursprungsanfragen enthalten sind:
  - `Origin`
- Cookies, die in Ursprungsanfragen enthalten sind: Keine

- Abfragezeichenfolgen, die in Ursprungsanforderungen enthalten sind: Keine

CORS-S3Origin

Diese Richtlinie in der CloudFront Konsole anzeigen

Diese Richtlinie enthält die Header, die CORS-Anforderungen (Cross-Origin Resource Sharing) aktivieren, wenn der Ursprung ein Amazon S3 Bucket ist.

Wenn Sie CloudFormation die AWS CLI oder die CloudFront API verwenden, lautet die ID für diese Richtlinie:

`88a5eaf4-2fd4-4709-b370-b4c650ea3fcf`

Diese Richtlinie hat folgende Einstellungen:

- Header, die in Ursprungsanfragen enthalten sind:
  - `Origin`
  - `Access-Control-Request-Headers`
  - `Access-Control-Request-Method`
- Cookies, die in Ursprungsanfragen enthalten sind: Keine
- Abfragezeichenfolgen, die in Ursprungsanforderungen enthalten sind: Keine

Elemental-MediaTailor-PersonalizedManifests

Diese Richtlinie in der CloudFront Konsole anzeigen

Diese Richtlinie wurde für die Verwendung mit einem Ursprung entwickelt, bei dem es sich um einen AWS Elemental MediaTailor -Endpunkt handelt.

Wenn Sie CloudFormation die AWS CLI oder die CloudFront API verwenden, lautet die ID für diese Richtlinie:

`775133bc-15f2-49f9-abea-afb2e0bf67d2`

Diese Richtlinie hat folgende Einstellungen:

- Header, die in Ursprungsanfragen enthalten sind:
  - `Origin`

  - `Access-Control-Request-Headers`
  - `Access-Control-Request-Method`
  - `User-Agent`
  - `X-Forwarded-For`
- Cookies, die in Ursprungsanfragen enthalten sind: Keine
- Abfragezeichenfolgen, die in Ursprungsanforderungen enthalten sind: Alle

HostHeaderOnly

Diese Richtlinie in der CloudFront Konsole anzeigen

Diese Richtlinie enthält nur den `Host`-Header der ursprünglichen Anforderung. Sie enthält keine Abfragezeichenfolgen oder Cookies.

Wenn Sie CloudFormation die AWS CLI oder die CloudFront API verwenden, lautet die ID für diese Richtlinie:

`bf0718e1-ba1e-49d1-88b1-f726733018ae`

Diese Richtlinie hat folgende Einstellungen:

- Header, die in Ursprungsanforderungen enthalten sind: Host
- Cookies, die in Ursprungsanfragen enthalten sind: Keine
- Abfragezeichenfolgen, die in Ursprungsanforderungen enthalten sind: Keine

UserAgentRefererHeaders

Diese Richtlinie in der CloudFront Konsole anzeigen

Diese Richtlinie enthält nur die Header `User-Agent` und `Referer`. Sie enthält keine Abfragezeichenfolgen oder Cookies.

Wenn Sie CloudFormation die AWS CLI oder die CloudFront API verwenden, lautet die ID für diese Richtlinie:

`acba4595-bd28-49b8-b9fe-13317c0390fa`

Diese Richtlinie hat folgende Einstellungen:

- Header, die in Ursprungsanfragen enthalten sind:
  - `User-Agent`
  - `Referer`
- Cookies, die in Ursprungsanfragen enthalten sind: Keine
- Abfragezeichenfolgen, die in Ursprungsanforderungen enthalten sind: Keine

CloudFront Anforderungsheader hinzufügen

Sie können so konfigurieren CloudFront , dass den Anfragen, die von Zuschauern CloudFront empfangen werden, spezifische HTTP-Header hinzugefügt und an Ihre Origin- oder Edge-Funktion weitergeleitet werden. Die Werte dieser HTTP-Header basieren auf den Eigenschaften des Viewers oder der Viewer-Anforderung. Die Header enthalten Informationen über den Gerätetyp, die IP-Adresse, den geografischen Standort, das Anforderungsprotokoll (HTTP oder HTTPS), die HTTP-Version, die TLS-Verbindungsdetails, den Fingerabdruck und den JA3 Fingerabdruck des Betrachters. JA4 Sie können das Cache-Verhalten Ihrer Distribution auch so konfigurieren, dass WebSocket Header weitergeleitet werden. Weitere Informationen finden Sie unter WebSockets Mit CloudFront Distributionen verwenden.

Mit diesen Headern kann Ihr Ursprung oder Ihre Edge-Funktion Informationen über den Viewer abrufen, sodass Sie keinen eigenen Code schreiben müssen, um diese Informationen zu ermitteln. Wenn Ihr Origin auf der Grundlage der Informationen in diesen Headern unterschiedliche Antworten zurückgibt, können Sie diese in den Cache-Schlüssel aufnehmen, sodass die Antworten separat CloudFront zwischengespeichert werden. Beispielsweise könnte der Ursprung basierend auf dem Land, in dem sich der Viewer befindet, mit Inhalten in einer spezifischen Sprache antworten oder er könnte mit Inhalten antworten, die auf einen spezifischen Gerätetyp zugeschnitten sind. Der Ursprung könnte diese Header auch in Protokolldateien schreiben, mit denen Sie Informationen über die Standorte und Gerätetypen der Viewer und vieles mehr ermitteln können.

Um diese Header in den Cache-Schlüssel einzufügen, verwenden Sie eine Cache-Richtlinie. Weitere Informationen erhalten Sie unter Steuern des Cache-Schlüssels mit einer Richtlinie und the section called “Grundlegendes zum Cache-Schlüssel”.

Wenn Sie Header am Ursprung empfangen, diese aber nicht in den Cache-Schlüssel einschließen möchten, verwenden Sie eine Ursprungsanforderungsrichtlinie. Weitere Informationen finden Sie unter Steuern von Ursprungsanforderungen anhand einer Richtlinie.

Themen

- Gerätetyp-Header
- Viewer-Standort-Header
- Header zur Bestimmung der Header-Struktur des Viewers
- TLS-bezogene Header
- Andere Header CloudFront

Gerätetyp-Header

Mit den folgenden Headern können Sie den Gerätetyp des Viewers ermitteln. CloudFront Legt den Wert dieser `User-Agent` Header basierend auf dem Wert des Headers auf oder fest. `true` `false` Wenn ein Gerät in mehr als eine Kategorie fällt, können mehrere Werte `true` sein. CloudFront Legt z. B. bei einigen Tablet-Geräten `CloudFront-Is-Mobile-Viewer` sowohl als auch `CloudFront-Is-Tablet-Viewer` auf `true` fest.

- `CloudFront-Is-Android-Viewer`— `true` Wird auf „Wenn“ gesetzt, wird CloudFront bestimmt, dass es sich bei dem Viewer um ein Gerät mit dem Betriebssystem Android handelt.
- `CloudFront-Is-Desktop-Viewer`— Auf „`true`Wenn“ eingestellt, wird CloudFront bestimmt, dass es sich bei dem Viewer um ein Desktop-Gerät handelt.
- `CloudFront-Is-IOS-Viewer`— Auf „`true`Wenn“ gesetzt, wird CloudFront bestimmt, dass es sich beim Viewer um ein Gerät mit einem mobilen Apple-Betriebssystem wie iPhone, iPod touch und einigen iPad-Geräten handelt.
- `CloudFront-Is-Mobile-Viewer`— Auf „`true`Wenn“ eingestellt, wird CloudFront bestimmt, dass es sich bei dem Betrachter um ein Mobilgerät handelt.
- `CloudFront-Is-SmartTV-Viewer`— Auf „`true`Wenn“ eingestellt, wird CloudFront bestimmt, dass es sich bei dem Zuschauer um ein Smart-TV handelt.
- `CloudFront-Is-Tablet-Viewer`— Auf „`true`Wenn“ eingestellt, wird CloudFront bestimmt, dass es sich bei dem Zuschauer um ein Tablet handelt.

Viewer-Standort-Header

Sie können die folgenden Header hinzufügen, um den Standort des Betrachters zu bestimmen. CloudFront bestimmt die Werte für diese Header auf der Grundlage der IP-Adresse des Betrachters. Bei Nicht-ASCII-Zeichen in den Werten dieser Header wird das Zeichen gemäß Abschnitt 1.2 von RFC CloudFront 3986 mit Prozent codiert.

- `CloudFront-Viewer-Address` – Enthält die IP-Adresse des Viewers und den Quell-Port der Anforderung. Der Header-Wert `198.51.100.10:46532` bedeutet beispielsweise, dass die IP-Adresse des Viewers 198.51.100.10 lautet und der Quell-Port der Anforderung 46532 ist.
- `CloudFront-Viewer-ASN` – Enthält die autonome Systemnummer (ASN) des Viewers.

> **Note**
>
> `CloudFront-Viewer-Address` und `CloudFront-Viewer-ASN` können in einer Ursprungsanforderungsrichtlinie, aber nicht in einer Cache-Richtlinie hinzugefügt werden.

- `CloudFront-Viewer-Country` – Enthält den zweistelligen Ländercode für das Land des Viewers. Die Liste der Ländercodes finden Sie unter ISO 3166-1 alpha-2.
- `CloudFront-Viewer-City` – Enthält den Namen der Stadt des Viewers.

Wenn Sie die folgenden Header hinzufügen, werden sie auf alle Anfragen CloudFront angewendet, mit Ausnahme derjenigen, die aus dem Netzwerk stammen: AWS

- `CloudFront-Viewer-Country-Name` – Enthält den Namen des Landes des Viewers.
- `CloudFront-Viewer-Country-Region` – Enthält einen Code (bis zu drei Zeichen), der die Region des Viewers darstellt. Die Region ist die Unterteilung der ersten Ebene (die breiteste oder am wenigsten spezifische) des Codes ISO 3166-2.
- `CloudFront-Viewer-Country-Region-Name` – Enthält den Namen der Region des Viewers. Die Region ist die Unterteilung der ersten Ebene (die breiteste oder am wenigsten spezifische) des Codes ISO 3166-2.
- `CloudFront-Viewer-Latitude` – Enthält den ungefähren Breitengrad des Viewers.
- `CloudFront-Viewer-Longitude` – Enthält den ungefähren Längengrad des Viewers.
- `CloudFront-Viewer-Metro-Code` – Enthält den Metro-Code des Viewers. Dieser wird nur verwendet, wenn sich der Viewer in den Vereinigten Staaten befindet.
- `CloudFront-Viewer-Postal-Code` – Enthält die Postleitzahl des Viewers.
- `CloudFront-Viewer-Time-Zone` Enthält die Zeitzone des Viewers im IANA-Zeitzonen-Datenbankformat (z. B. `America/Los_Angeles`).

> **Note**
>
> `CloudFront-Viewer-City`, `CloudFront-Viewer-Metro-Code` und `CloudFront-Viewer-Postal-Code` sind möglicherweise nicht für jede IP-Adresse verfügbar. Einige IP-Adressen können nicht mit ausreichender Genauigkeit geolokalisiert werden, um diese Informationen zu erhalten.

Header zur Bestimmung der Header-Struktur des Viewers

Sie können jetzt die folgenden Header hinzufügen, um den Viewer leichter anhand der gesendeten Header identifizieren zu können. So können beispielsweise verschiedene Browser HTTP-Header in einer bestimmten Reihenfolge senden. Wenn der im `User-Agent`-Header angegebene Browser nicht mit der erwarteten Header-Reihenfolge dieses Browsers übereinstimmt, können Sie die Anforderung ablehnen. Auch wenn der `CloudFront-Viewer-Header-Count`-Wert nicht mit der Anzahl der Header in `CloudFront-Viewer-Header-Order` übereinstimmt, können Sie die Anforderung ablehnen.

- `CloudFront-Viewer-Header-Order` – Enthält die Header-Namen des Viewers in der angeforderten Reihenfolge, getrennt durch einen Doppelpunkt. Beispiel: `CloudFront-Viewer-Header-Order: Host:User-Agent:Accept:Accept-Encoding`. Header, die das Zeichenlimit von 7 680 überschreiten, werden abgeschnitten.
- `CloudFront-Viewer-Header-Count` – Enthält die Gesamtzahl der Header des Viewers.

TLS-bezogene Header

Sie können die folgenden Header hinzufügen, um den Fingerabdruck, den JA3 Fingerabdruck und die TLS-Verbindungsdetails JA4 des Betrachters zu ermitteln:

- `CloudFront-Viewer-JA3-Fingerprint`— Enthält den JA3 Fingerabdruck des Betrachters. Anhand des JA3 Fingerabdrucks können Sie feststellen, ob die Anfrage von einem bekannten Client stammt, ob es sich dabei um Malware oder einen böswilligen Bot oder um eine erwartete Anwendung handelt (auf der Zulassungsliste).
- `CloudFront-Viewer-JA4-Fingerprint`— Enthält den JA4 Fingerabdruck des Betrachters. Ähnlich wie beim JA3 Fingerabdruck können Sie anhand des JA4 Fingerabdrucks feststellen, ob die Anfrage von einem bekannten Client stammt, ob es sich dabei um Malware oder einen böswilligen Bot oder um eine erwartete Anwendung handelt (auf der Zulassungsliste). Sie können

den Fingerprint verwenden, um eine Datenbank mit bekannten guten und schlechten Akteuren zu erstellen, die Sie bei der Überprüfung von HTTP-Anforderungen verwenden können. Anschließend können Sie den Header-Wert auf Ihren Anwendungs-Webservern oder in Ihren Lambda @Edge und CloudFront Functions überprüfen, um den Header-Wert mit einer Liste bekannter Malware-Fingerabdrücke zu vergleichen, um böswillige Clients zu blockieren.

- `CloudFront-Viewer-TLS`— Enthält die SSL/TLS Version, die Chiffre und Informationen über den SSL/TLS Handshake, der für die Verbindung zwischen dem Betrachter und verwendet wurde. CloudFront Der Header-Wert weist das folgende Format auf:

  ```
  SSL/TLS_version:cipher:handshake_information
  ```

  Der *`handshake_information`*-Header kann einen der folgenden Werte enthalten:

  - `fullHandshake`— Für die Sitzung wurde ein vollständiger Handshake durchgeführt. SSL/TLS
  - `sessionResumed`— Eine vorherige SSL/TLS Sitzung wurde wieder aufgenommen.
  - `connectionReused`— Eine vorherige SSL/TLS Verbindung wurde wiederverwendet.

  Im Folgenden finden Sie einige Beispielwerte für diesen Header:

  ```
  TLSv1.3:TLS_AES_128_GCM_SHA256:sessionResumed
  ```

  ```
  TLSv1.2:ECDHE-ECDSA-AES128-GCM-SHA256:connectionReused
  ```

  ```
  TLSv1.1:ECDHE-RSA-AES128-SHA256:fullHandshake
  ```

  ```
  TLSv1:ECDHE-RSA-AES256-SHA:fullHandshake
  ```

  Eine vollständige Liste der möglichen SSL/TLS Versionen und Chiffren, die in diesem Header-Wert enthalten sein können, finden Sie unter. the section called “Unterstützte Protokolle und Chiffren zwischen Viewern und CloudFront”

> **Hinweise**
>
> - Die JA4 Fingerabdrücke JA3 und die Fingerabdrücke werden aus dem Paket abgeleitet. SSL/TLS `Client Hello` Sie sind nur für HTTPS-Anforderungen vorhanden.

- Dese TLS-bezogenen Header können Sie zu einer Ursprungsanforderungsrichtlinie hinzufügen, aber nicht zu einer Cache-Richtlinie.

Andere Header CloudFront

Sie können die folgenden Header hinzufügen, um den ursprünglichen Anforderungs-URI, die Abfragezeichenfolgeparameter und Werte der ursprünglichen Anforderung, das Protokoll und die Version zu ermitteln:

- `CloudFront-Error-Uri` – enthält den ursprünglichen Anforderung-URI, der vom Viewer empfangen wurde.
- `CloudFront-Error-Args` – enthält die Abfragezeichenfolgeparameter und Werte der ursprünglichen Anforderung.
- `CloudFront-Forwarded-Proto` – Enthält das Protokoll der Anforderung des Viewers (HTTP oder HTTPS).
- `CloudFront-Viewer-Http-Version` – Enthält die HTTP-Version der Anforderung des Viewers.

Erläuterung des Zusammenwirkens von Ursprungsanforderungsrichtlinien und Cache-Richtlinien

Sie können eine CloudFront Richtlinie für Quellanfragen verwenden, um die Anfragen zu kontrollieren, die an den Absender CloudFront gesendet werden. Diese Anfragen werden als Ursprungsanfragen bezeichnet. Um eine Ursprungsanforderungsrichtlinie verwenden zu können, müssen Sie der Cache-Verhaltensweise eine Cache-Richtlinie anhängen. Sie können eine Ursprungsanforderungsrichtlinie in einem Cacheverhalten nicht ohne eine Cache-Richtlinie verwenden. Weitere Informationen finden Sie unter Steuern von Ursprungsanforderungen anhand einer Richtlinie.

Richtlinien für Ursprungsanfragen und Cache-Richtlinien arbeiten zusammen, um die Werte zu bestimmen, die in den Quellanforderungen CloudFront enthalten sind. Alle URL-Abfragezeichenfolgen, HTTP-Header und Cookies, die Sie im Cache-Schlüssel (über eine Cache-Richtlinie) festlegen, werden automatisch auch in Ursprungsanforderungen eingefügt. Alle zusätzlichen Abfragezeichenfolgen, Header und Cookies, die Sie in einer Ursprungsanforderungsrichtlinie angeben, sind auch in Ursprungsanforderungen enthalten (nicht jedoch im Cache-Schlüssel).

Ursprungsanforderungs- und Cache-Richtlinien verfügen über Einstellungen, die anscheinend miteinander in Konflikt stehen. Beispielsweise kann eine Richtlinie bestimmte Werte zulassen, während eine andere sie blockiert. In der folgenden Tabelle wird erklärt, welche Werte in Ursprungsanfragen CloudFront enthalten sind, wenn Sie die Einstellungen einer Quell-Anforderungsrichtlinie und einer Cache-Richtlinie zusammen verwenden. Diese Einstellungen gelten im Allgemeinen für alle Arten von Werten: Abfragezeichenfolgen, Header und Cookies. In einer Cache-Richtlinie können Sie jedoch nicht alle Header angeben oder eine Header-Blockliste verwenden.

	Ursprungsanforderungsrichtlinie			
	Keine	Alle	Zulassungsliste	Blockierliste
Cache-Richtlinie				
Keine	In der Ursprungsanforderung sind keine Werte aus der Viewer-Anforderung enthalten, mit Ausnahme der Standardwerte, die in jeder Ursprungsanforderung enthalten sind. Weitere Informationen finden Sie unter Steuern von Ursprungsanforderungen anhand einer Richtlinie.	Alle Werte aus der Viewer-Anforderung sind in der Ursprungsanforderung enthalten.	Nur die in der Ursprungsanforderungsrichtlinie angegebenen Werte sind in der Ursprungsanforderung enthalten.	Alle Werte aus der Viewer-Anforderung mit Ausnahme der Werte, die in der Ursprungsanforderungsrichtlinie angegeben sind, sind in der Ursprungsanforderung enthalten.
Alle	Alle Abfragezeichenfolgen und Cookies aus	Alle Werte aus der Viewer-Anforderung sind	Alle Abfragezeichenfolgen und Cookies aus	Alle Abfragezeichenfolgen und Cookies aus

	Ursprungsanforderungsrichtlinie			
	Keine	Alle	Zulassungsliste	Blockierliste
Hinweis: Sie können nicht alle Header in einer Cache-Richtlinie angeben.	der Viewer-Anforderung sind in der Ursprungsanforderung enthalten.	in der Ursprungsanforderung enthalten.	der Viewer-Anforderung sowie alle Header, die in der Ursprungsanforderungsrichtlinie angegeben sind, sind in der Ursprungsanforderung enthalten.	der Viewer-Anforderung sind in der Ursprungsanforderung enthalten; auch die, die in der Blockierliste der Ursprungsanforderungsrichtlinie angegeben sind. Die Einstellung der Cache-Richtlinie hat Vorrang vor der Blockierliste der Ursprungsanforderungsrichtlinie.

	Ursprungsanforderungsrichtlinie			
	Keine	Alle	Zulassungsliste	Blockierliste
Zulassungsliste	Nur die angegebenen Werte aus der Viewer-Anforderung sind in der Ursprungsanforderung enthalten.	Alle Werte aus der Viewer-Anforderung sind in der Ursprungsanforderung enthalten.	Alle in der Cache-Richtlinie oder der Ursprungsanforderungsrichtlinie angegebenen Werte sind in der Ursprungsanforderung enthalten.	Die in der Cache-Richtlinie angegebenen Werte sind in der Ursprungsanforderung enthalten, auch wenn dieselben Werte in der Blockierliste der Ursprungsanforderungsrichtlinie angegeben sind. Die Zulassungsliste der Cache-Richtlinie hat Vorrang vor der Blockierliste der Ursprungsanforderungsrichtlinie.

	Ursprungsanforderungsrichtlinie			
	Keine	Alle	Zulassungsliste	Blockierliste
Blockierliste Hinweis: In einer Blockierliste für Cache-Richtlinien können Sie keine Header angeben.	Alle Abfragezeichenfolgen und Cookies aus der Viewer-Anforderung mit Ausnahme der angegebenen sind in der Ursprungsanforderung enthalten.	Alle Werte aus der Viewer-Anforderung sind in der Ursprungsanforderung enthalten.	Die in der Ursprungsanforderungsrichtlinie angegebenen Werte sind in der Ursprungsanforderung enthalten, auch wenn dieselben Werte in der Blockierliste der Cache-Richtlinie angegeben sind. Die Zulassungsliste der Ursprungsanforderungsrichtlinie hat Vorrang vor der Blockierliste der Cache-Richtlinie.	Alle Werte aus der Viewer-Anforderung mit Ausnahme der Werte, die in der Cache-Richtlinie oder der Ursprungsanforderungsrichtlinie angegeben sind, sind in der Ursprungsanforderung enthalten.

Hinzufügen oder Entfernen von HTTP-Headern in CloudFront Antworten mit einer Richtlinie

Sie können so konfigurieren CloudFront , dass die HTTP-Header in den Antworten, die an Zuschauer (Webbrowser und andere Clients) gesendet werden, geändert werden. CloudFront kann Header entfernen, die es vom Ursprung erhalten hat, oder der Antwort Header hinzufügen, bevor die Antwort an die Zuschauer gesendet wird. Für diese Änderungen müssen Sie weder Code schreiben noch den Ursprung ändern.

Du kannst beispielsweise Überschriften wie `X-Powered-By` und entfernen, `Vary` sodass diese Überschriften CloudFront nicht in den Antworten enthalten sind, die an Zuschauer gesendet werden. Sie können auch HTTP-Header wie beispielsweise die folgenden hinzufügen:

- `Cache-Control`-Header zur Steuerung des Browser-Cachings
- `Access-Control-Allow-Origin`-Header zum Ermöglichen von Cross-Origin Resource Sharing (CORS). Sie können auch andere CORS-Header hinzufügen.
- Ein Satz von allgemeinen Sicherheits-Headern wie beispielsweise `Strict-Transport-Security`, `Content-Security-Policy` und `X-Frame-Options`
- Eine `Server-Timing` Kopfzeile, in der Informationen zur Leistung und zur Weiterleitung der Anfrage und der Antwort angezeigt werden. CloudFront

Um die Header anzugeben, die in HTTP-Antworten CloudFront hinzugefügt oder entfernt werden, verwenden Sie eine Richtlinie für Antwortheader. Sie fügen eine Richtlinie für Antwortheader an ein oder mehrere Cache-Verhaltensweisen an. CloudFront ändert die Header in den Antworten, die auf Anfragen gesendet werden, die dem Cacheverhalten entsprechen. CloudFrontändert die Header in den Antworten, die es aus dem Cache bereitstellt, und diejenigen, die es vom Ursprung aus weiterleitet. Wenn die ursprüngliche Antwort einen oder mehrere der Header enthält, die in einer Antwort-Header-Richtlinie hinzugefügt wurden, kann die Richtlinie angeben, ob der Header CloudFront verwendet wird, den sie vom Ursprung erhalten hat, oder ob dieser Header mit dem Header in der Antwort-Header-Richtlinie überschrieben wird.

> **Note**
>
> Wenn Sie Ihren Antwort-Header-Richtlinien Header hinzufügen, die das Browser-Caching steuern`Cache-Control`, CloudFront fügt diese Header beispielsweise nur der

Zuschauerantwort hinzu. Diese Header haben keinen Einfluss darauf, wie das angeforderte Objekt CloudFront zwischengespeichert wird.

CloudFront stellt vordefinierte Richtlinien für Antwortheader, sogenannte verwaltete Richtlinien, für allgemeine Anwendungsfälle bereit. Sie können diese verwalteten Richtlinien verwenden oder eigene Richtlinien erstellen. Sie können eine einzelne Response-Header-Richtlinie an mehrere Cache-Verhaltensweisen in mehreren Distributionen in Ihrem anhängen. AWS-Konto

Weitere Informationen finden Sie hier:

Themen

- Grundlegendes zu Antwortheader-Richtlinien
- Erstellen von Antwortheader-Richtlinien
- Verwenden verwalteter Antwortheader-Richtlinien

Grundlegendes zu Antwortheader-Richtlinien

Sie können eine Richtlinie für Antwort-Header verwenden, um die HTTP-Header festzulegen, die Amazon CloudFront entfernt oder in Antworten, die es an Zuschauer sendet, hinzufügt. Weitere Informationen zu Antwort-Header-Richtlinien und zu Gründen für ihre Verwendung finden Sie unter Hinzufügen oder Entfernen von Antwortheadern mit einer Richtlinie.

In den folgenden Themen werden die Einstellungen in einer Antwort-Header-Richtlinie erläutert. Die Einstellungen sind in Kategorien unterteilt, die in den folgenden Themen behandelt werden.

Themen

- Richtliniendetails (Metadaten)
- CORS-Header
- Sicherheits-Header
- Benutzerdefinierte Header
- Entfernen von Headern
- Server-Timing Kopfzeile

Richtliniendetails (Metadaten)

Die Einstellungen für Richtliniendetails enthalten Metadaten zu einer Antwort-Header-Richtlinie.

- Name – Ein Name zur Identifizierung der Antwort-Header-Richtlinie. Verwenden Sie den Namen in der Konsole, um die Richtlinie einem Cache-Verhalten anzufügen.
- Beschreibung (optional) – Ein Kommentar zur Beschreibung der Antwort-Header-Richtlinie. Dies ist optional, kann Ihnen jedoch helfen, den Zweck der Richtlinie zu identifizieren.

CORS-Header

Mit den Cross-Origin Resource Sharing (CORS)-Einstellungen können Sie CORS-Header in einer Antwort-Header-Richtlinie hinzufügen und konfigurieren.

Bei dieser Liste liegt der Fokus darauf, wie Einstellungen und gültige Werte in einer Antwort-Header-Richtlinie angegeben werden. Weitere Informationen zu den jeweiligen Headern und deren Verwendung für reale CORS-Anforderungen und -Antworten finden Sie unter Cross-Origin Resource Sharing in den MDN-Webdokumenten und in der CORS-Protokollspezifikation.

Access-Control-Allow-Credentials

Dies ist eine boolesche Einstellung (`true`oder`false`), die bestimmt, ob der `Access-Control-Allow-Credentials` Header in Antworten auf CORS-Anfragen CloudFront hinzugefügt wird. Wenn diese Einstellung auf gesetzt ist`true`, wird der `Access-Control-Allow-Credentials: true` Header in Antworten auf CORS-Anfragen CloudFront hinzugefügt. Andernfalls wird dieser Header CloudFront nicht zu den Antworten hinzugefügt.

Access-Control-Allow-Headers

Gibt die Header-Namen an, die als Werte für den `Access-Control-Allow-Headers` Header in Antworten auf CORS-Preflight-Anfragen CloudFront verwendet werden. Zu den gültigen Werten für diese Einstellung gehören HTTP-Header-Namen oder das Platzhalterzeichen (`*`), das anzeigt, dass alle Header zulässig sind.

> **Note**
>
> Der `Authorization`-Header darf keinen Platzhalter verwenden und muss explizit aufgeführt werden.

Beispiele für die gültige Verwendung des Platzhalterzeichens

Beispiel	Übereinstimmung mit	Keine Übereinstimmung mit
`x-amz-*`	`x-amz-test` `x-amz-`	`x-amz`
`x-*-amz`	`x-test-amz` `x--amz`	
`*`	Alle Header außer `Authorization`	`Authorization`

Access-Control-Allow-Methods

Gibt die HTTP-Methoden an, die als Werte für den `Access-Control-Allow-Methods` Header in Antworten auf CORS-Preflight-Anfragen CloudFront verwendet werden. Gültige Werte sind `GET`, `DELETE`, `HEAD`, `OPTIONS`, `PATCH`, `POST`, `PUT` und `ALL`. `ALL` ist ein spezieller Wert, der alle aufgelisteten HTTP-Methoden enthält.

Access-Control-Allow-Origin

Gibt die Werte an, die im `Access-Control-Allow-Origin` Antwortheader verwendet werden CloudFront können. Zu den gültigen Werten für diese Einstellung gehören ein spezifischer Ursprung (z. B. `http://www.example.com`) oder das Platzhalterzeichen (`*`), das angibt, dass alle Ursprünge zulässig sind.

Hinweise

- Das Platzhalterzeichen (`*`) ist als äußerster linker Teil der Subdomain (`*.example.org`) zulässig.
- Das Platzhalterzeichen (`*`) ist an folgenden Stellen nicht zulässig:
  - Top-level Domänen (`example.*`)
  - rechts neben Subdomains (`test.*.example.org`) oder innerhalb beliebiger Subdomains (`*test.example.org`)
  - Innerhalb von Begriffen (`exa*mple.org`)

Beispiele für die Verwendung des Platzhalterzeichens finden Sie in der folgenden Tabelle.

Beispiel	Übereinstimmung mit	Keine Übereinstimmung mit

`http://*.example.org`	`http://www.example.org`
	
	`http://test.example.org`
		`https``://test.example.org`
		
		`https``://test.example.org:123`
		
		`http://test.example.org:123`

`*.example.org`	`test.example.org`
	
	`test.test.example.org`
	
	`.example.org`
	
	`http://test.example.org`
	
	`https://test.example.org`
		`http://test.example.org:123`
		
		`https://test.example.org:123`

`example.org`	`http://example.org`
	
	`https://example.org`	

`http://example.org`		**`https`**`://example.org` `http://example.org:123`

`http://example.org:*`	`http://example.org:123`
	
	`http://example.org`	

Beispiel	Übereinstimmung mit	Keine Übereinstimmung mit

`http://example.org:1*3`	`http://example.org:123`
	
	`http://example.org:1893`
	
	`http://example.org:13`	

`*.example.org:1*`	`test.example.org:123`	

Access-Control-Expose-Headers

Gibt die Header-Namen an, die als Werte für den `Access-Control-Expose-Headers` Header in Antworten auf CORS-Anfragen CloudFront verwendet werden. Zu den gültigen Werten für diese Einstellung gehören HTTP-Header-Namen oder das Platzhalterzeichen (`*`).

Access-Control-Max-Age

Eine Anzahl von Sekunden, die als Wert für den `Access-Control-Max-Age` Header bei Antworten auf CORS-Preflight-Anfragen CloudFront verwendet wird.

Origin override (Ursprungsüberschreibung)

Eine boolesche Einstellung, die bestimmt, wie CloudFront sich verhält, wenn die Antwort vom Ursprung einen der CORS-Header enthält, die auch in der Richtlinie enthalten sind.

- Wenn dieser Wert auf gesetzt ist `true` und die Quellantwort einen CORS-Header enthält, der ebenfalls in der Richtlinie enthalten ist, wird der Antwort der CORS-Header in der Richtlinie CloudFront hinzugefügt. CloudFront sendet dann diese Antwort an den Betrachter. CloudFront ignoriert den Header, den es vom Ursprung erhalten hat.
- Wenn dieser Wert auf gesetzt ist `false` und die Quellantwort einen CORS-Header enthält (unabhängig davon, ob der CORS-Header in der Richtlinie enthalten ist), CloudFront schließt er den CORS-Header ein, den sie vom Ursprung in die Antwort erhalten hat. CloudFront fügt der Antwort, die an den Betrachter gesendet wird, keine CORS-Header in der Richtlinie hinzu.

Sicherheits-Header

Mit den Einstellungen für Sicherheits-Header können Sie mehrere sicherheitsbezogene HTTP-Antwort-Header in einer Antwort-Header-Richtlinie hinzufügen und konfigurieren.

Diese Liste beschreibt, wie Sie Einstellungen und gültige Werte in einer Antwort-Header-Richtlinie angeben können. Weitere Informationen zu diesen Headern und ihrer Verwendung in realen HTTP-Antworten finden Sie unter den Links zu den MDN-Webdokumenten.

Content-Security-Policy

Gibt die Richtlinien für Inhaltssicherheitsrichtlinien an, die als Werte für den `Content-Security-Policy` Antwortheader CloudFront verwendet werden.

Weitere Informationen zu diesem Header und den gültigen Richtlinien finden Sie unter Content-Security-Policy in den MDN-Webdokumenten.

> **Note**
>
> Der `Content-Security-Policy`-Header-Wert ist auf 1783 Zeichen begrenzt.

Referrer-Policy

Gibt die Referrer-Richtlinienrichtlinie an, die als Wert für den `Referrer-Policy` Antwortheader CloudFront verwendet wird. Gültige Werte für diese Einstellung sind `no-referrer`, `no-referrer-when-downgrade`, `origin`, `origin-when-cross-origin`, `same-origin`, `strict-origin`, `strict-origin-when-cross-origin` und `unsafe-url`.

Weitere Informationen zu diesem Header und diesen Richtlinien finden Sie unter Referrer-Policy in den MDN-Webdokumenten.

Strict-Transport-Security

Gibt die Anweisungen und Einstellungen an, die als Wert für den `Strict-Transport-Security` Antwortheader CloudFront verwendet werden. Für diese Einstellung geben Sie separat Folgendes an:

- Eine Anzahl von Sekunden, die als Wert für die `max-age` Direktive dieses Headers CloudFront verwendet wird
- Eine boolesche Einstellung (`true`oder`false`) für`preload`, die bestimmt, ob CloudFront die `preload` Direktive in den Wert dieses Headers aufgenommen wird

- Eine boolesche Einstellung (`true`oder`false`) für`includeSubDomains`, die bestimmt, ob die `includeSubDomains` Direktive in CloudFront den Wert dieses Headers aufgenommen wird

Weitere Informationen zu diesem Header und diesen Richtlinien finden Sie unter Strict-Transport-Security in den MDN-Webdokumenten.

X-Content-Type-Options

Dies ist eine boolesche Einstellung (`true`oder`false`), die bestimmt, ob der Header zu Antworten CloudFront hinzugefügt wird`X-Content-Type-Options`. Wenn diese Einstellung aktiviert ist`true`, wird der `X-Content-Type-Options: nosniff` Header zu den Antworten CloudFront hinzugefügt. Andernfalls CloudFront wird dieser Header nicht hinzugefügt.

Weitere Informationen zu diesem Header finden Sie unter X-Content-Type-Options in den MDN-Webdokumenten.

X-Frame-Options

Gibt die Direktive an, die als Wert für den `X-Frame-Options` Antwortheader CloudFront verwendet wird. Gültige Werte für diese Einstellung sind `DENY` oder `SAMEORIGIN`.

Weitere Informationen zu diesem Header und diesen Richtlinien finden Sie unter X-Frame-Options in den MDN-Webdokumenten.

X-XSS-Protection

Gibt die Direktiven und Einstellungen an, die als Wert für den `X-XSS-Protection` Antwortheader CloudFront verwendet werden. Für diese Einstellung geben Sie separat Folgendes an:

- Die `X-XSS-Protection`-Einstellung `0` (deaktiviert XSS-Filterung) oder `1` (aktiviert XSS-Filterung)
- Eine boolesche Einstellung (`true`oder`false`) für`block`, die bestimmt, ob die `mode=block` Direktive CloudFront in den Wert für diesen Header aufgenommen wird
- Ein Berichts-URI, der bestimmt, ob CloudFront die `report=`*`reporting URI`* Direktive in den Wert für diesen Header aufgenommen wird

Sie können `true` für `block` oder einen Berichts-URI angeben. Sie können jedoch nicht beides zusammen angeben. Weitere Informationen zu diesem Header und diesen Richtlinien finden Sie unter X-XSS-Protection in den MDN-Webdokumenten.

Origin override (Ursprungsüberschreibung)

Jede dieser Sicherheits-Header-Einstellungen enthält eine boolesche Einstellung (`true`oder`false`), die bestimmt, wie CloudFront sich verhält, wenn die Antwort vom Ursprung diesen Header enthält.

Wenn diese Einstellung auf gesetzt ist `true` und die Quellantwort den Header enthält, wird der Header in der Richtlinie zu der Antwort CloudFront hinzugefügt, die an den Viewer gesendet wird. Der vom Ursprung empfangene Header wird dabei ignoriert.

Wenn diese Einstellung auf gesetzt ist `false` und die ursprüngliche Antwort den Header enthält, CloudFront schließt sie den Header, den sie vom Ursprung erhalten hat, in die Antwort ein, die sie an den Viewer sendet.

Wenn die ursprüngliche Antwort den Header nicht enthält, wird der Antwort, die an den Betrachter gesendet wird, den Header in der Richtlinie CloudFront hinzugefügt. CloudFront tut dies, wenn diese Einstellung auf `true` oder gesetzt ist`false`.

Benutzerdefinierte Header

Sie können benutzerdefinierte Header-Einstellungen verwenden, um benutzerdefinierte HTTP-Header in einer Response-Header-Richtlinie hinzuzufügen und zu konfigurieren. CloudFront fügt diese Header zu jeder Antwort hinzu, die an die Zuschauer zurückgegeben wird. Bei benutzerdefinierten Headern geben Sie auch den Wert für den jeweiligen Header an, auch wenn die Angabe eines Werts optional ist. Dies liegt daran, dass ein Antwortheader ohne Wert hinzugefügt werden CloudFront kann.

Jeder benutzerdefinierte Header verfügt zudem über eine eigene Einstellung für Origin override (Ursprungsüberschreibung):

- Wenn diese Einstellung auf gesetzt ist `true` und die Quellantwort den benutzerdefinierten Header enthält, der in der Richtlinie enthalten ist, wird der benutzerdefinierte Header in der Richtlinie zu der Antwort CloudFront hinzugefügt, die an den Viewer gesendet wird. Der vom Ursprung empfangene Header wird dabei ignoriert.
- Wenn diese Einstellung aktiviert ist `false` und die Quellantwort den benutzerdefinierten Header enthält, der in der Richtlinie enthalten ist, CloudFront schließt sie den benutzerdefinierten Header, den sie vom Ursprung erhalten hat, in die Antwort ein, die an den Betrachter gesendet wird.

- Wenn die ursprüngliche Antwort nicht den benutzerdefinierten Header enthält, der in der Richtlinie enthalten ist, CloudFront fügt der an den Betrachter gesendeten Antwort den benutzerdefinierten Header in der Richtlinie hinzu. CloudFront tut dies, wenn diese Einstellung auf `true` oder gesetzt ist`false`.

Entfernen von Headern

Sie können Kopfzeilen angeben, die Sie aus den Antworten entfernen CloudFront möchten, die vom Absender empfangen werden, sodass die Kopfzeilen nicht in den Antworten enthalten sind, die an Zuschauer CloudFront gesendet werden. CloudFront entfernt die Header aus jeder Antwort, die an Zuschauer gesendet wird, unabhängig davon, ob die Objekte aus dem CloudFront Cache oder vom Ursprung aus bereitgestellt werden. Beispielsweise können Sie Header entfernen, die für Browser nutzlos sind, wie z. B. `X-Powered-By` oder`Vary`, sodass diese Header aus den Antworten CloudFront entfernt werden, die an Zuschauer gesendet werden.

Wenn Sie mithilfe einer Richtlinie für Antwortheader zu entfernende Header angeben, werden die Header zuerst CloudFront entfernt und dann alle Header hinzugefügt, die in anderen Abschnitten der Richtlinie für Antwortheader angegeben sind (CORS-Header, Sicherheitsheader, benutzerdefinierte Header usw.). Wenn Sie einen Header angeben, der entfernt werden soll, denselben Header aber auch in einem anderen Abschnitt der Richtlinie hinzufügen, wird der Header in die Antworten aufgenommen, die an CloudFront die Zuschauer gesendet werden.

> **Note**
>
> Du kannst eine Richtlinie für Antwort-Header verwenden, um die `Server` `Date` Und-Header zu entfernen, die vom Absender CloudFront empfangen wurden, sodass diese Header (so, wie sie vom Absender empfangen wurden) nicht in den Antworten enthalten sind, die an Zuschauer CloudFront gesendet werden. In diesem Fall wird den Antworten, die an Zuschauer gesendet werden, jedoch eine eigene Version dieser Header CloudFront hinzugefügt. Für den `Server` Header, der CloudFront hinzugefügt wird, lautet `CloudFront` der Wert des Headers.

Header, die Sie nicht entfernen können

Sie können die folgenden Header nicht mithilfe einer Antwort-Header-Richtlinie entfernen. Wenn Sie diese Header im Abschnitt Remove headers (Header entfernen) einer Antwort-Header-Richtlinie

(`ResponseHeadersPolicyRemoveHeadersConfig` in der API) angeben, erhalten Sie eine Fehlermeldung.

- `Connection`
- `Content-Encoding`
- `Content-Length`
- `Expect`
- `Host`
- `Keep-Alive`
- `Proxy-Authenticate`
- `Proxy-Authorization`
- `Proxy-Connection`
- `Trailer`
- `Transfer-Encoding`
- `Upgrade`
- `Via`
- `Warning`
- `X-Accel-Buffering`
- `X-Accel-Charset`
- `X-Accel-Limit-Rate`
- `X-Accel-Redirect`
- `X-Amz-Cf-.*`
- `X-Amzn-Auth`
- `X-Amzn-Cf-Billing`
- `X-Amzn-Cf-Id`
- `X-Amzn-Cf-Xff`
- `X-Amzn-ErrorType`
- `X-Amzn-Fle-Profile`
- `X-Amzn-Header-Count`
- `X-Amzn-Header-Order`

- `X-Amzn-Lambda-Integration-Tag`
- `X-Amzn-RequestId`
- `X-Edge-.*`
- `X-Forwarded-Proto`
- `X-Real-Ip`

Server-Timing Kopfzeile

Verwenden Sie die `Server-Timing` Header-Einstellung, um den `Server-Timing` Header in HTTP-Antworten zu aktivieren, die von gesendet wurden CloudFront. Sie können diesen Header verwenden, um Kennzahlen anzuzeigen, die Ihnen helfen können, Erkenntnisse über das Verhalten und die Leistung von CloudFront und Ihren Ursprung zu gewinnen. Sie können beispielsweise anzeigen, welche Cache-Ebene einen Cache-Treffer geliefert hat. Sie können auch die Latenz des ersten Byte vom Ursprung anzeigen, wenn ein Cache-Fehler vorliegt. Die Metriken in der `Server-Timing` Kopfzeile können Ihnen helfen, Probleme zu beheben oder die Effizienz Ihrer Konfiguration CloudFront oder Ihrer Originalkonfiguration zu testen.

Weitere Informationen zur Verwendung des `Server-Timing` Headers mit CloudFront finden Sie in den folgenden Themen.

Um den Header `Server-Timing` zu aktivieren, müssen Sie eine Antwort-Header-Richtlinie erstellen (oder bearbeiten).

Themen

- Abtastrate und Pragma-Anforderungs-Header
- Server-Timing Header vom Ursprung
- Server-Timing Header-Metriken
- Server-Timing Beispiele für Header

Abtastrate und Pragma-Anforderungs-Header

Wenn Sie den Header `Server-Timing` in einer Antwort-Header-Richtlinie aktivieren, geben Sie auch die Abtastrate an. Die Stichprobenrate ist eine Zahl 0—100 (einschließlich), die den Prozentsatz der Beantwortungen angibt, CloudFront zu denen Sie die `Server-Timing` Kopfzeile hinzufügen möchten. Wenn Sie die Samplerate auf 100 festlegen, wird der HTTP-Antwort für jede Anforderung, die dem Cacheverhalten entspricht, an das die Richtlinie für `Server-Timing`

Antwortheader angehängt ist, der Header CloudFront hinzugefügt. Wenn Sie den Wert auf 50 setzen, wird der Header zu 50% der Antworten für Anfragen CloudFront hinzugefügt, die dem Cacheverhalten entsprechen. Sie können als Abtastrate eine beliebige Zahl von 0 bis 100 mit bis zu vier Dezimalstellen festlegen.

Wenn die Sampling-Rate auf einen Wert unter 100 eingestellt ist, können Sie nicht kontrollieren, zu welchen Antworten der `Server-Timing` Header CloudFront hinzugefügt wird, sondern nur den Prozentsatz. Sie können jedoch den Header `Pragma` mit dem Wert `server-timing` in einer HTTP-Anforderung hinzufügen, um den Header `Server-Timing` in der Antwort auf diese Anforderung zu empfangen. Dies funktioniert unabhängig davon, auf welchen Wert die Abtastrate festgelegt ist. Selbst wenn die Sampling-Rate auf Null (0) gesetzt ist, wird der `Server-Timing` Header zur Antwort CloudFront hinzugefügt, wenn die Anfrage den `Pragma: server-timing` Header enthält.

Server-Timing Header vom Ursprung

Wenn ein Cache-Fehler auftritt und die Anfrage an den Ursprung CloudFront weitergeleitet wird, könnte der Ursprung in seiner Antwort auf CloudFront einen `Server-Timing` Header enthalten. In diesem Fall CloudFront fügt es seine Metriken dem `Server-Timing` Header hinzu, den es vom Ursprung erhalten hat. Die Antwort, die CloudFront an den Viewer gesendet wird, enthält einen einzelnen `Server-Timing` Header, der den Wert, der vom Ursprung stammt, und die CloudFront hinzugefügten Metriken enthält. Der Header-Wert vom Ursprung kann am Ende oder zwischen zwei Gruppen von Metriken stehen, die CloudFront den Header ergänzen.

Bei einem Cache-Treffer enthält die Antwort, die an den Viewer CloudFront gesendet wird, einen einzelnen `Server-Timing` Header, der nur die CloudFront Metriken im Header-Wert enthält (der Wert vom Ursprung ist nicht enthalten).

Server-Timing Header-Metriken

Wenn der `Server-Timing` Header zu einer HTTP-Antwort CloudFront hinzugefügt wird, enthält der Wert des Headers eine oder mehrere Metriken, die Ihnen helfen können, Erkenntnisse über das Verhalten und die Leistung von CloudFront und Ihren Ursprung zu gewinnen. Die folgende Liste enthält alle Metriken und ihre möglichen Werte. Ein `Server-Timing` Header enthält nur einige dieser Metriken, abhängig von der Art der Anfrage und der Antwort CloudFront.

Einige dieser Metriken sind nur mit einem Namen (ohne Wert) im Header `Server-Timing` enthalten. Andere bestehen aus einem Namen und einem Wert. Wenn eine Metrik einen Wert aufweist, werden Name und Wert durch ein Semikolon (`;`) getrennt. Wenn der Header mehr als eine Metrik enthält, werden die Metriken durch ein Komma (`,`) getrennt.

cdn-cache-hit

CloudFront hat eine Antwort aus dem Cache bereitgestellt, ohne eine Anfrage an den Ursprung zu stellen.

cdn-cache-refresh

CloudFront hat eine Antwort aus dem Cache bereitgestellt, nachdem eine Anfrage an den Ursprung gesendet wurde, um zu überprüfen, ob das zwischengespeicherte Objekt noch gültig ist. In diesem Fall CloudFront wurde nicht das vollständige Objekt vom Ursprung abgerufen.

cdn-cache-miss

CloudFront hat die Antwort aus dem Cache nicht bereitgestellt. In diesem Fall wurde das vollständige Objekt vom Ursprung CloudFront angefordert, bevor die Antwort zurückgegeben wurde.

cdn-pop

Enthält einen Wert, der beschreibt, welcher CloudFront Point of Presence (POP) die Anfrage bearbeitet hat.

cdn-rid

Enthält einen Wert mit dem CloudFront eindeutigen Bezeichner für die Anforderung. Sie können diese Anforderungskennung (Request Identifier, RID) bei der Fehlerbehebung mit dem Support verwenden.

cdn-hit-layer

Diese Metrik ist vorhanden, wenn CloudFront eine Antwort aus dem Cache bereitgestellt wird, ohne eine Anfrage an den Ursprung zu stellen. Sie enthält einen der folgenden Werte:

- EDGE — CloudFront hat die zwischengespeicherte Antwort von einem POP-Standort aus bereitgestellt.
- REC — CloudFront hat die zwischengespeicherte Antwort von einem regionalen Edge-Cache-Standort (REC) bereitgestellt.
- Origin Shield — CloudFront hat die zwischengespeicherte Antwort des REC bereitgestellt, der als Origin Shield fungiert.

cdn-upstream-layer

Wenn CloudFront das vollständige Objekt vom Ursprung angefordert wird, ist diese Metrik vorhanden und enthält einen der folgenden Werte:

- EDGE – Ein POP-Standort hat die Anforderung direkt an den Ursprung gesendet.
- REC – Ein REC-Standort hat die Anforderung direkt an den Ursprung gesendet.
- Origin Shield – Der REC, der als Origin Shield fungiert, hat die Anforderung direkt an den Ursprung gesendet.

cdn-upstream-dns

Enthält einen Wert mit der Anzahl der Millisekunden, die für das Abrufen des DNS-Datensatzes für den Ursprung aufgewendet wurden. Ein Wert von Null (0) gibt an, dass ein zwischengespeichertes DNS-Ergebnis CloudFront verwendet oder eine bestehende Verbindung wiederverwendet wurde.

cdn-upstream-connect

Enthält einen Wert mit der Anzahl der Millisekunden zwischen dem Abschluss der DNS-Anforderung des Ursprungs und dem Abschluss einer TCP-Verbindung (und ggf. TLS-Verbindung) zum Ursprung. Ein Wert von Null (0) gibt an, dass eine bestehende Verbindung CloudFront wiederverwendet wurde.

cdn-upstream-fbl

Enthält einen Wert mit der Anzahl der Millisekunden zwischen dem Abschluss der HTTP-Anforderung des Ursprungs und dem Zeitpunkt, an dem das erste Byte in der Antwort vom Ursprung empfangen wird (Latenz des ersten Byte).

cdn-downstream-fbl

Enthält einen Wert mit der Anzahl der Millisekunden zwischen dem Zeitpunkt, an dem der Edge-Standort die Anforderung vollständig empfangen hat, und dem Zeitpunkt, an dem das erste Byte der Antwort an den Viewer gesendet wird.

Server-Timing Beispiele für Header

Im Folgenden finden Sie Beispiele für einen `Server-Timing` Header, den ein Viewer möglicherweise empfängt, CloudFront wenn die `Server-Timing` Header-Einstellung aktiviert ist.

Example– Cache-Fehler

Das folgende Beispiel zeigt einen `Server-Timing` Header, den ein Viewer möglicherweise erhält, wenn sich das angeforderte Objekt nicht im CloudFront Cache befindet.

```
Server-Timing: cdn-upstream-layer;desc="EDGE",cdn-upstream-dns;dur=0,cdn-upstream-
connect;dur=114,cdn-upstream-fbl;dur=177,cdn-cache-miss,cdn-pop;desc="PHX50-C2",cdn-
rid;desc="yNPsyYn7skvTzwWkq3Wcc8Nj_foxUjQUe9H1ifslzWhb0w7aLbFvGg==",cdn-downstream-
fbl;dur=436
```

Dieser `Server-Timing`-Header gibt Folgendes an:

- Die Quellanforderung wurde von einem POP-Standort (CloudFront Point of Presence) aus gesendet (`cdn-upstream-layer;desc="EDGE"`).
- CloudFront verwendete ein zwischengespeichertes DNS-Ergebnis für den Ursprung (`cdn-upstream-dns;dur=0`).
- Es dauerte 114 Millisekunden, CloudFront bis die TCP-Verbindung (und TLS, falls zutreffend) zum Ursprung () hergestellt war. `cdn-upstream-connect;dur=114`
- Nach Abschluss der Anfrage () dauerte es 177 Millisekunden, CloudFront bis das erste Byte der Antwort vom Ursprung empfangen wurde. `cdn-upstream-fbl;dur=177`
- Das angeforderte Objekt befand sich nicht im CloudFront Cache (). `cdn-cache-miss`
- Die Anforderung wurde an dem durch den Code `PHX50-C2` identifizierten Edge-Standort empfangen (`cdn-pop;desc="PHX50-C2"`).
- Die CloudFront eindeutige ID für diese Anfrage war `yNPsyYn7skvTzwWkq3Wcc8Nj_foxUjQUe9H1ifslzWhb0w7aLbFvGg==` (`cdn-rid;desc="yNPsyYn7skvTzwWkq3Wcc8Nj_foxUjQUe9H1ifslzWhb0w7aLbFvGg=="`).
- Nach Erhalt der Viewer-Anfrage () dauerte es 436 Millisekunden, bis das erste Byte der Antwort an den Betrachter gesendet wurde. CloudFront `cdn-downstream-fbl;dur=436`

Example– Cache-Treffer

Das folgende Beispiel zeigt einen `Server-Timing` Header, den ein Viewer möglicherweise erhält, wenn sich das angeforderte Objekt im Cache befindet. CloudFront

```
Server-Timing: cdn-cache-hit,cdn-pop;desc="SEA19-C1",cdn-
rid;desc="nQBz4aJU2kP9iC3KHEq7vFxfMozu-VYBwGzkW9diOpeVc7xsrLKj-g==",cdn-hit-
layer;desc="REC",cdn-downstream-fbl;dur=137
```

Dieser `Server-Timing`-Header gibt Folgendes an:

- Das angeforderte Objekt war im Cache vorhanden (`cdn-cache-hit`).

- Die Anforderung wurde an dem durch den Code `SEA19-C1` identifizierten Edge-Standort empfangen (`cdn-pop;desc="SEA19-C1"`).
- Die CloudFront eindeutige ID für diese Anfrage war `nQBz4aJU2kP9iC3KHEq7vFxfMozu-VYBwGzkW9diOpeVc7xsrLKj-g==` (`cdn-rid;desc="nQBz4aJU2kP9iC3KHEq7vFxfMozu-VYBwGzkW9diOpeVc7xsrLKj-g=="`).
- Das angeforderte Objekt wurde an einem REC-Standort (regionaler Edge-Cache) zwischengespeichert (`cdn-hit-layer;desc="REC"`).
- Nach Erhalt der Viewer-Anfrage () dauerte es 137 Millisekunden, bis das erste Byte der Antwort an den Zuschauer gesendet wurde. CloudFront `cdn-downstream-fbl;dur=137`

Erstellen von Antwortheader-Richtlinien

Sie können eine Antwort-Header-Richtlinie verwenden, um die HTTP-Header anzugeben, die Amazon in HTTP-Antworten CloudFront hinzufügt oder entfernt. Weitere Informationen zu Antwort-Header-Richtlinien und zu Gründen für ihre Verwendung finden Sie unter Hinzufügen oder Entfernen von Antwortheadern mit einer Richtlinie.

Sie können in der Konsole eine Richtlinie für Antwort-Header erstellen. CloudFront Oder Sie können eine erstellen AWS CloudFormation, indem Sie die AWS Command Line Interface (AWS CLI) oder die CloudFront API verwenden. Nachdem Sie eine Response-Header-Richtlinie erstellt haben, fügen Sie sie einem oder mehreren Cache-Verhalten in einer CloudFront Distribution hinzu.

Prüfen Sie, ob eine der verwalteten Antwort-Header-Richtlinien für Ihren Anwendungsfall geeignet ist, bevor Sie eine benutzerdefinierte Antwort-Header-Richtlinie erstellen. Wenn dies der Fall ist, können Sie sie an Ihr Cache-Verhalten anfügen. Auf diese Weise müssen Sie keine eigene Antwort-Header-Richtlinie erstellen oder verwalten.

Console

So erstellen Sie eine Antwort-Header-Richtlinie (Konsole):

1. Melden Sie sich bei der an AWS-Managementkonsole und wechseln Sie dann auf der Seite Richtlinien in der CloudFront Konsole unter zur Registerkarte Antwortheader. https://console.aws.amazon.com/cloudfront/v4/home#/policies/responseHeaders
2. Wählen Sie Create response headers policy (Antwort-Header-Richtlinie erstellen) aus.
3. Gehen Sie im Formular Create response headers policy (Antwort-Header-Richtlinie erstellen) wie folgt vor:

a. Geben Sie im Bereich Details unter Name einen Namen für die Antwort-Header-Richtlinie und (optional) unter Description (Beschreibung) eine Beschreibung des Zwecks der Richtlinie ein.

b. Aktivieren Sie im Bereich Cross-Origin Resource Sharing (CORS) den Schalter Configure CORS (Konfigurieren von CORS) und konfigurieren Sie alle CORS-Header, die Sie der Richtlinie hinzufügen möchten. Wenn Sie möchten, dass die konfigurierten Header die vom Origin CloudFront empfangenen Header überschreiben, aktivieren Sie das Kontrollkästchen Origin-Override.

   Weitere Informationen zu den CORS-Header-Einstellungen finden Sie unter the section called “CORS-Header”.

c. Aktivieren Sie im Bereich Security headers (Sicherheits-Header) den Schalter und konfigurieren Sie alle Sicherheits-Header, die der Richtlinie hinzugefügt werden sollen.

   Weitere Informationen zu den Sicherheits-Header-Einstellungen finden Sie unter the section called “Sicherheits-Header”.

d. Fügen Sie im Bereich Custom headers (Benutzerdefinierte Header) alle benutzerdefinierten Header hinzu, die in der Richtlinie eingefügt werden sollen.

   Weitere Informationen zu benutzerdefinierten Header-Einstellungen finden Sie unter the section called “Benutzerdefinierte Header”.

e. Fügen Sie im Bereich „Kopfzeilen entfernen“ die Namen aller Header hinzu, die Sie aus der Antwort des Originals entfernen und nicht in die Antwort aufnehmen möchten CloudFront , CloudFront die an die Zuschauer gesendet wird.

   Weitere Informationen zu den Einstellungen für das Entfernen von Headern finden Sie unter the section called “Entfernen von Headern”.

f. Wählen Sie im Bereich Server-Timing header (Server-Timing-Header) die Umschalttaste Enable (Aktivieren) aus und geben Sie eine Abtastrate (eine Zahl zwischen 0 und 100, jeweils inklusive) ein.

   Weitere Informationen zum Header `Server-Timing` finden Sie unter the section called “Server-Timing Kopfzeile”.

4. Wählen Sie Create (Erstellen) aus, um die Richtlinie zu erstellen.

Nachdem Sie eine Richtlinie für Antwort-Header erstellt haben, können Sie sie an ein Cache-Verhalten in einer Verteilung anhängen. CloudFront

So fügen Sie eine Antwort-Header-Richtlinie an eine vorhandene Verteilung an (Konsole):

1. Öffnen Sie die Seite „Verteilungen“ in der CloudFront Konsole unter. https://console.aws.amazon.com/cloudfront/v4/home#/distributions
2. Wählen Sie die Verteilung aus, die Sie aktualisieren möchten, und anschließend die Registerkarte Verhaltensweisen aus.
3. Wählen Sie das Cache-Verhalten, das Sie aktualisieren möchten, und anschließend Edit (Bearbeiten) aus.

   Um ein neues Cacheverhalten zu erstellen, wählen Sie Verhalten erstellen aus.
4. Wählen Sie für Response headers policy (Antwort-Header-Richtlinie) die Richtlinie aus, die zum Cache-Verhalten hinzugefügt werden soll.
5. Wählen Sie Save changes (Änderungen speichern) aus, um das Cache-Verhalten zu aktualisieren. Wenn Sie ein neues Cache-Verhalten erstellen, wählen Sie Create behavior (Verhalten erstellen) aus.

So fügen Sie eine Antwort-Header-Richtlinie an eine neue Verteilung an (Konsole):

1. Öffnen Sie die CloudFront Konsole unter. https://console.aws.amazon.com/cloudfront/v4/home
2. Wählen Sie Create distribution (Distribution erstellen) aus.
3. Wählen Sie für Response headers policy (Antwort-Header-Richtlinie) die Richtlinie aus, die zum Cache-Verhalten hinzugefügt werden soll.
4. Wählen Sie die anderen Einstellungen für Ihre Verteilung aus. Weitere Informationen finden Sie unter the section called “Alle Distributionseinstellungen”.
5. Wählen Sie Create distribution (Verteilung erstellen) aus, um die Verteilung zu erstellen.

CloudFormation

Verwenden Sie den `AWS::CloudFront::ResponseHeadersPolicy` Ressourcentyp CloudFormation, um eine Response-Header-Richtlinie mit zu erstellen. Das folgende Beispiel zeigt die CloudFormation Vorlagensyntax im YAML-Format zum Erstellen einer Antwort-Header-Richtlinie.

```
Type: AWS::CloudFront::ResponseHeadersPolicy
Properties:
  ResponseHeadersPolicyConfig:
    Name: EXAMPLE-Response-Headers-Policy
    Comment: Example response headers policy for the documentation
    CorsConfig:
      AccessControlAllowCredentials: false
      AccessControlAllowHeaders:
        Items:
          - '*'
      AccessControlAllowMethods:
        Items:
          - GET
          - OPTIONS
      AccessControlAllowOrigins:
        Items:
          - https://example.com
          - https://docs.example.com
      AccessControlExposeHeaders:
        Items:
          - '*'
      AccessControlMaxAgeSec: 600
      OriginOverride: false
    CustomHeadersConfig:
      Items:
        - Header: Example-Custom-Header-1
          Value: value-1
          Override: true
        - Header: Example-Custom-Header-2
          Value: value-2
          Override: true
    SecurityHeadersConfig:
      ContentSecurityPolicy:
        ContentSecurityPolicy: default-src 'none'; img-src 'self'; script-src
 'self'; style-src 'self'; object-src 'none'; frame-ancestors 'none'
        Override: false
      ContentTypeOptions: # You don't need to specify a value for 'X-Content-Type-
Options'.
                          # Simply including it in the template sets its value to
 'nosniff'.
        Override: false
      FrameOptions:
        FrameOption: DENY
```

```
        Override: false
      ReferrerPolicy:
        ReferrerPolicy: same-origin
        Override: false
      StrictTransportSecurity:
        AccessControlMaxAgeSec: 63072000
        IncludeSubdomains: true
        Preload: true
        Override: false
      XSSProtection:
        ModeBlock: true # You can set ModeBlock to 'true' OR set a value for
ReportUri, but not both
        Protection: true
        Override: false
    ServerTimingHeadersConfig:
      Enabled: true
      SamplingRate: 50
    RemoveHeadersConfig:
      Items:
        - Header: Vary
        - Header: X-Powered-By
```

Weitere Informationen finden Sie unter AWS::CloudFront::ResponseHeadersRichtlinie im AWS CloudFormation Benutzerhandbuch.

CLI

Verwenden Sie den aws cloudfront create-response-headers-policy Befehl, um eine Response-Header-Richtlinie mit dem AWS Command Line Interface (AWS CLI) zu erstellen. Sie können eine Eingabedatei verwenden, um die Eingabeparameter für den Befehl bereitzustellen, anstatt jeden einzelnen Parameter als Befehlszeileneingabe anzugeben.

So erstellen Sie eine Antwort-Header-Richtlinie (CLI mit Eingabedatei):

1. Verwenden Sie den folgenden Befehl zum Erstellen einer Datei mit dem Namen `response-headers-policy.yaml`. Diese Datei enthält alle Eingabeparameter für den Befehl create-response-headers-policy.

```
aws cloudfront create-response-headers-policy --generate-cli-skeleton yaml-input
 > response-headers-policy.yaml
```

2. Öffnen Sie die Datei `response-headers-policy.yaml`, die Sie gerade erstellt haben. Bearbeiten Sie die Datei und geben Sie einen Richtliniennamen und die gewünschte Konfiguration der Antwort-Header-Richtlinie an. Speichern Sie anschließend die Datei.

   Weitere Informationen zu den Einstellungen für die Antwort-Header-Richtlinie finden Sie unter the section called “Grundlegendes zu Antwortheader-Richtlinien”.

3. Verwenden Sie den folgenden Befehl, um die Antwort-Header-Richtlinie zu erstellen. Die Richtlinie, die Sie erstellen, verwendet die Eingabeparameter aus der Datei `response-headers-policy.yaml`.

   ```
   aws cloudfront create-response-headers-policy --cli-input-yaml file://response-headers-policy.yaml
   ```

   Notieren Sie den `Id`-Wert in der Befehlsausgabe. Dies ist die ID der Antwort-Header-Richtlinie. Sie benötigen ihn, um die Richtlinie an das Cache-Verhalten einer CloudFront Distribution anzuhängen.

So fügen Sie eine Antwort-Header-Richtlinie an eine vorhandene Verteilung an (CLI mit Eingabedatei):

1. Verwenden Sie den folgenden Befehl, um die Verteilungskonfiguration für die CloudFront Distribution zu speichern, die Sie aktualisieren möchten. Ersetzen Sie es *`distribution_ID`* durch die Distribution-ID.

   ```
   aws cloudfront get-distribution-config --id distribution_ID --output yaml > dist-config.yaml
   ```

2. Öffnen Sie die Datei mit dem Namen `dist-config.yaml`, die Sie gerade erstellt haben. Bearbeiten Sie die Datei, indem Sie die folgenden Änderungen am Cache-Verhalten vornehmen, um die Antwort-Header-Richtlinie zu verwenden.

   - Fügen Sie im Cache-Verhalten ein Feld mit dem Namen `ResponseHeadersPolicyId` hinzu. Verwenden Sie für den Wert des Feldes die ID der Antwort-Header-Richtlinie, die Sie nach dem Erstellen der Richtlinie notiert haben.
   - Benennen Sie das Feld `ETag` in `IfMatch` um, ändern Sie jedoch nicht den Wert des Feldes.

Speichern Sie die Datei, wenn Sie fertig sind.

3. Verwenden Sie den folgenden Befehl, um die Verteilung so zu aktualisieren, dass die Antwort-Header-Richtlinie verwendet wird. Ersetzen Sie es *distribution_ID* durch die Distributions-ID.

```
aws cloudfront update-distribution --id distribution_ID --cli-input-yaml file://
dist-config.yaml
```

So fügen Sie eine Antwort-Header-Richtlinie an eine neue Verteilung an (CLI mit Eingabedatei):

1. Verwenden Sie den folgenden Befehl zum Erstellen einer Datei mit dem Namen `distribution.yaml`. Diese Datei enthält alle Eingabeparameter für den Befehl create-distribution.

```
aws cloudfront create-distribution --generate-cli-skeleton yaml-input >
 distribution.yaml
```

2. Öffnen Sie die Datei `distribution.yaml`, die Sie gerade erstellt haben. Geben Sie im Standard-Cache-Verhalten in das `ResponseHeadersPolicyId`-Feld die ID der Antwort-Header-Richtlinie ein, die Sie nach dem Erstellen der Richtlinie notiert haben. Fahren Sie mit der Bearbeitung der Datei fort, um die gewünschten Verteilungseinstellungen anzugeben, und speichern Sie die Datei, wenn Sie fertig sind.

   Weitere Informationen zu den Verteilungseinstellungen finden Sie unter Referenz für alle Distributionseinstellungen.

3. Verwenden Sie den folgenden Befehl, um die Verteilung mit Eingabeparametern aus der Datei `distribution.yaml` zu erstellen.

```
aws cloudfront create-distribution --cli-input-yaml file://distribution.yaml
```

API

Um eine Response-Header-Richtlinie mit der CloudFront API zu erstellen, verwenden Sie CreateResponseHeadersPolicy. Weitere Informationen zu den Feldern, die Sie in diesem API-Aufruf angeben, finden Sie in the section called “Grundlegendes zu Antwortheader-Richtlinien” und in der API-Referenzdokumentation für Ihr AWS SDK oder einen anderen API-Client.

Nachdem Sie eine Antwort-Header-Richtlinie erstellt haben, können Sie sie mit einem der folgenden API-Aufrufe an eine Cache-Verhaltensweise anfügen:

- Um es an ein Cache-Verhalten in einer vorhandenen Distribution anzuhängen, verwenden Sie UpdateDistribution.
- Um es an ein Cache-Verhalten in einer neuen Distribution anzuhängen, verwenden Sie CreateDistribution.

Geben Sie für beide API-Aufrufe die ID der Antwort-Header-Richtlinie in das Feld `ResponseHeadersPolicyId` innerhalb eines Cache-Verhaltens ein. Weitere Informationen zu den anderen Feldern, die Sie in diesen API-Aufrufen angeben, finden Sie in Referenz für alle Distributionseinstellungen und in der API-Referenzdokumentation für Ihr AWS SDK oder einen anderen API-Client.

Verwenden verwalteter Antwortheader-Richtlinien

Mit einer CloudFront Antwort-Header-Richtlinie können Sie die HTTP-Header angeben, die Amazon CloudFront entfernt oder zu Antworten hinzufügt, die es an Zuschauer sendet. Weitere Informationen zu Antwort-Header-Richtlinien und zu Gründen für ihre Verwendung finden Sie unter Hinzufügen oder Entfernen von Antwortheadern mit einer Richtlinie.

CloudFront bietet verwaltete Richtlinien für Antwort-Header, die Sie an das Cache-Verhalten in Ihren Distributionen anhängen können. CloudFront Bei Verwendung einer verwalteten Antwort-Header-Richtlinie müssen Sie keine eigene Richtlinie schreiben oder verwalten. Die verwalteten Richtlinien enthalten Sätze von HTTP-Antwort-Headern für häufige Anwendungsfälle.

Um eine verwaltete Richtlinie für Antwort-Header zu verwenden, fügen Sie sie einem Cache-Verhalten in Ihrer Distribution an. Das Verfahren entspricht dem der Erstellung einer benutzerdefinierten Antwort-Header-Richtlinie. Anstatt jedoch eine neue Richtlinie zu erstellen, fügen Sie eine der verwalteten Richtlinien an. Sie hängen die Richtlinie entweder anhand des Namens (mit

der Konsole) oder anhand der ID (mit CloudFormation AWS CLI, dem oder dem AWS SDKs) an. Die Namen und IDs sind im folgenden Abschnitt aufgeführt.

Weitere Informationen finden Sie unter the section called “Erstellen von Antwortheader-Richtlinien”.

In den folgenden Themen werden die verwalteten Antwort-Header-Richtlinien beschrieben, die Sie verwenden können.

Themen

- Cors-und- SecurityHeadersPolicy
- CORS-With-Preflight
- CORS- - with-preflight-and SecurityHeadersPolicy
- SecurityHeadersPolicy
- SimpleCORS

Cors-und- SecurityHeadersPolicy

Diese Richtlinie in der Konsole anzeigen CloudFront

Verwenden Sie diese verwaltete Richtlinie, um einfache CORS-Anforderungen von jedem Ursprung zuzulassen. Diese Richtlinie fügt außerdem allen Antworten, die an Zuschauer CloudFront gesendet werden, eine Reihe von Sicherheitsheadern hinzu. Diese Richtlinie verbindet die the section called “SimpleCORS”- und the section called “SecurityHeadersPolicy”-Richtlinien in einer Richtlinie.

Wenn Sie CloudFormation die AWS CLI oder die CloudFront API verwenden, lautet die ID für diese Richtlinie:

`e61eb60c-9c35-4d20-a928-2b84e02af89c`

Richtlinieneinstellungen

	Header-Name	Header-Wert	Override origin? (Ursprung überschreiben?)
CORS-Header:	`Access-Control-Allow-Origin`	*	Nein

	Header-Name	Header-Wert	Override origin? (Ursprung überschreiben?)
Sicherheits-Header:	`Referrer-Policy`	`strict-origin-when-cross-origin`	Nein
	`Strict-Transport-Security`	`max-age=31536000`	Nein
	`X-Content-Type-Options`	`nosniff`	Ja
	`X-Frame-Options`	`SAMEORIGIN`	Nein
	`X-XSS-Protection`	`1; mode=block`	Nein

CORS-With-Preflight

Diese Richtlinie in der CloudFront Konsole anzeigen

Mit dieser verwalteten Richtlinie werden CORS-Anforderungen von jedem Ursprung zugelassen, einschließlich Preflight-Anforderungen. CloudFront Fügt bei Preflight-Anfragen (mit der `OPTIONS` HTTP-Methode) alle drei der folgenden Header zur Antwort hinzu. Fügt bei einfachen CORS-Anfragen nur den CloudFront Header hinzu. `Access-Control-Allow-Origin`

Wenn die vom Ursprung CloudFront empfangene Antwort einen dieser Header enthält, CloudFront verwendet es den empfangenen Header (und seinen Wert) in seiner Antwort an den Betrachter. CloudFrontverwendet den Header in dieser Richtlinie nicht.

Wenn Sie CloudFormation die AWS CLI oder die CloudFront API verwenden, lautet die ID für diese Richtlinie:

`5cc3b908-e619-4b99-88e5-2cf7f45965bd`

Richtlinieneinstellungen

	Header-Name	Header-Wert	Override origin? (Ursprung überschre iben?)
CORS-Header:	`Access-Control-Allow-Methods`	`DELETE`, `GET`, `HEAD`, `OPTIONS`, `PATCH`, `POST`, `PUT`	Nein
	`Access-Control-Allow-Origin`	`*`	
	`Access-Control-Expose-Headers`	`*`	

CORS- - with-preflight-and SecurityHeadersPolicy

Diese Richtlinie in der CloudFront Konsole anzeigen

Verwenden Sie diese verwaltete Richtlinie, um CORS-Anforderungen von jedem Ursprung zuzulassen. Dies umfasst auch Preflight-Anforderungen. Diese Richtlinie fügt außerdem allen Antworten, die an Zuschauer CloudFront gesendet werden, eine Reihe von Sicherheitsheadern hinzu. Diese Richtlinie verbindet die the section called “CORS-With-Preflight”- und the section called “SecurityHeadersPolicy”-Richtlinien in einer Richtlinie.

Wenn Sie CloudFormation die AWS CLI oder die CloudFront API verwenden, lautet die ID für diese Richtlinie:

`eaab4381-ed33-4a86-88ca-d9558dc6cd63`

Richtlinieneinstellungen

	Header-Name	Header-Wert	Override origin? (Ursprung überschreiben?)
CORS-Header:	`Access-Control-Allow-Methods`	`DELETE, GET, HEAD, OPTIONS, PATCH, POST, PUT`	Nein
	`Access-Control-Allow-Origin`	`*`	
	`Access-Control-Expose-Headers`	`*`	
Sicherheits-Header:	`Referrer-Policy`	`strict-origin-when-cross-origin`	Nein
	`Strict-Transport-Security`	`max-age=31536000`	Nein
	`X-Content-Type-Options`	`nosniff`	Ja
	`X-Frame-Options`	`SAMEORIGIN`	Nein
	`X-XSS-Protection`	`1; mode=block`	Nein

SecurityHeadersPolicy

Diese Richtlinie in der CloudFront Konsole anzeigen

Verwenden Sie diese verwaltete Richtlinie, um allen Antworten, die an Zuschauer CloudFront gesendet werden, eine Reihe von Sicherheitsheadern hinzuzufügen. Weitere Informationen zu diesen Sicherheits-Headern finden Sie in den Websicherheitsrichtlinien von Mozilla.

Mit dieser Richtlinie für Antwort-Header werden alle Antworten CloudFront `X-Content-Type-Options: nosniff` hinzugefügt. Dies ist der Fall, wenn die Antwort, die von der Quelle CloudFront

empfangen wurde, diesen Header enthielt und wenn nicht. Für alle anderen Header in dieser Richtlinie gilt: Wenn die Antwort, die CloudFront vom Ursprung eingeht, den Header enthält, CloudFront wird der empfangene Header (und sein Wert) in der Antwort an den Betrachter verwendet. In dieser Richtlinie wird der Header nicht verwendet.

Wenn Sie CloudFormation die AWS CLI oder die CloudFront API verwenden, lautet die ID für diese Richtlinie:

`67f7725c-6f97-4210-82d7-5512b31e9d03`

Richtlinieneinstellungen

	Header-Name	Header-Wert	Override origin? (Ursprung überschreiben?)
Sicherheits-Header:	`Referrer-Policy`	`strict-origin-when-cross-origin`	Nein
	`Strict-Transport-Security`	`max-age=31536000`	Nein
	`X-Content-Type-Options`	`nosniff`	Ja
	`X-Frame-Options`	`SAMEORIGIN`	Nein
	`X-XSS-Protection`	`1; mode=block`	Nein

SimpleCORS

Diese Richtlinie in der CloudFront Konsole anzeigen

Verwenden Sie diese verwaltete Richtlinie, um einfache CORS-Anforderungen von jedem Ursprung zuzulassen. CloudFront Fügt mit dieser Richtlinie den Header `Access-Control-Allow-Origin: *` zu allen Antworten für einfache CORS-Anfragen hinzu.

Wenn die Antwort, die vom Ursprung CloudFront empfangen wird, den `Access-Control-Allow-Origin` Header enthält, CloudFront verwendet er diesen Header (und seinen Wert) in seiner Antwort an den Betrachter. CloudFront verwendet den Header in dieser Richtlinie nicht.

Wenn Sie CloudFormation die AWS CLI oder die CloudFront API verwenden, lautet die ID für diese Richtlinie:

`60669652-455b-4ae9-85a4-c4c02393f86c`

Richtlinieneinstellungen

	Header-Name	Header-Wert	Override origin? (Ursprung überschreiben?)
CORS-Header:	`Access-Control-Allow-Origin`	`*`	Nein

Verhalten von Anfragen und Antworten

In den folgenden Themen wird beschrieben, wie Anfragen und Antworten CloudFront behandelt werden.

Sie erfahren, wie mit Amazon S3 oder benutzerdefinierten Ursprüngen CloudFront interagiert, verschiedene HTTP-Methoden und Header verarbeitet, Statuscodes verarbeitet und das Caching und die Fehlerantworten verwaltet.

Themen

- Wie CloudFront verarbeitet HTTP- und HTTPS-Anfragen
- Verhalten von Anfragen und Antworten für Amazon-S3-Ursprünge
- Verhalten von Anfragen und Antworten für benutzerdefinierte Ursprungsserver
- Verhalten von Anfragen und Antworten für Ursprungsgruppen
- Hinzufügen benutzerdefinierter Header zu Ursprungsanforderungen
- Wie CloudFront verarbeitet Teilanfragen für ein Objekt (Bereichs-GETs)
- Wie CloudFront verarbeitet HTTP 3xx-Statuscodes von Ihrem Ursprung
- Wie CloudFront verarbeitet die HTTP 4xx- und 5xx-Statuscodes von Ihrem Ursprung
- Erstellen von benutzerdefinierten Fehlerantworten

Wie CloudFront verarbeitet HTTP- und HTTPS-Anfragen

Für Amazon S3-Ursprünge CloudFront akzeptiert es standardmäßig Anfragen sowohl im HTTP- als auch im HTTPS-Protokoll für Objekte in einer CloudFront Distribution. CloudFront leitet die Anfragen dann mit demselben Protokoll, in dem die Anfragen gestellt wurden, an Ihren Amazon S3-Bucket weiter.

Bei benutzerdefinierten Ursprüngen können Sie beim Erstellen Ihrer Verteilung festlegen, wie CloudFront auf Ihren Ursprung zugreifen: nur HTTP oder entsprechend dem vom Betrachter verwendeten Protokoll. Weitere Informationen zur Behandlung von HTTP- CloudFront und HTTPS-Anfragen für benutzerdefinierte Ursprünge finden Sie unterProtokolle.

Weitere Informationen dazu, wie Sie Ihre Verteilung so einschränken, dass Endbenutzer nur mit HTTPS auf Objekte zugreifen können, finden Sie unter Verwenden Sie HTTPS mit CloudFront.

> **Note**
>
> Die Gebühr für HTTPS-Anfragen ist höher als die Gebühr für HTTP-Anfragen. Weitere Informationen zu den Abrechnungstarifen finden Sie unter CloudFront Preise.

Verhalten von Anfragen und Antworten für Amazon-S3-Ursprünge

In den folgenden Abschnitten erfahren Sie, wie Anfragen und Antworten CloudFront verarbeitet werden, wenn Sie Amazon S3 als Ausgangspunkt verwenden:

Themen

- Wie CloudFront verarbeitet und leitet Anfragen an Ihren Amazon S3-Ursprung weiter
- Wie CloudFront verarbeitet Antworten von Ihrem Amazon S3-Ursprung

Wie CloudFront verarbeitet und leitet Anfragen an Ihren Amazon S3-Ursprung weiter

Erfahren Sie, wie Viewer-Anfragen CloudFront verarbeitet und die Anfragen an Ihren Amazon S3-Ursprung weitergeleitet werden.

Inhalt

- Cache-Dauer und Mindest-TTL
- Client-IP-Adressen
- Bedingte GET-Anforderungen
- Cookies
- Cross-origin gemeinsame Nutzung von Ressourcen (CORS)
- GET-Anfragen mit Anfragetext
- HTTP-Methoden
- HTTP-Anforderungsheader, die CloudFront entfernt oder aktualisiert werden
- Maximale Länge einer Anfrage und maximale Länge einer URL
- OCSP-Stapling
- Protokolle

- Abfragezeichenfolgen
- Timeout der Ursprungsverbindung und Verbindungsversuche zum Ursprung
- Ursprungs-Reaktions-Timeout
- Gleichzeitige Anfragen für dasselbe Objekt (Zusammenfassung von Anfragen)

Cache-Dauer und Mindest-TTL

Um zu steuern, wie lange Ihre Objekte in einem CloudFront Cache bleiben, bevor sie CloudFront eine weitere Anfrage an Ihren Ursprung weiterleiten, können Sie:

- Ihren Ursprungsserver so konfigurieren, dass jedem Objekt ein `Cache-Control`- oder `Expires`-Header-Feld hinzugefügt wird
- Geben Sie einen Wert für die Mindest-TTL im CloudFront Cache-Verhalten an.
- Den Standardwert von 24 Stunden verwenden

Weitere Informationen finden Sie unter Verwalten der Dauer, die Inhalte im Cache bleiben (Ablauf).

Client-IP-Adressen

Wenn ein Viewer eine Anfrage an einen Anforderungsheader sendet CloudFront und keinen `X-Forwarded-For` Anforderungsheader enthält, CloudFront ruft er die IP-Adresse des Viewers von der TCP-Verbindung ab, fügt einen `X-Forwarded-For` Header hinzu, der die IP-Adresse enthält, und leitet die Anfrage an den Ursprung weiter. Wenn CloudFront z. B. die IP-Adresse `192.0.2.2` von der TCP-Verbindung abruft, wird der folgende Header an den Ursprungs-Server weitergeleitet:

```
X-Forwarded-For: 192.0.2.2
```

Wenn ein Viewer eine Anfrage an einen Request-Header sendet CloudFront und einen `X-Forwarded-For` Request-Header einschließt, CloudFront ruft er die IP-Adresse des Viewers von der TCP-Verbindung ab, hängt sie an das Ende des `X-Forwarded-For` Headers an und leitet die Anfrage an den Ursprung weiter. Wenn die Viewer-Anfrage beispielsweise die IP-Adresse der TCP-Verbindung enthält `X-Forwarded-For: 192.0.2.4,192.0.2.3` und `192.0.2.2` von ihr CloudFront abruft, leitet sie den folgenden Header an den Ursprung weiter:

```
X-Forwarded-For: 192.0.2.4,192.0.2.3,192.0.2.2
```

> **Note**
>
> Der `X-Forwarded-For`-Header enthält IPv4-Adressen (z. B. 192.0.2.44) und IPv6-Adressen (z. B. 2001:0db8:85a3::8a2e:0370:7334).
> Verwenden Sie beim Analysieren von IPv6-Adressen im `X-Forwarded-For` Header Standardbibliotheken zum Analysieren von IP-Adressen, die jedes gültige RFC 4291-IPv6-Format verarbeiten können.

Bedingte GET-Anforderungen

Wenn eine CloudFront Anfrage für ein Objekt aus einem Edge-Cache eingeht, das abgelaufen ist, leitet es die Anfrage an den Amazon S3-Ursprung weiter, um die neueste Version des Objekts abzurufen oder um von Amazon S3 eine Bestätigung zu erhalten, dass der CloudFront Edge-Cache bereits über die neueste Version verfügt. Als Amazon S3 das Objekt ursprünglich an sendete CloudFront, enthielt es einen `ETag` Wert und einen `LastModified` Wert in der Antwort. Fügt in der neuen Anfrage CloudFront , die an Amazon S3 weitergeleitet wird, einen oder beide der folgenden Header CloudFront hinzu:

- Einen `If-Match`- oder `If-None-Match`-Header mit dem `ETag`-Wert für die abgelaufene Version des Objekts
- Einen `If-Modified-Since`-Header mit dem `LastModified`-Wert für die abgelaufene Version des Objekts

Amazon S3 verwendet diese Informationen, um zu ermitteln, ob das Objekt aktualisiert wurde und daher, ob das gesamte Objekt an einen HTTP 304-Statuscode (nicht geändert) zurückgegeben werden soll CloudFront oder nur an einen HTTP 304-Statuscode zurückgegeben werden soll.

Cookies

Amazon S3 verarbeitet keine Cookies. Wenn Sie ein Cache-Verhalten so konfigurieren, dass Cookies an einen Amazon S3-Ursprung weitergeleitet werden, CloudFront leitet es die Cookies weiter, Amazon S3 ignoriert sie jedoch. Alle zukünftigen Anfragen zu demselben Objekt werden mithilfe des im Cache vorhandenen Objekts bedient – unabhängig davon, ob Sie Änderungen an den Cookies vornehmen.

Cross-origin gemeinsame Nutzung von Ressourcen (CORS)

Wenn Sie CloudFront die Amazon S3-Einstellungen für die ursprungsübergreifende gemeinsame Nutzung von Ressourcen respektieren möchten, konfigurieren Sie, dass ausgewählte Header CloudFront an Amazon S3 weitergeleitet werden. Weitere Informationen finden Sie unter Zwischenspeichern von Inhalten auf der Grundlage von Anforderungsheadern.

GET-Anfragen mit Anfragetext

Wenn eine `GET` Viewer-Anfrage einen Text enthält, wird der HTTP-Statuscode 403 (Forbidden) an den Betrachter CloudFront zurückgegeben.

HTTP-Methoden

Wenn Sie die Konfiguration so konfigurieren CloudFront , dass alle unterstützten HTTP-Methoden verarbeitet werden, CloudFront akzeptiert es die folgenden Anfragen von Viewern und leitet sie an Ihren Amazon S3-Ursprung weiter:

- `DELETE`
- `GET`
- `HEAD`
- `OPTIONS`
- `PATCH`
- `POST`
- `PUT`

CloudFront speichert Antworten auf `GET` und `HEAD` Anfragen immer im Cache. Sie können auch so konfigurieren CloudFront , dass Antworten auf `OPTIONS` Anfragen zwischengespeichert werden. CloudFront speichert keine Antworten auf Anfragen, die die anderen Methoden verwenden.

Wenn Sie mehrteilige Uploads verwenden möchten, um Objekte zu einem Amazon S3-Bucket hinzuzufügen, müssen Sie Ihrer Distribution eine CloudFront Origin Access Control (OAC) hinzufügen und dem OAC die erforderlichen Berechtigungen erteilen. Weitere Informationen finden Sie unter the section called “Beschränken des Zugriffs auf einen Amazon-S3-Ursprung”.

> **Important**
>
> Wenn Sie so konfigurieren CloudFront , dass alle CloudFront unterstützten HTTP-Methoden akzeptiert und an Amazon S3 weitergeleitet werden, müssen Sie ein CloudFront OAC erstellen, um den Zugriff auf Ihre Amazon S3-Inhalte einzuschränken, und dem OAC die erforderlichen Berechtigungen erteilen. Wenn Sie beispielsweise so konfigurieren CloudFront , dass diese Methoden akzeptiert und weitergeleitet werden, weil Sie die `PUT` Methode verwenden möchten, müssen Sie die Amazon S3-Bucket-Richtlinien so konfigurieren, dass `DELETE` Anfragen angemessen behandelt werden, sodass Zuschauer keine Ressourcen löschen können, die Sie nicht möchten. Weitere Informationen finden Sie unter the section called “Beschränken des Zugriffs auf einen Amazon-S3-Ursprung”.

Informationen zu den von Amazon S3 unterstützten Operationen finden Sie in der Amazon S3-Dokumentation.

HTTP-Anforderungsheader, die CloudFront entfernt oder aktualisiert werden

CloudFront entfernt oder aktualisiert einige Header, bevor Anfragen an Ihren Amazon S3-Ursprung weitergeleitet werden. Für die meisten Header ist dieses Verhalten dasselbe wie für benutzerdefinierte Ursprünge. Eine vollständige Liste der HTTP-Anforderungsheader und deren Verarbeitung finden CloudFront Sie unter. Header und CloudFront Verhalten von HTTP-Anfragen (benutzerdefiniert und Amazon S3-Ursprünge)

Maximale Länge einer Anfrage und maximale Länge einer URL

Die maximale Länge einer Anfrage, einschließlich des Pfads, der Abfragezeichenfolge (falls vorhanden) und der Header, beträgt 32.768 Byte.

CloudFront erstellt eine URL aus der Anfrage. Die maximale Länge dieser URL beträgt 8 192 Byte.

Wenn eine URL die maximale Länge überschreitet, wird der HTTP-Statuscode 414 (URI Too Long) an den Viewer CloudFront zurückgegeben. Wenn eine Anforderung die maximale Länge überschreitet, weil die Header-Größe überschritten wird, wird der HTTP-Statuscode 494 an den Viewer CloudFront zurückgegeben. In beiden Fällen CloudFront wird dann die TCP-Verbindung zum Viewer beendet.

OCSP-Stapling

Wenn ein Betrachter eine HTTPS-Anfrage für ein Objekt einreicht CloudFront oder der Betrachter bei der Zertifizierungsstelle (CA) bestätigen muss, dass das SSL-Zertifikat für die Domain nicht gesperrt wurde. OCSP-Stapling beschleunigt die Zertifikatsvalidierung, indem es ermöglicht CloudFront , das Zertifikat zu validieren und die Antwort von der CA zwischenzuspeichern, sodass der Client das Zertifikat nicht direkt bei der CA validieren muss.

Die Leistungsverbesserung durch OCSP-Stapling ist ausgeprägter, wenn CloudFront viele HTTPS-Anfragen für Objekte in derselben Domain eingehen. Jeder Server an einem CloudFront -Edge-Standort muss eine separate Validierungsanfrage senden. Wenn viele HTTPS-Anfragen für dieselbe Domain eingehen, CloudFront erhält jeder Server am Edge-Standort bald eine Antwort von der CA, die er an ein Paket im SSL-Handshake heften kann. Wenn der Betrachter davon überzeugt ist, dass das Zertifikat gültig ist, CloudFront kann er das angeforderte Objekt bereitstellen. Wenn Ihre Verteilung nicht viel Datenverkehr an einem CloudFront -Edge-Standort generiert, werden neue Anfragen mit einer höheren Wahrscheinlichkeit an einen Server weitergeleitet, der das Zertifikat noch nicht bei der CA validiert hat. In diesem Fall führt der Viewer den Validierungsschritt separat durch und der CloudFront Server stellt das Objekt bereit. Dieser CloudFront Server sendet auch eine Validierungsanforderung an die CA. Wenn er also das nächste Mal eine Anfrage erhält, die denselben Domainnamen enthält, erhält er eine Validierungsantwort von der CA.

Protokolle

CloudFront leitet HTTP- oder HTTPS-Anfragen an den Ursprungsserver weiter, basierend auf dem Protokoll der Viewer-Anfrage, entweder HTTP oder HTTPS.

> **Important**
>
> Wenn Ihr Amazon S3-Bucket als Website-Endpunkt konfiguriert ist, können Sie nicht so konfigurieren CloudFront , dass HTTPS für die Kommunikation mit Ihrem Ursprung verwendet wird, da Amazon S3 in dieser Konfiguration keine HTTPS-Verbindungen unterstützt.

Abfragezeichenfolgen

Sie können konfigurieren, ob Abfragezeichenfolgenparameter an Ihren Amazon S3-Ursprung CloudFront weitergeleitet werden sollen. Weitere Informationen finden Sie unter Zwischenspeichern von Inhalten auf der Grundlage von Abfragezeichenfolgeparametern.

Timeout der Ursprungsverbindung und Verbindungsversuche zum Ursprung

Das Timeout der Ursprungsverbindung ist die Anzahl der Sekunden, die CloudFront beim Versuch, eine Verbindung zum Ursprung herzustellen, vergeht.

Die Anzahl der Versuche, eine Verbindung zum Ursprung herzustellen, gibt an, wie oft CloudFront versucht wurde, eine Verbindung zum Ursprung herzustellen.

Zusammen bestimmen diese Einstellungen, wie lange CloudFront versucht wird, eine Verbindung zum Ursprung herzustellen, bevor ein Failover zum sekundären Ursprung erfolgt (im Fall einer Ursprungsgruppe) oder eine Fehlermeldung an den Viewer zurückgegeben wird. CloudFront Wartet standardmäßig bis zu 30 Sekunden (3 Versuche von jeweils 10 Sekunden), bevor versucht wird, eine Verbindung zum sekundären Ursprung herzustellen, oder eine Fehlermeldung zurückgegeben wird. Sie können diese Zeit reduzieren, indem Sie ein kürzeres Verbindungs-Timeout, weniger Versuche oder beides angeben.

Weitere Informationen finden Sie unter Steuern von Timeouts und Verbindungsversuchen für Ursprünge.

Ursprungs-Reaktions-Timeout

Das Ursprungs-Reaktions-Timeout, das auch als Ursprungs-Lese-Timeout oder Ursprungs-Anforderungs-Timeout bezeichnet wird, gilt für Folgendes:

- Die Zeitspanne in Sekunden, die auf eine Antwort CloudFront wartet, nachdem eine Anfrage an den Ursprung weitergeleitet wurde.
- Die Zeitdauer in Sekunden, die nach dem Empfang eines Pakets mit einer Antwort vom Ursprung und vor dem Empfang des nächsten Pakets CloudFront gewartet wird.

CloudFront Das Verhalten hängt von der HTTP-Methode der Viewer-Anfrage ab:

- `GET`und `HEAD` Anfragen — Wenn der Absender nicht innerhalb von 30 Sekunden reagiert oder für 30 Sekunden nicht mehr reagiert, wird CloudFront die Verbindung unterbrochen. Wenn die angegebene Anzahl von Ursprungsverbindungsversuchen mehr als 1 ist, versuchen CloudFront Sie erneut, eine vollständige Antwort zu erhalten. CloudFront versucht es bis zu dreimal, abhängig vom Wert der Einstellung für die ursprünglichen Verbindungsversuche. Wenn der Ursprung beim letzten Versuch keine Antwort sendet, unternimmt CloudFront erst dann einen weiteren Versuch, wenn die nächste Anfrage für Inhalte auf demselben Ursprung empfangen wird.

- `DELETE`, `OPTIONS` `PATCHPUT`, und `POST` Anfragen — Wenn der Absender nicht innerhalb von 30 Sekunden antwortet, wird die Verbindung CloudFront unterbrochen und es wird nicht erneut versucht, den Absender zu kontaktieren. Der Client kann die Anfrage erneut senden, falls erforderlich.

Sie können das Reaktions-Timeout für einen Amazon S3-Ursprung (ein S3-Bucket, der nicht mit statischem Website-Hosting konfiguriert ist) nicht ändern.

Gleichzeitige Anfragen für dasselbe Objekt (Zusammenfassung von Anfragen)

Wenn ein CloudFront Edge-Standort eine Anfrage für ein Objekt erhält und sich das Objekt nicht im Cache befindet oder das zwischengespeicherte Objekt abgelaufen ist, wird die Anfrage CloudFront sofort an den Ursprung gesendet. Wenn jedoch gleichzeitig Anfragen für dasselbe Objekt eingehen — das heißt, wenn zusätzliche Anfragen für dasselbe Objekt (mit demselben Cacheschlüssel) an der Edge-Position ankommen, bevor die Antwort auf die erste Anfrage CloudFront empfangen wird — wird eine CloudFront Pause eingelegt, bevor die zusätzlichen Anfragen an den Ursprung weitergeleitet werden. Diese kurze Pause trägt dazu bei, die Belastung des Ursprungs zu verringern. CloudFront sendet die Antwort von der ursprünglichen Anfrage auf alle Anfragen, die während der Pause eingegangen sind. Dies wird als Request Collapsing (Zusammenfassung von Anfragen) bezeichnet. In CloudFront Protokollen wird die erste Anfrage `Miss` im `x-edge-result-type` Feld als a identifiziert, und die reduzierten Anfragen werden als a identifiziert. `Hit` Weitere Hinweise zu CloudFront Protokollen finden Sie unterthe section called “CloudFront und Edge-Funktionsprotokollierung”.

CloudFront blendet nur Anfragen aus, die einen Cacheschlüssel gemeinsam verwenden. Wenn die zusätzlichen Anfragen nicht denselben Cacheschlüssel verwenden, weil Sie beispielsweise so konfiguriert CloudFront haben, dass das Zwischenspeichern auf der Grundlage von Anforderungsheadern, Cookies oder Abfragezeichenfolgen erfolgt, CloudFront leitet alle Anfragen mit einem eindeutigen Cacheschlüssel an Ihren Ursprung weiter.

Um die Reduzierung von Anforderungen insgesamt zu verhindern, können Sie die verwaltete Cache-Richtlinie `CachingDisabled` verwenden, die auch das Caching verhindert. Weitere Informationen finden Sie unter Verwenden verwalteter Cache-Richtlinien.

Wenn Sie eine Reduzierung von Anforderungen für bestimmte Objekte verhindern möchten, können Sie die minimale TTL für das Cacheverhalten auf 0 setzen und den Ursprung so konfigurieren, dass `Cache-Control: private`, `Cache-Control: no-store`, `Cache-Control: no-cache`, `Cache-Control: max-age=0` oder `Cache-Control: s-maxage=0` gesendet wird. Diese

Konfigurationen erhöhen die Belastung Ihres Ursprungs und führen zu zusätzlicher Latenz für die gleichzeitigen Anfragen, die unterbrochen werden, während auf die Antwort auf die CloudFront erste Anfrage gewartet wird.

Wie CloudFront verarbeitet Antworten von Ihrem Amazon S3-Ursprung

Erfahren Sie, wie Antworten aus Ihrem Amazon S3-Ursprung CloudFront verarbeitet werden.

Inhalt

- Abgebrochene Anfragen
- HTTP-Antwort-Header, die CloudFront entfernt oder aktualisiert werden
- Maximale Dateigröße, die zwischengespeichert werden kann
- Umleitungen

Abgebrochene Anfragen

Wenn sich ein Objekt nicht im Edge-Cache befindet und ein Viewer eine Sitzung beendet (z. B. einen Browser schließt), nachdem CloudFront er das Objekt von Ihrem Ursprung abgerufen hat, aber bevor er das angeforderte Objekt bereitstellen kann, wird das Objekt CloudFront nicht am Edge-Standort zwischengespeichert.

HTTP-Antwort-Header, die CloudFront entfernt oder aktualisiert werden

CloudFront entfernt oder aktualisiert die folgenden Header-Felder, bevor die Antwort von Ihrem Amazon S3-Ursprung an den Viewer weitergeleitet wird:

- `X-Amz-Id-2`
- `X-Amz-Request-Id`
- `Set-Cookie`— Wenn Sie CloudFront die Weiterleitung von Cookies konfigurieren, wird das `Set-Cookie` Header-Feld an die Clients weitergeleitet. Weitere Informationen finden Sie unter Zwischenspeichern von Inhalten auf der Grundlage von Cookies.
- `Trailer`
- `Transfer-Encoding`— Wenn Ihr Amazon S3-Ursprung dieses Header-Feld zurückgibt, CloudFront wird der Wert auf gesetzt, `chunked` bevor die Antwort an den Viewer zurückgegeben wird.

- `Upgrade`
- `Via`— CloudFront setzt den Wert in der Antwort an den Zuschauer auf den folgenden Wert:

  `Via: http-version alphanumeric-string.cloudfront.net (CloudFront)`

  Der Wert ist beispielsweise wie folgt:

  `Via: 1.1 1026589cc7887e7a0dc7827b4example.cloudfront.net (CloudFront)`

Maximale Dateigröße, die zwischengespeichert werden kann

Die maximale Größe eines Antworttextes, der in seinem Cache CloudFront gespeichert wird, beträgt 50 GB. Dazu gehören auch Antworten für aufgeteilte Übertragungen, in denen kein Wert für die `Content-Length`-Kopfzeile angegeben wurde.

Sie können ein Objekt zwischenspeichern, das größer als diese Größe ist, indem Sie Bereichsanforderungen verwenden, um die Objekte in Teilen anzufordern, die jeweils 50 GB oder weniger groß sind. CloudFront CloudFrontspeichert diese Teile im Cache, da jeder von ihnen 50 GB oder weniger groß ist. Nachdem der Viewer alle Teile des Objekts abgerufen hat, kann er das ursprüngliche, größere Objekt rekonstruieren. Weitere Informationen finden Sie unter Verwenden von Bereichsanforderungen zum Zwischenspeichern großer Objekte.

Umleitungen

Sie können einen Amazon S3-Bucket so konfigurieren, dass alle Anfragen an einen anderen Host-Namen umgeleitet werden. Dabei kann es sich um einen anderen Amazon S3-Bucket oder um einen HTTP-Server handeln. Wenn Sie einen Bucket so konfigurieren, dass er alle Anfragen umleitet, und wenn der Bucket der Ursprung einer CloudFront Verteilung ist, empfehlen wir, dass Sie den Bucket so konfigurieren, dass alle Anfragen an eine CloudFront Distribution umgeleitet werden, indem Sie entweder den Domainnamen für die Verteilung (z. B. d111111abcdef8.cloudfront.net) oder einen alternativen Domainnamen (ein CNAME) verwenden, der einer Distribution zugeordnet ist (z. B. example.com). Andernfalls werden Viewer-Anfragen umgangen CloudFront und die Objekte werden direkt vom neuen Ursprung aus bedient.

> **Note**
>
> Wenn Sie Viewer-Anforderungen an einen alternativen Domain-Namen umleiten, müssen Sie auch den DNS-Service für Ihre Domain aktualisieren, indem Sie einen CNAME-Datensatz

hinzufügen. Weitere Informationen finden Sie unter Verwenden von benutzerdefinierten URLs durch Hinzufügen von alternativen Domainnamen (CNAMEs).

Wenn Sie einen Bucket so konfigurieren, dass er alle Anfragen umleitet, geschieht Folgendes:

1. Ein Viewer (z. B. ein Browser) fordert ein Objekt von an CloudFront.
2. CloudFront leitet die Anfrage an den Amazon S3-Bucket weiter, der der Ursprung Ihrer Distribution ist.
3. Amazon S3 gibt einen HTTP-Statuscode 301 (dauerhaft verschoben) sowie den neuen Speicherort zurück.
4. CloudFront speichert den Umleitungsstatuscode und den neuen Standort im Cache und gibt die Werte an den Viewer zurück. CloudFront folgt nicht der Weiterleitung, um das Objekt vom neuen Standort abzurufen.
5. Der Viewer sendet eine weitere Anfrage für das Objekt, aber dieses Mal gibt der Viewer den neuen Standort an, von dem es stammt CloudFront:

   - Wenn der Amazon S3-Bucket alle Anfragen an eine CloudFront Distribution umleitet und dabei entweder den Domainnamen für die Verteilung oder einen alternativen Domainnamen verwendet, CloudFront fordert er das Objekt vom Amazon S3-Bucket oder vom HTTP-Server am neuen Standort an. Wenn der neue Standort das Objekt zurückgibt, wird es an den Viewer CloudFront zurückgegeben und an einem Edge-Standort zwischengespeichert.
   - Wenn der Amazon S3-Bucket Anfragen an einen anderen Standort umleitet, wird die zweite Anfrage umgangen. CloudFront Der Amazon S3-Bucket oder der HTTP-Server am neuen Standort gibt das Objekt direkt an den Viewer zurück, sodass das Objekt niemals in einem Edge-Cache zwischengespeichert wird. CloudFront

Verhalten von Anfragen und Antworten für benutzerdefinierte Ursprungsserver

In den folgenden Abschnitten erfahren Sie, wie Anfragen und Antworten CloudFront verarbeitet werden, wenn Sie benutzerdefinierte Ursprünge verwenden:

Themen

- Wie CloudFront verarbeitet und leitet Anfragen an Ihren benutzerdefinierten Ursprung weiter

- Wie CloudFront verarbeitet Antworten von Ihrem benutzerdefinierten Ursprung

Wie CloudFront verarbeitet und leitet Anfragen an Ihren benutzerdefinierten Ursprung weiter

Erfahre, wie Zuschaueranfragen CloudFront verarbeitet und die Anfragen an deinen benutzerdefinierten Absender weitergeleitet werden.

Inhalt

- Authentifizierung
- Cache-Dauer und Mindest-TTL
- Client-IP-Adressen
- Client-side SSL-Authentifizierung
- Komprimierung
- Bedingte Anforderungen
- Cookies
- Cross-origin gemeinsame Nutzung von Ressourcen (CORS)
- Verschlüsselung
- GET-Anfragen mit Anfragetext
- HTTP-Methoden
- Header und CloudFront Verhalten von HTTP-Anfragen (benutzerdefiniert und Amazon S3-Ursprünge)
- HTTP-Version
- Maximale Länge einer Anfrage und maximale Länge einer URL
- OCSP-Stapling
- Persistente Verbindungen
- Protokolle
- Abfragezeichenfolgen
- Timeout der Ursprungsverbindung und Verbindungsversuche zum Ursprung
- Ursprungs-Reaktions-Timeout
- Gleichzeitige Anfragen für dasselbe Objekt (Zusammenfassung von Anfragen)

- User-Agent-Header

Authentifizierung

Wenn Sie den `Authorization`-Header an Ihren Ursprung weiterleiten, können Sie Ihren Ursprungsserver so konfigurieren, dass eine Client-Authentifizierung für die folgenden Arten von Anforderungen angefordert wird:

- `DELETE`
- `GET`
- `HEAD`
- `PATCH`
- `PUT`
- `POST`

Für `OPTIONS` Anfragen kann die Client-Authentifizierung nur konfiguriert werden, wenn Sie die folgenden CloudFront Einstellungen verwenden:

- CloudFront ist so konfiguriert, dass der `Authorization` Header an Ihren Ursprung weitergeleitet wird
- CloudFront ist so konfiguriert, dass die Antwort auf `OPTIONS` Anfragen nicht zwischengespeichert wird

Weitere Informationen finden Sie unter Konfigurieren Sie CloudFront , um den Authorization-Header weiterzuleiten.

Sie können HTTP oder HTTPS verwenden, um Anforderungen an Ihren Ursprungsserver weiterzuleiten. Weitere Informationen finden Sie unter Verwenden Sie HTTPS mit CloudFront.

Cache-Dauer und Mindest-TTL

Um zu steuern, wie lange Ihre Objekte in einem CloudFront Cache bleiben, bevor CloudFront sie eine weitere Anfrage an Ihren Ursprung weiterleiten, können Sie:

- Ihren Ursprungsserver so konfigurieren, dass jedem Objekt ein `Cache-Control`- oder `Expires`-Header-Feld hinzugefügt wird
- Geben Sie einen Wert für die Mindest-TTL im CloudFront Cache-Verhalten an.

- Den Standardwert von 24 Stunden verwenden

Weitere Informationen finden Sie unter Verwalten der Dauer, die Inhalte im Cache bleiben (Ablauf).

Client-IP-Adressen

Wenn ein Viewer eine Anfrage an einen Anforderungsheader sendet CloudFront und keinen `X-Forwarded-For` Anforderungsheader enthält, CloudFront ruft er die IP-Adresse des Viewers von der TCP-Verbindung ab, fügt einen `X-Forwarded-For` Header hinzu, der die IP-Adresse enthält, und leitet die Anfrage an den Ursprung weiter. Wenn CloudFront z. B. die IP-Adresse `192.0.2.2` von der TCP-Verbindung abruft, wird der folgende Header an den Ursprungs-Server weitergeleitet:

```
X-Forwarded-For: 192.0.2.2
```

Wenn ein Viewer eine Anfrage an einen Request-Header sendet CloudFront und einen `X-Forwarded-For` Request-Header einschließt, CloudFront ruft er die IP-Adresse des Viewers von der TCP-Verbindung ab, hängt sie an das Ende des `X-Forwarded-For` Headers an und leitet die Anfrage an den Ursprung weiter. Wenn die Viewer-Anfrage beispielsweise die IP-Adresse der TCP-Verbindung enthält `X-Forwarded-For: 192.0.2.4,192.0.2.3` und `192.0.2.2` von ihr CloudFront abruft, leitet sie den folgenden Header an den Ursprung weiter:

```
X-Forwarded-For: 192.0.2.4,192.0.2.3,192.0.2.2
```

Einige Anwendungen, wie Load Balancer (einschließlich Elastic Load Balancing), Web Application Firewalls, Reverse-Proxys, Intrusion Prevention-Systeme und API Gateway, fügen die IP-Adresse des CloudFront Edge-Servers, der die Anfrage weitergeleitet hat, an das Ende des Headers an. `X-Forwarded-For` Wenn beispielsweise `X-Forwarded-For: 192.0.2.2` in einer Anfrage CloudFront enthalten ist, dass sie an ELB weiterleitet, und wenn die IP-Adresse des CloudFront Edge-Servers 192.0.2.199 lautet, enthält die Anforderung, die Ihre EC2-Instance empfängt, den folgenden Header:

```
X-Forwarded-For: 192.0.2.2,192.0.2.199
```

> **Note**
>
> Der `X-Forwarded-For`-Header enthält IPv4-Adressen (z. B. 192.0.2.44) und IPv6-Adressen (z. B. 2001:0db8:85a3::8a2e:0370:7334).
> Verwenden Sie beim Analysieren von IPv6-Adressen im `X-Forwarded-For` Header Standardbibliotheken zum Analysieren von IP-Adressen, die jedes gültige RFC 4291-IPv6-Format verarbeiten können.

Beachten Sie auch, dass der `X-Forwarded-For` Header von jedem Knoten auf dem Pfad zum aktuellen Server geändert werden kann (). CloudFront Weitere Informationen finden Sie im Abschnitt 8.1 unter RFC 7239. Sie können den Header auch mithilfe von CloudFront Edge-Berechnungsfunktionen ändern.

Client-side SSL-Authentifizierung

CloudFront unterstützt die gegenseitige TLS-Authentifizierung (mTLS), bei der sowohl der Client als auch der Server sich gegenseitig mithilfe von Zertifikaten authentifizieren. Wenn mTLS konfiguriert ist, CloudFront kann es Client-Zertifikate während des TLS-Handshakes validieren und optional CloudFront Funktionen ausführen, um eine benutzerdefinierte Validierungslogik zu implementieren.

Bei Ursprüngen, die clientseitige Zertifikate anfordern, wenn mTLS nicht konfiguriert ist, wird die Anfrage gelöscht. CloudFront

Weitere Informationen zur Konfiguration von mTLS finden Sie unter. ???

CloudFront unterstützt keine Client-Authentifizierung mit clientseitigen SSL-Zertifikaten. Wenn ein Ursprung ein clientseitiges Zertifikat anfordert, CloudFront wird die Anfrage gelöscht.

Komprimierung

Weitere Informationen finden Sie unter Bereitstellen von komprimierten Dateien.

Bedingte Anforderungen

Wenn CloudFront er von einem Edge-Cache eine Anforderung für ein Objekt erhält, das abgelaufen ist, leitet er die Anforderung an den Ursprung weiter, um entweder die neueste Version des Objekts abzurufen oder um vom Ursprung eine Bestätigung zu erhalten, dass der CloudFront Edge-Cache bereits über die neueste Version verfügt. In der Regel enthielt die Antwort einen Wert CloudFront, einen `ETag` Wert oder beide Werte, als der Ursprung das Objekt das letzte Mal gesendet hat. `LastModified` Fügt in der neuen Anfrage CloudFront , die an den Ursprung weitergeleitet wird, eines oder beide der folgenden Elemente CloudFront hinzu:

- Einen `If-Match`- oder `If-None-Match`-Header mit dem `ETag`-Wert für die abgelaufene Version des Objekts
- Einen `If-Modified-Since`-Header mit dem `LastModified`-Wert für die abgelaufene Version des Objekts

Der Ursprung verwendet diese Informationen, um zu ermitteln, ob das Objekt aktualisiert wurde und ob daher das gesamte Objekt an einen HTTP 304-Statuscode (nicht geändert) zurückgegeben werden soll CloudFront oder nur an einen HTTP 304-Statuscode zurückgegeben werden soll.

> **Note**
>
> `If-Modified-Since`und `If-None-Match` bedingte Anfragen werden nicht unterstützt, wenn die Konfiguration so konfiguriert CloudFront ist, dass Cookies weitergeleitet werden (alle oder eine Teilmenge).
> Weitere Informationen finden Sie unter Zwischenspeichern von Inhalten auf der Grundlage von Cookies.

Cookies

Sie können so konfigurieren CloudFront , dass Cookies an Ihren Ursprung weitergeleitet werden. Weitere Informationen finden Sie unter Zwischenspeichern von Inhalten auf der Grundlage von Cookies.

Cross-origin gemeinsame Nutzung von Ressourcen (CORS)

Wenn Sie die Einstellungen CloudFront für die ursprungsübergreifende gemeinsame Nutzung von Ressourcen berücksichtigen möchten, konfigurieren Sie, CloudFront dass der `Origin` Header an Ihren Ursprung weitergeleitet wird. Weitere Informationen finden Sie unter Zwischenspeichern von Inhalten auf der Grundlage von Anforderungsheadern.

Verschlüsselung

Sie können festlegen, dass Zuschauer HTTPS verwenden, um Anfragen an Ihren benutzerdefinierten Ursprung zu senden CloudFront, CloudFront und Anfragen an Ihren benutzerdefinierten Ursprung weiterleiten müssen, indem Sie das Protokoll verwenden, das vom Viewer verwendet wird. Weitere Informationen finden Sie in den folgenden Verteilungseinstellungen:

- Viewer-Protokollrichtlinien
- Protokoll (nur benutzerdefinierte Ursprünge)

CloudFront leitet HTTPS-Anfragen mithilfe der Protokolle SSLv3, TLSv1.0 TLSv1.1, TLSv1.2 und an den Ursprungsserver weiter. TLSv1.3 Für benutzerdefinierte Ursprünge können Sie die SSL-

Protokolle auswählen, die Sie für die Kommunikation mit CloudFront Ihrem Absender verwenden möchten:

- Wenn Sie die CloudFront Konsole verwenden, wählen Sie Protokolle aus, indem Sie die Kontrollkästchen Origin SSL Protocols verwenden. Weitere Informationen finden Sie unter Eine Distribution erstellen.
- Wenn Sie die CloudFront API verwenden, geben Sie Protokolle mithilfe des `OriginSslProtocols` Elements an. Weitere Informationen finden Sie unter OriginSslProtocols und DistributionConfig in der CloudFront Amazon-API-Referenz.

Wenn der Ursprung ein Amazon S3-Bucket ist, CloudFront wird standardmäßig verwendet TLSv1.3.

> **Important**
>
> Andere Versionen von SSL und TLS werden nicht unterstützt.

Weitere Informationen zur Verwendung von HTTPS mit CloudFront finden Sie unterVerwenden Sie HTTPS mit CloudFront. Eine Liste der Chiffren, die die HTTPS-Kommunikation zwischen Zuschauern und sowie zwischen CloudFront und Ihrem Absender CloudFront unterstützen CloudFront, finden Sie unter. Unterstützte Protokolle und Chiffren zwischen Viewern und CloudFront

GET-Anfragen mit Anfragetext

Wenn eine `GET` Viewer-Anfrage einen Text enthält, wird der HTTP-Statuscode 403 (Forbidden) an den Betrachter CloudFront zurückgegeben.

HTTP-Methoden

Wenn Sie die Konfiguration so konfigurieren CloudFront , dass alle unterstützten HTTP-Methoden verarbeitet werden, CloudFront akzeptiert es die folgenden Anfragen von Viewern und leitet sie an Ihren benutzerdefinierten Ursprung weiter:

- `DELETE`
- `GET`
- `HEAD`
- `OPTIONS`
- `PATCH`

- `POST`
- `PUT`

CloudFront speichert Antworten auf `GET` und `HEAD` Anfragen immer im Cache. Sie können auch so konfigurieren CloudFront , dass Antworten auf `OPTIONS` Anfragen zwischengespeichert werden. CloudFront speichert Antworten auf Anfragen, die die anderen Methoden verwenden, nicht im Cache.

Informationen zur Konfiguration Ihres benutzerdefinierten Ursprungsservers für die Verarbeitung dieser Methoden finden Sie in der Dokumentation zu Ihrem Ursprungsserver.

> **Important**
>
> Wenn Sie so konfigurieren CloudFront , dass alle CloudFront unterstützten HTTP-Methoden akzeptiert und an Ihren Ursprung weitergeleitet werden, konfigurieren Sie Ihren Ursprungsserver so, dass er alle Methoden verarbeitet. Wenn Sie beispielsweise so konfigurieren CloudFront , dass diese Methoden akzeptiert und weitergeleitet werden, weil Sie sie verwenden möchten`POST`, müssen Sie Ihren Ursprungsserver so konfigurieren, dass `DELETE` Anfragen angemessen verarbeitet werden, sodass Zuschauer keine Ressourcen löschen können, die Sie nicht möchten. Weitere Informationen finden Sie in der Dokumentation zu Ihrem HTTP-Server.

Header und CloudFront Verhalten von HTTP-Anfragen (benutzerdefiniert und Amazon S3-Ursprünge)

In der folgenden Tabelle sind HTTP-Anfrage-Header aufgelistet, die Sie sowohl an benutzerdefinierte als auch Amazon S3-Ursprünge weiterleiten können (mit Ausnahmen, auf die hingewiesen wird). Für jeden Header umfasst die Tabelle Informationen über Folgendes:

- CloudFront Verhalten, wenn Sie nicht so konfigurieren CloudFront , dass der Header an Ihren Ursprung weitergeleitet wird, was dazu führt CloudFront , dass Ihre Objekte auf der Grundlage von Header-Werten zwischengespeichert werden.
- Ob Sie so konfigurieren können CloudFront , dass Objekte auf der Grundlage von Header-Werten für diesen Header zwischengespeichert werden.

  Sie können so konfigurieren CloudFront , dass Objekte auf der Grundlage von Werten in den `User-Agent` Headern `Date` und zwischengespeichert werden, wir empfehlen dies jedoch nicht.

Diese Header haben viele mögliche Werte, und das Caching auf der Grundlage ihrer Werte würde dazu führen, dass deutlich mehr Anfragen CloudFront an Ihren Ursprung weitergeleitet werden.

Weitere Informationen zum Zwischenspeichern auf Basis von Header-Werten finden Sie unter Zwischenspeichern von Inhalten auf der Grundlage von Anforderungsheadern.

Header	Verhalten, wenn Sie nicht so konfigurieren, dass das Zwischenspeichern CloudFront auf der Grundlage von Header-Werten erfolgt	Das Zwischenspeichern auf Basis von Header-Werten wird unterstützt
Other-defined Header	Ältere Cache-Einstellungen — CloudFront leitet die Header an Ihren Ursprung weiter.	Ja
`Accept`	CloudFront entfernt den Header.	Ja
`Accept-Charset`	CloudFront entfernt den Header.	Ja
`Accept-Encoding`	Wenn der Wert `gzip` oder enthält`br`, CloudFront leitet er einen normalisierten `Accept-Encoding` Header an Ihren Ursprung weiter. Weitere Informationen erhalten Sie unter Komprimierungsunterstützung und Bereitstellen von komprimierten Dateien.	Ja
`Accept-Language`	CloudFront entfernt den Header.	Ja
`Authorization`	•	Ja

Header	Verhalten, wenn Sie nicht so konfigurieren, dass das Zwischenspeichern CloudFront auf der Grundlage von Header-Werten erfolgt	Das Zwischenspeichern auf Basis von Header-Werten wird unterstützt
	`GET`und `HEAD` Anfragen — CloudFront entfernt das `Authorization` Header-Feld, bevor die Anfrage an Ihren Ursprung weitergeleitet wird. • `OPTIONS`Anfragen — CloudFront entfernt das `Authorization` Header-Feld, bevor die Anfrage an Ihren Ursprung weitergeleitet wird, wenn Sie so konfigurieren CloudFront , dass Antworten auf `OPTIONS` Anfragen zwischeng espeichert werden. CloudFront leitet das `Authorization` Header-Feld an Ihren Ursprung weiter, wenn Sie nicht so konfigurierenCloudFront , dass Antworten auf OPTIONS-Anfragen zwischengespeichert werden. • `DELETE`, `PATCH``POST`, und `PUT` Anfragen — entfernt das Header-Feld CloudFront nicht, bevor die Anfrage an Ihren Ursprung weitergeleitet wird.	
`Cache-Control`	CloudFront leitet den Header an Ihren Ursprung weiter.	Nein

Header	Verhalten, wenn Sie nicht so konfigurieren, dass das Zwischenspeichern CloudFront auf der Grundlage von Header-Werten erfolgt	Das Zwischenspeichern auf Basis von Header-Werten wird unterstützt
`CloudFront-Forwarded-Proto`	CloudFront fügt den Header nicht hinzu, bevor die Anfrage an Ihren Ursprung weitergeleitet wird. Weitere Informationen finden Sie unter Konfigurieren der Zwischenspeicherung basierend auf dem Protokoll der Anforderung.	Ja
`CloudFront-Is-Desktop-Viewer`	CloudFront fügt den Header nicht hinzu, bevor die Anfrage an Ihren Ursprung weitergeleitet wird. Weitere Informationen finden Sie unter Konfigurieren der Zwischenspeicherung basierend auf dem Gerätetyp.	Ja
`CloudFront-Is-Mobile-Viewer`	CloudFront fügt den Header nicht hinzu, bevor die Anfrage an Ihren Ursprung weitergeleitet wird. Weitere Informationen finden Sie unter Konfigurieren der Zwischenspeicherung basierend auf dem Gerätetyp.	Ja
`CloudFront-Is-Tablet-Viewer`	CloudFront fügt den Header nicht hinzu, bevor die Anfrage an Ihren Ursprung weitergeleitet wird. Weitere Informationen finden Sie unter Konfigurieren der Zwischenspeicherung basierend auf dem Gerätetyp.	Ja

Header	Verhalten, wenn Sie nicht so konfigurieren, dass das Zwischenspeichern CloudFront auf der Grundlage von Header-Werten erfolgt	Das Zwischenspeichern auf Basis von Header-Werten wird unterstützt
`CloudFront-Viewer-Country`	CloudFront fügt den Header nicht hinzu, bevor die Anfrage an Ihren Ursprung weitergeleitet wird.	Ja
`Connection`	CloudFront ersetzt diesen Header durch `Connection: Keep-Alive` bevor die Anfrage an Ihren Ursprung weitergeleitet wird.	Nein
`Content-Length`	CloudFront leitet den Header an Ihren Ursprung weiter.	Nein
`Content-MD5`	CloudFront leitet den Header an Ihren Ursprung weiter.	Ja
`Content-Type`	CloudFront leitet den Header an Ihren Ursprung weiter.	Ja
`Cookie`	Wenn Sie CloudFront die Weiterleitung von Cookies konfigurieren, wird das `Cookie` Header-Feld an Ihren Ursprung weitergeleitet. Wenn Sie dies nicht tun, CloudFront wird das `Cookie` Header-Feld entfernt. Weitere Informationen finden Sie unter Zwischenspeichern von Inhalten auf der Grundlage von Cookies.	Nein

Header	Verhalten, wenn Sie nicht so konfigurieren, dass das Zwischenspeichern CloudFront auf der Grundlage von Header-Werten erfolgt	Das Zwischenspeichern auf Basis von Header-Werten wird unterstützt
`Date`	CloudFront leitet den Header an Ihren Ursprung weiter.	Ja, wird aber nicht empfohlen
`Expect`	CloudFront entfernt den Header.	Ja
`From`	CloudFront leitet den Header an Ihren Ursprung weiter.	Ja
`Host`	CloudFront setzt den Wert auf den Domainnamen des Ursprungs, der dem angeforderten Objekt zugeordnet ist. Sie können nicht auf der Grundlage des Host-Headers für Amazon S3 oder MediaStore Origins zwischenspeichern.	Ja (benutzerdefiniert) Nein (S3 und MediaStore)
`If-Match`	CloudFront leitet den Header an Ihren Ursprung weiter.	Ja
`If-Modified-Since`	CloudFront leitet den Header an Ihren Ursprung weiter.	Ja
`If-None-Match`	CloudFront leitet den Header an Ihren Ursprung weiter.	Ja

Header	Verhalten, wenn Sie nicht so konfigurieren, dass das Zwischenspeichern CloudFront auf der Grundlage von Header-Werten erfolgt	Das Zwischenspeichern auf Basis von Header-Werten wird unterstützt
`If-Range`	CloudFront leitet den Header an Ihren Ursprung weiter.	Ja
`If-Unmodified-Since`	CloudFront leitet den Header an Ihren Ursprung weiter.	Ja
`Max-Forwards`	CloudFront leitet den Header an Ihren Ursprung weiter.	Nein
`Origin`	CloudFront leitet den Header an Ihren Ursprung weiter.	Ja
`Pragma`	CloudFront leitet den Header an Ihren Ursprung weiter.	Nein
`Proxy-Authenticate`	CloudFront entfernt den Header.	Nein
`Proxy-Authorization`	CloudFront entfernt den Header.	Nein
`Proxy-Connection`	CloudFront entfernt den Header.	Nein

Header	Verhalten, wenn Sie nicht so konfigurieren, dass das Zwischenspeichern CloudFront auf der Grundlage von Header-Werten erfolgt	Das Zwischenspeichern auf Basis von Header-Werten wird unterstützt
`Range`	CloudFront leitet den Header an Ihren Ursprung weiter. Weitere Informationen finden Sie unter Wie CloudFront verarbeitet Teilanfragen für ein Objekt (Bereichs-GETs).	Ja, standardmäßig
`Referer`	CloudFront entfernt den Header.	Ja
`Request-Range`	CloudFront leitet den Header an Ihren Ursprung weiter.	Nein
`TE`	CloudFront entfernt den Header.	Nein
`Trailer`	CloudFront entfernt den Header.	Nein
`Transfer-Encoding`	CloudFront leitet den Header an Ihren Ursprung weiter.	Nein
`Upgrade`	CloudFront entfernt den Header, es sei denn, Sie haben eine WebSocket Verbindung hergestellt.	Nein (außer für WebSocket Verbindungen)

Header	Verhalten, wenn Sie nicht so konfigurieren, dass das Zwischenspeichern CloudFront auf der Grundlage von Header-Werten erfolgt	Das Zwischens peichern auf Basis von Header-We rten wird unterstützt
`User-Agent`	CloudFront ersetzt den Wert dieses Header-Feldes durch`Amazon CloudFront` . Wenn CloudFront Sie Ihre Inhalte auf der Grundlage des Geräts, das der Benutzer verwendet, zwischens peichern möchten, finden Sie weitere Informati onen unterKonfigurieren der Zwischenspeicherung basierend auf dem Gerätetyp.	Ja, wird aber nicht empfohlen
`Via`	CloudFront leitet den Header an Ihren Ursprung weiter.	Ja
`Warning`	CloudFront leitet den Header an Ihren Ursprung weiter.	Ja
`X-Amz-Cf-Id`	CloudFront fügt den Header zur Viewer-Anfrage hinzu, bevor die Anfrage an Ihren Ursprung weitergeleitet wird. Der Header-Wert enthält eine verschlüsselte Zeichenfolge, die die Anfrage eindeutig bezeichnet.	Nein
`X-Edge-*`	CloudFront entfernt alle `X-Edge-*` Header.	Nein
`X-Forwarded-For`	CloudFront leitet den Header an deinen Ursprung weiter. Weitere Informationen finden Sie unter Client-IP-Adressen.	Ja

Header	Verhalten, wenn Sie nicht so konfigurieren, dass das Zwischenspeichern CloudFront auf der Grundlage von Header-Werten erfolgt	Das Zwischenspeichern auf Basis von Header-Werten wird unterstützt
`X-Forwarded-Proto`	CloudFront entfernt den Header.	Nein
`X-HTTP-Method-Override`	CloudFront entfernt den Header.	Ja
`X-Real-IP`	CloudFront entfernt den Header.	Nein

HTTP-Version

CloudFront leitet Anfragen an Ihren benutzerdefinierten Ursprung weiter mit HTTP/1.1.

Maximale Länge einer Anfrage und maximale Länge einer URL

Die maximale Länge einer Anfrage, einschließlich des Pfads, der Abfragezeichenfolge (falls vorhanden) und der Header, beträgt 32.768 Byte.

CloudFront erstellt eine URL aus der Anfrage. Die maximale Länge dieser URL beträgt 8 192 Byte.

Wenn eine URL die maximale Länge überschreitet, wird der HTTP-Statuscode 414 (URI Too Long) an den Viewer CloudFront zurückgegeben. Wenn eine Anforderung die maximale Länge überschreitet, weil die Header-Größe überschritten wird, wird der HTTP-Statuscode 494 an den Viewer CloudFront zurückgegeben. In beiden Fällen CloudFront wird dann die TCP-Verbindung zum Viewer beendet.

OCSP-Stapling

Wenn ein Betrachter eine HTTPS-Anfrage für ein Objekt sendet, muss entweder CloudFront oder der Betrachter bei der Zertifizierungsstelle (CA) bestätigen, dass das SSL-Zertifikat für die Domain nicht

gesperrt wurde. OCSP-Stapling beschleunigt die Zertifikatsvalidierung, da CloudFront das Zertifikat validiert und die Antwort der CA zwischengespeichert werden kann, sodass der Client das Zertifikat nicht direkt bei der CA validieren muss.

Die Leistungssteigerung durch OCSP-Stapling ist deutlicher spürbar, wenn CloudFront viele HTTPS-Anfragen für Objekte in derselben Domain erhält. Jeder Server an einem CloudFront Edge-Standort muss eine separate Validierungsanforderung einreichen. Wenn CloudFront viele HTTPS-Anfragen für dieselbe Domain erhält, hat jeder Server an dem Edge-Standort nach kurzer Zeit hat eine Antwort von der CA vorliegen, die er an ein Paket im SSL-Handshake „heften" (engl. „to staple") kann; wenn der Betrachter von der Gültigkeit des Zertifikats überzeugt ist, kann CloudFront das angeforderte Objekt übertragen. Wenn Ihre Distribution an einem CloudFront Edge-Standort nicht viel Verkehr empfängt, ist es wahrscheinlicher, dass neue Anfragen an einen Server weitergeleitet werden, der das Zertifikat noch nicht von der CA validiert hat. In diesem Fall führt der Viewer den Validierungsschritt separat durch und der CloudFront Server stellt das Objekt bereit. Dieser CloudFront Server sendet auch eine Validierungsanforderung an die CA. Wenn er also das nächste Mal eine Anfrage erhält, die denselben Domainnamen enthält, erhält er eine Validierungsantwort von der CA.

Persistente Verbindungen

Wenn CloudFront er eine Antwort von Ihrem Absender erhält, versucht er, die Verbindung für einige Sekunden aufrechtzuerhalten, falls in diesem Zeitraum eine weitere Anfrage eintrifft. Durch eine persistente Verbindung wird die Zeit gespart, die erforderlich ist, um die TCP-Verbindung erneut herzustellen und einen weiteren TLS-Handshake für nachfolgende Anforderungen durchzuführen.

Weitere Informationen, einschließlich solcher zur Konfiguration der Dauer ständiger Verbindungen, finden Sie unter Keep-alive Timeout (nur benutzerdefinierte und VPC-Ursprünge) im Abschnitt Referenz für alle Distributionseinstellungen.

Protokolle

CloudFront leitet HTTP- oder HTTPS-Anfragen auf der Grundlage der folgenden Kriterien an den Ursprungsserver weiter:

- Das Protokoll der Anfrage, an die der Viewer sendet CloudFront, entweder HTTP oder HTTPS.
- Der Wert des Felds Origin Protocol Policy in der CloudFront Konsole oder, wenn Sie die CloudFront API verwenden, das `OriginProtocolPolicy` Element im `DistributionConfig` komplexen Typ. In der CloudFront Konsole lauten die Optionen „Nur HTTP", „Nur HTTPS" und „Match Viewer".

Wenn Sie „Nur HTTP“ oder „Nur HTTPS“ angeben, CloudFront leitet Anfragen mithilfe des angegebenen Protokolls an den Ursprungsserver weiter, unabhängig vom Protokoll in der Viewer-Anfrage.

Wenn Sie Match Viewer angeben, CloudFront leitet es Anfragen mithilfe des Protokolls in der Viewer-Anfrage an den Originalserver weiter. Beachten Sie, dass CloudFront das Objekt nur einmal zwischenspeichert, auch wenn Betrachter ihre Anfragen sowohl über HTTP als auch über HTTPS übertragen.

> **Important**
>
> CloudFront Leitet eine Anfrage mithilfe des HTTPS-Protokolls an den Ursprung weiter und wenn der Ursprungsserver ein ungültiges Zertifikat oder ein selbstsigniertes Zertifikat zurückgibt, wird die TCP-Verbindung CloudFront unterbrochen.

Hinweise zum Aktualisieren einer Distribution mithilfe der CloudFront Konsole finden Sie unter. Eine Verteilung aktualisieren Informationen zur Aktualisierung einer Distribution mithilfe der CloudFront API finden Sie UpdateDistribution in der CloudFront Amazon-API-Referenz.

Abfragezeichenfolgen

Sie können konfigurieren, ob Abfragezeichenfolgenparameter an Ihren Ursprung CloudFront weitergeleitet werden sollen. Weitere Informationen finden Sie unter Zwischenspeichern von Inhalten auf der Grundlage von Abfragezeichenfolgeparametern.

Timeout der Ursprungsverbindung und Verbindungsversuche zum Ursprung

Das Timeout für die Ursprungsverbindung gibt an, wie viele Sekunden CloudFront gewartet werden, wenn versucht wird, eine Verbindung zum Ursprung herzustellen.

Die Anzahl der Versuche, eine Verbindung zum Ursprung herzustellen, gibt an, wie oft CloudFront versucht wurde, eine Verbindung zum Ursprung herzustellen.

Zusammen bestimmen diese Einstellungen, wie lange CloudFront versucht wird, eine Verbindung zum Ursprung herzustellen, bevor ein Failover zum sekundären Ursprung erfolgt (im Fall einer Ursprungsgruppe) oder eine Fehlermeldung an den Viewer zurückgegeben wird. CloudFront Wartet standardmäßig bis zu 30 Sekunden (3 Versuche von jeweils 10 Sekunden), bevor versucht wird, eine Verbindung zum sekundären Ursprung herzustellen, oder eine Fehlermeldung zurückgegeben wird.

Sie können diese Zeit reduzieren, indem Sie ein kürzeres Verbindungs-Timeout, weniger Versuche oder beides angeben.

Weitere Informationen finden Sie unter Steuern von Timeouts und Verbindungsversuchen für Ursprünge.

Ursprungs-Reaktions-Timeout

Das Ursprungs-Reaktions-Timeout, das auch als Ursprungs-Lese-Timeout oder Ursprungs-Anforderungs-Timeout bezeichnet wird, gilt für Folgendes:

- Die Zeitspanne in Sekunden, die auf eine Antwort CloudFront wartet, nachdem eine Anfrage an den Ursprung weitergeleitet wurde.
- Die Zeitdauer in Sekunden, die nach dem Empfang eines Pakets mit einer Antwort vom Ursprung und vor dem Empfang des nächsten Pakets CloudFront gewartet wird.

CloudFront Das Verhalten hängt von der HTTP-Methode der Viewer-Anfrage ab:

- `GET`und `HEAD` Anfragen — Wenn der Absender nicht reagiert oder innerhalb der Dauer des Antwort-Timeouts nicht mehr reagiert, wird die CloudFront Verbindung unterbrochen. Wenn die angegebene Anzahl von Verbindungsversuchen über der Quelle mehr als 1 ist, versuchen CloudFront Sie erneut, eine vollständige Antwort zu erhalten. CloudFront versucht es bis zu dreimal, abhängig vom Wert der Einstellung für die ursprünglichen Verbindungsversuche. Wenn der Ursprung beim letzten Versuch keine Antwort sendet, unternimmt CloudFront erst dann einen weiteren Versuch, wenn die nächste Anfrage für Inhalte auf demselben Ursprung empfangen wird.
- `DELETE`, `OPTIONS` `PATCH``PUT`, und `POST` Anfragen — Wenn der Absender für die Dauer des Lesezeitlimits nicht reagiert, CloudFront wird die Verbindung unterbrochen und es wird nicht erneut versucht, den Ursprung zu kontaktieren. Der Client kann die Anfrage erneut senden, falls erforderlich.

Weitere Informationen, einschließlich Informationen zum Konfigurieren des Reaktions-Timeouts für den Ursprungs-Server, finden Sie unter Antwort-Timeout.

Gleichzeitige Anfragen für dasselbe Objekt (Zusammenfassung von Anfragen)

Wenn ein CloudFront Edge-Standort eine Anfrage für ein Objekt erhält und sich das Objekt nicht im Cache befindet oder das zwischengespeicherte Objekt abgelaufen ist, wird die Anfrage CloudFront

sofort an den Ursprung gesendet. Wenn jedoch gleichzeitig Anfragen für dasselbe Objekt eingehen — das heißt, wenn zusätzliche Anfragen für dasselbe Objekt (mit demselben Cacheschlüssel) an der Edge-Position ankommen, bevor die Antwort auf die erste Anfrage CloudFront empfangen wird — wird eine CloudFront Pause eingelegt, bevor die zusätzlichen Anfragen an den Ursprung weitergeleitet werden. Diese kurze Pause trägt dazu bei, die Belastung des Ursprungs zu verringern. CloudFront sendet die Antwort von der ursprünglichen Anfrage auf alle Anfragen, die es während der Pause erhalten hat. Dies wird als Request Collapsing (Zusammenfassung von Anfragen) bezeichnet. In CloudFront Protokollen wird die erste Anfrage `Miss` im `x-edge-result-type` Feld als a identifiziert, und die reduzierten Anfragen werden als a identifiziert. `Hit` Weitere Hinweise zu CloudFront Protokollen finden Sie unterthe section called “CloudFront und Edge-Funktionsprotokollierung”.

CloudFront blendet nur Anfragen aus, die einen Cacheschlüssel gemeinsam verwenden. Wenn die zusätzlichen Anfragen nicht denselben Cacheschlüssel verwenden, weil Sie beispielsweise so konfiguriert CloudFront haben, dass das Zwischenspeichern auf der Grundlage von Anforderungsheadern, Cookies oder Abfragezeichenfolgen erfolgt, CloudFront leitet alle Anfragen mit einem eindeutigen Cacheschlüssel an Ihren Ursprung weiter.

Um die Reduzierung von Anforderungen insgesamt zu verhindern, können Sie die verwaltete Cache-Richtlinie `CachingDisabled` verwenden, die auch das Caching verhindert. Weitere Informationen finden Sie unter Verwenden verwalteter Cache-Richtlinien.

Wenn Sie eine Reduzierung von Anforderungen für bestimmte Objekte verhindern möchten, können Sie die minimale TTL für das Cacheverhalten auf 0 setzen und den Ursprung so konfigurieren, dass `Cache-Control: private`, `Cache-Control: no-store`, `Cache-Control: no-cache`, `Cache-Control: max-age=0` oder `Cache-Control: s-maxage=0` gesendet wird. Diese Konfigurationen erhöhen die Belastung Ihres Ursprungs und führen zu zusätzlicher Latenz für die gleichzeitigen Anfragen, die unterbrochen werden, während auf die Antwort auf die CloudFront erste Anfrage gewartet wird.

`User-Agent`-Header

Wenn Sie je CloudFront nach Gerät, das ein Benutzer zum Anzeigen Ihrer Inhalte verwendet, verschiedene Versionen Ihrer Objekte zwischenspeichern möchten, empfehlen wir Ihnen, die Konfiguration so CloudFront zu konfigurieren, dass eine oder mehrere der folgenden Header an Ihren benutzerdefinierten Ursprung weitergeleitet werden:

- `CloudFront-Is-Desktop-Viewer`
- `CloudFront-Is-Mobile-Viewer`

- `CloudFront-Is-SmartTV-Viewer`
- `CloudFront-Is-Tablet-Viewer`

CloudFront Legt basierend auf dem Wert des `User-Agent` Headers den Wert dieser Header auf `true` oder `false` vor der Weiterleitung der Anfrage an Ihren Ursprung fest. Wenn ein Gerät in mehr als eine Kategorie fällt, können mehrere Werte sei `true`. Beispielsweise setzt CloudFront bei einigen Tablet-Geräten möglicherweise `CloudFront-Is-Mobile-Viewer` und `CloudFront-Is-Tablet-Viewer` auf `true`. Weitere Hinweise zur Konfiguration des Caches CloudFront auf der Grundlage von Anforderungsheadern finden Sie unter. Zwischenspeichern von Inhalten auf der Grundlage von Anforderungsheadern

Sie können so konfigurieren CloudFront , dass Objekte auf der Grundlage von Werten im `User-Agent` Header zwischengespeichert werden, wir empfehlen dies jedoch nicht. Der `User-Agent` Header hat viele mögliche Werte, und das Caching auf der Grundlage dieser Werte würde CloudFront dazu führen, dass deutlich mehr Anfragen an Ihren Ursprung weitergeleitet werden.

Wenn Sie nicht so konfigurieren CloudFront , dass Objekte auf der Grundlage von Werten im `User-Agent` Header zwischengespeichert werden, CloudFront fügen Sie einen `User-Agent` Header mit dem folgenden Wert hinzu, bevor eine Anfrage an Ihren Ursprung weitergeleitet wird:

```
User-Agent = Amazon CloudFront
```

CloudFront fügt diesen Header hinzu, unabhängig davon, ob die Anfrage des Viewers einen `User-Agent` Header enthält. Wenn die Anfrage vom Viewer einen `User-Agent` Header enthält, CloudFront wird dieser entfernt.

Wie CloudFront verarbeitet Antworten von Ihrem benutzerdefinierten Ursprung

Erfahren Sie, wie Antworten aus Ihrem benutzerdefinierten Ursprung CloudFront verarbeitet werden.

Inhalt

- 100 Antworten fortsetzen
- Caching
- Abgebrochene Anfragen
- Inhaltsvereinbarung
- Cookies

- Abgebrochene TCP-Verbindungen
- HTTP-Antwort-Header, die CloudFront entfernt oder ersetzt werden
- Maximale Dateigröße, die zwischengespeichert werden kann
- Ursprung nicht verfügbar
- Umleitungen
- Transfer-Encoding-Header

`100` Antworten fortsetzen

Ihr Absender kann nicht mehr als eine 100-Continue-Antwort an CloudFront senden. CloudFront Erwartet nach der ersten 100-Continue-Antwort eine HTTP-Antwort mit 200 OK. Wenn Ihr Absender nach der ersten eine weitere 100-Continue-Antwort sendet, CloudFront wird ein Fehler zurückgegeben.

Caching

- Stellen Sie sicher, dass der Ursprungsserver gültige und genaue Werte für die `Last-Modified` Header-Felder `Date` und festlegt.
- CloudFront respektiert normalerweise einen `Cache-Control: no-cache` Header in der Antwort vom Ursprung. Eine Ausnahme von dieser Regel wird unter Gleichzeitige Anfragen für dasselbe Objekt (Zusammenfassung von Anfragen) beschrieben.

Abgebrochene Anfragen

Wenn sich ein Objekt nicht im Edge-Cache befindet und ein Viewer eine Sitzung beendet (z. B. einen Browser schließt), nachdem CloudFront er das Objekt von Ihrem Ursprung abgerufen hat, aber bevor er das angeforderte Objekt bereitstellen kann, wird das Objekt CloudFront nicht an der Edge-Position zwischengespeichert.

Inhaltsvereinbarung

Wenn Ihr Ursprung `Vary:*` in der Antwort zurückkehrt und der Wert von Minimum TTL für das entsprechende Cacheverhalten 0 ist, wird das Objekt CloudFront zwischengespeichert, aber dennoch jede nachfolgende Anfrage für das Objekt an den Ursprung weitergeleitet, um zu bestätigen, dass der Cache die neueste Version des Objekts enthält. CloudFront enthält keine bedingten Header wie oder. `If-None-Match` `If-Modified-Since` Daher gibt Ihr Ursprung das Objekt als Antwort auf jede Anfrage an zurück. CloudFront

Wenn Ihr Ursprung `Vary:*` in der Antwort zurückgegeben wird und wenn der Wert von Minimum TTL für das entsprechende Cacheverhalten ein anderer Wert ist, CloudFront verarbeitet der `Vary` Header wie unter beschriebenHTTP-Antwort-Header, die CloudFront entfernt oder ersetzt werden.

Cookies

Wenn Sie Cookies für ein Cache-Verhalten aktivieren und der Ursprung Cookies mit einem Objekt zurückgibt, werden sowohl das Objekt als auch die Cookies CloudFront zwischengespeichert. Beachten Sie, dass diese Vorgehensweise die Zwischenspeicherbarkeit für ein Objekt reduziert. Weitere Informationen finden Sie unter Zwischenspeichern von Inhalten auf der Grundlage von Cookies.

Abgebrochene TCP-Verbindungen

Wenn die TCP-Verbindung zwischen CloudFront und Ihrem Ursprung unterbrochen wird, während Ihr Ursprung ein Objekt zurückgibt CloudFront, hängt CloudFront das Verhalten davon ab, ob Ihr Ursprung einen `Content-Length` Header in der Antwort enthalten hat:

- Content-Length Header — CloudFront gibt das Objekt an den Viewer zurück, sobald er das Objekt von Ihrem Ursprung erhält. Wenn der Wert des `Content-Length` Headers jedoch nicht der Größe des Objekts entspricht, wird das Objekt CloudFront nicht zwischengespeichert.
- Transfer-Encoding: Chunked — CloudFront gibt das Objekt an den Viewer zurück, sobald er das Objekt von Ihrem Ursprung abgerufen hat. Wenn die Chunk-Antwort jedoch nicht vollständig ist, wird CloudFront das Objekt nicht zwischengespeichert.
- Kein Content-Length Header — CloudFront gibt das Objekt an den Viewer zurück und speichert es im Cache, aber das Objekt ist möglicherweise nicht vollständig. Ohne einen `Content-Length` Header kann CloudFront nicht bestimmen, ob die TCP-Verbindung versehentlich oder absichtlich verworfen wurde.

Wir empfehlen, dass Sie Ihren HTTP-Server so konfigurieren, dass er einen `Content-Length` Header hinzufügt, um zu CloudFront verhindern, dass Teilobjekte zwischengespeichert werden.

HTTP-Antwort-Header, die CloudFront entfernt oder ersetzt werden

CloudFront entfernt oder aktualisiert die folgenden Header-Felder, bevor die Antwort von Ihrem Ursprung an den Viewer weitergeleitet wird:

- `Set-Cookie`— Wenn Sie CloudFront die Weiterleitung von Cookies konfigurieren, wird das `Set-Cookie` Header-Feld an die Clients weitergeleitet. Weitere Informationen finden Sie unter Zwischenspeichern von Inhalten auf der Grundlage von Cookies.
- `Trailer`
- `Transfer-Encoding`— Wenn Ihr Ursprung dieses Header-Feld zurückgibt, CloudFront wird der Wert auf gesetzt, `chunked` bevor die Antwort an den Betrachter zurückgegeben wird.
- `Upgrade`
- `Vary` – Beachten Sie Folgendes:
  - Wenn Sie so konfigurieren CloudFront , dass einer der gerätespezifischen Header an Ihren Ursprung weitergeleitet wird (`CloudFront-Is-Desktop-Viewer`,,`CloudFront-Is-Tablet-Viewer`) `CloudFront-Is-Mobile-ViewerCloudFront-Is-SmartTV-Viewer`, und Sie Ihren Ursprung so konfigurieren, dass er CloudFront zurückkehrt CloudFront, kehrt `Vary:User-Agent` er `Vary:User-Agent` zum Viewer zurück. Weitere Informationen finden Sie unter Konfigurieren der Zwischenspeicherung basierend auf dem Gerätetyp.
  - Wenn Sie Ihren Ursprung so konfigurieren, dass er einen `Accept-Encoding` oder `Cookie` in der `Vary` Kopfzeile CloudFront enthält, werden die Werte in die Antwort an den Viewer aufgenommen.
  - Wenn Sie so konfigurieren, CloudFront dass Header an Ihren Ursprung weitergeleitet werden, und wenn Sie Ihren Ursprung so konfigurieren, dass die Header-Namen CloudFront in der Kopfzeile an den `Vary` Header CloudFront zurückgegeben werden (z. B.`Vary:Accept-Charset,Accept-Language`), wird der `Vary` Header mit diesen Werten an den Viewer zurückgegeben.
  - Hinweise dazu, wie ein Wert von `*` in der `Vary` Kopfzeile CloudFront verarbeitet wird, finden Sie unterInhaltsvereinbarung.
  - Wenn Sie Ihren Ursprung so konfigurieren, dass er andere Werte in den `Vary` Header einbezieht, CloudFront werden die Werte entfernt, bevor die Antwort an den Viewer zurückgegeben wird.
- `Via`— CloudFront setzt den Wert in der Antwort an den Zuschauer auf den folgenden Wert:

  ```
  Via: http-version alphanumeric-string.cloudfront.net (CloudFront)
  ```

  Der Wert ist beispielsweise wie folgt:

  ```
  Via: 1.1 1026589cc7887e7a0dc7827b4example.cloudfront.net (CloudFront)
  ```

Maximale Dateigröße, die zwischengespeichert werden kann

Die maximale Größe eines Antworttextes, der in seinem Cache CloudFront gespeichert wird, beträgt 50 GB. Dazu gehören auch Antworten für aufgeteilte Übertragungen, in denen kein Wert für die `Content-Length`-Kopfzeile angegeben wurde.

Sie können ein Objekt zwischenspeichern, das größer als diese Größe ist, indem Sie Bereichsanforderungen verwenden, um die Objekte in Teilen anzufordern, die jeweils 50 GB oder weniger groß sind. CloudFront CloudFrontspeichert diese Teile im Cache, da jeder von ihnen 50 GB oder weniger groß ist. Nachdem der Viewer alle Teile des Objekts abgerufen hat, kann er das ursprüngliche, größere Objekt rekonstruieren. Weitere Informationen finden Sie unter Verwenden von Bereichsanforderungen zum Zwischenspeichern großer Objekte.

Ursprung nicht verfügbar

Wenn Ihr Originalserver nicht verfügbar ist und eine Anforderung für ein Objekt CloudFront erhält, das sich im Edge-Cache befindet, aber abgelaufen ist (z. B. weil der in der `Cache-Control max-age` Anweisung angegebene Zeitraum abgelaufen ist), wird CloudFront entweder die abgelaufene Version des Objekts bereitgestellt oder eine benutzerdefinierte Fehlerseite angezeigt. Weitere Informationen zum CloudFront Verhalten bei der Konfiguration benutzerdefinierter Fehlerseiten finden Sie unterWie CloudFront verarbeitet Fehler, wenn Sie benutzerdefinierte Fehlerseiten konfiguriert haben.

In einigen Fällen wird ein Objekt, das selten angefordert wird, entfernt und ist im Edge-Cache nicht mehr verfügbar. CloudFront kann ein Objekt, das entfernt wurde, nicht bereitstellen.

Umleitungen

Wenn Sie den Speicherort eines Objekts auf dem Ursprungsserver ändern, können Sie Ihren Webserver so konfigurieren, dass Anfragen an den neuen Speicherort umgeleitet werden. Nachdem Sie die Weiterleitung konfiguriert haben, sendet ein Betrachter, wenn er zum ersten Mal eine Anfrage für das Objekt CloudFront sendet, die Anfrage an den Ursprung, und der Absender reagiert mit einer Weiterleitung (z. B.). `302 Moved Temporarily` CloudFront speichert die Weiterleitung im Cache und gibt sie an den Viewer zurück. CloudFront folgt der Weiterleitung nicht.

Sie können Ihren Webserver so konfigurieren, dass Anfragen an einen der folgenden Speicherorte umgeleitet werden:

- Die neue URL des Objekts auf dem Ursprungsserver. Wenn der Zuschauer der Weiterleitung zur neuen URL folgt, umgeht er die URL CloudFront und geht direkt zum Ursprung. Daher empfehlen wir, Anforderungen nicht an die neue URL des Objekts auf dem Ursprungsserver weiterzuleiten.

- Die neue CloudFront URL für das Objekt. Wenn der Viewer die Anfrage sendet, die die neue CloudFront URL enthält, CloudFront ruft er das Objekt von der neuen Position auf Ihrem Ursprung ab, speichert es an der Edge-Position im Cache und gibt das Objekt an den Viewer zurück. Nachfolgende Anfragen für das Objekt werden von dem Edge-Standort bedient. Dadurch werden Latenzzeiten und Arbeitslasten vermieden, die bei Viewer-Anforderungen für das Objekt an den Ursprungsserver entstehen. Allerdings fallen bei jeder neuen Anfrage für das Objekt Gebühren für zwei Anfragen an CloudFront berechnet.

`Transfer-Encoding`-Header

CloudFront unterstützt nur den `chunked` Wert des Headers. `Transfer-Encoding` Wenn Ihr Ursprung CloudFront zurückkehrt`Transfer-Encoding: chunked`, gibt das Objekt an den Client zurück, sobald es an der Edge-Position empfangen wird, und speichert das Objekt für nachfolgende Anfragen im Chunk-Format.

Wenn der Viewer eine `Range GET` Anfrage stellt und der Ursprung CloudFront zurückkehrt`Transfer-Encoding: chunked`, wird das gesamte Objekt an den Viewer zurückgegeben und nicht der angeforderte Bereich.

Wir empfehlen, dass Sie die Abschnittscodierung verwenden, wenn die Länge des Inhalts Ihrer Antwort nicht im Voraus ermittelt werden kann. Weitere Informationen finden Sie unter Abgebrochene TCP-Verbindungen.

Verhalten von Anfragen und Antworten für Ursprungsgruppen

Anfragen an eine Ursprungsgruppe funktionieren genauso wie Anfragen an einen Ursprung, der nicht als Ursprungsgruppe eingerichtet ist, außer wenn ein Ursprungs-Failover vorliegt. Wie bei jedem anderen Ursprung wird der Inhalt den Zuschauern aus dem Cache bereitgestellt, wenn CloudFront eine Anfrage eingeht und der Inhalt bereits an einem Edge-Standort zwischengespeichert ist. Wenn ein Cache-Fehler vorliegt und der Ursprung eine Ursprungsgruppe ist, werden Viewer-Anforderungen an den primären Ursprung in der Ursprungsgruppe weitergeleitet.

Das Anfrage- und Antwortverhalten für den primären Ursprung ist identisch mit dem für einen Ursprung, der nicht zu einer Ursprungsgruppe gehört. Weitere Informationen erhalten Sie unter Verhalten von Anfragen und Antworten für Amazon-S3-Ursprünge und Verhalten von Anfragen und Antworten für benutzerdefinierte Ursprungsserver.

Nachfolgend werden die Verhaltensweisen bei Origin Failover beschrieben, wenn der primäre Ursprung bestimmte HTTP-Statuscodes ausgibt:

- HTTP 2xx-Statuscode (erfolgreich): Die Datei wird CloudFront zwischengespeichert und an den Betrachter zurückgegeben.
- HTTP 3xx-Statuscode (Umleitung): CloudFront gibt den Statuscode an den Viewer zurück.
- HTTP 4xx- oder 5xx-Statuscode (client/server Fehler): Wenn der zurückgegebene Statuscode für einen Failover konfiguriert wurde, wird dieselbe Anfrage CloudFront an den sekundären Ursprung in der Ursprungsgruppe gesendet.
- HTTP 4xx- oder 5xx-Statuscode (client/server Fehler): Wenn der zurückgegebene Statuscode nicht für Failover konfiguriert wurde, wird der Fehler an den Viewer CloudFront zurückgegeben.

CloudFront Failover zum sekundären Ursprung erfolgt nur, wenn die HTTP-Methode der Viewer-Anforderung auf`GET`, `HEAD` oder gesetzt ist. `OPTIONS` CloudFront führt kein Failover durch, wenn der Viewer eine andere HTTP-Methode sendet (z. B. `POSTPUT`, usw.).

Wenn eine Anfrage an einen sekundären Ursprung CloudFront gesendet wird, ist das Antwortverhalten dasselbe wie bei einem CloudFront Ursprung, der keiner Ursprungsgruppe angehört.

Weitere Informationen zu Ursprungsgruppen finden Sie unter Optimieren Sie die Hochverfügbarkeit mit CloudFront Origin-Failover.

Hinzufügen benutzerdefinierter Header zu Ursprungsanforderungen

Sie können so konfigurieren CloudFront , dass den Anfragen, die an Ihren Ursprung gesendet werden, benutzerdefinierte Header hinzugefügt werden. Mit benutzerdefinierten Headern können Sie Informationen aus Ihrem Ursprung senden und erfassen, die Sie mit typischen Viewer-Anforderungen nicht erhalten. Sie können die Header sogar für jeden Ursprung anpassen. CloudFrontunterstützt benutzerdefinierte Header für benutzerdefinierte Ursprünge und Amazon S3-Ursprünge.

Inhalt

- Anwendungsfälle
- Konfiguriere CloudFront , um benutzerdefinierte Header zu Quellanfragen hinzuzufügen
- Benutzerdefinierte Header, die nicht zu CloudFront Quellanfragen hinzugefügt werden können
- Konfigurieren Sie CloudFront , um den Authorization-Header weiterzuleiten

Anwendungsfälle

Sie können benutzerdefinierte Header wie in den folgenden Beispielen verwenden:

Identifizierung von Anfragen von CloudFront

Sie können die Anfragen identifizieren, von denen Ihr Absender eingeht CloudFront. Das kann nützlich sein, wenn du wissen möchtest, ob Nutzer das CDN umgehen CloudFront, oder wenn du mehr als ein CDN verwendest und Informationen darüber erhalten möchtest, welche Anfragen von jedem CDN kommen.

> **Note**
>
> Wenn Sie einen Amazon S3-Ursprung verwenden und die Protokollierung des Zugriffs auf Amazon S3 aktivieren, enthalten die Protokolle keine Header-Informationen.

Bestimmen, welche Anforderungen von einer bestimmten Verteilung stammen

Wenn Sie mehrere CloudFront Distributionen so konfigurieren, dass sie denselben Ursprung verwenden, können Sie in jeder Distribution unterschiedliche benutzerdefinierte Header hinzufügen. Anschließend können Sie mit den Protokollen Ihres Ursprungs bestimmen, welche Anforderungen aus welcher CloudFront-Verteilung stammen.

Cross-Origin Resource Sharing (CORS) aktivieren

Wenn einige Ihrer Zuschauer Cross-Origin Resource Sharing (CORS) nicht unterstützen, können Sie so konfigurieren, dass der `Origin` Header immer CloudFront zu Anfragen hinzugefügt wird, die an Ihren Ursprung gesendet werden. Dann können Sie Ihren Ursprung so konfigurieren, dass der `Access-Control-Allow-Origin`-Header für jede Anforderung zurückgegeben wird. Sie müssen auch CloudFront so konfigurieren, dass die CORS-Einstellungen eingehalten werden.

Steuern des Zugriffs auf Inhalt

Mit benutzerdefinierten Header können Sie den Zugriff auf Inhalte steuern. Indem du deinen Absender so konfigurierst, dass er nur auf Anfragen reagiert, wenn sie einen benutzerdefinierten Header enthalten, der von hinzugefügt wird CloudFront, verhinderst du, dass Nutzer deine Inhalte umgehen CloudFront und direkt vom Absender aus auf deine Inhalte zugreifen. Weitere Informationen finden Sie unter Beschränken des Zugriffs auf Dateien auf benutzerdefinierten Ursprungsservern.

Konfiguriere CloudFront , um benutzerdefinierte Header zu Quellanfragen hinzuzufügen

Um eine Verteilung so zu konfigurieren, dass benutzerdefinierte Header Anforderungen hinzugefügt werden, die er an Ihren Ursprung sendet, aktualisieren Sie die Ursprungskonfiguration mit einer der folgenden Methoden:

- CloudFront Konsole — Wenn Sie eine Distribution erstellen oder aktualisieren, geben Sie Header-Namen und -Werte in den Einstellungen „Benutzerdefinierte Header hinzufügen“ an. Weitere Informationen finden Sie unter Benutzerdefinierten Header hinzufügen.
- CloudFront API — Geben Sie für jeden Ursprung, dem Sie benutzerdefinierte Header hinzufügen möchten, die Header-Namen und -Werte im `CustomHeaders` Feld darin an. `Origin` Weitere Informationen finden Sie unter CreateDistribution oder UpdateDistribution in der CloudFront Amazon-API-Referenz.

Wenn die von Ihnen angegebenen Header-Namen und -Werte nicht bereits in der Viewer-Anfrage enthalten sind, CloudFront fügen Sie sie der ursprünglichen Anfrage hinzu. Wenn ein Header vorhanden ist, wird der Header-Wert CloudFront überschrieben, bevor die Anfrage an den Ursprung weitergeleitet wird.

Informationen zu den Kontingenten für benutzerdefinierte Ursprungsheader finden Sie unter Kontingente für Header.

Benutzerdefinierte Header, die nicht zu CloudFront Quellanfragen hinzugefügt werden können

Sie können nicht so konfigurieren CloudFront , dass Anfragen, die an Ihren Ursprung gesendet werden, einer der folgenden Header hinzugefügt wird:

- `Cache-Control`
- `Connection`
- `Content-Length`
- `Cookie`
- `Host`
- `If-Match`
- `If-Modified-Since`

- `If-None-Match`
- `If-Range`
- `If-Unmodified-Since`
- `Max-Forwards`
- `Pragma`
- `Proxy-Authenticate`
- `Proxy-Authorization`
- `Proxy-Connection`
- `Range`
- `Request-Range`
- `TE`
- `Trailer`
- `Transfer-Encoding`
- `Upgrade`
- `Via`
- Header, die mit `X-Amz-` beginnen
- Header, die mit `X-Edge-` beginnen
- `X-Real-Ip`

Konfigurieren Sie CloudFront , um den Authorization-Header weiterzuleiten

Wenn eine Viewer-Anfrage an Ihren Absender CloudFront weitergeleitet CloudFront wird, werden standardmäßig einige Viewer-Header entfernt, einschließlich des `Authorization` Headers. Um sicherzustellen, dass Ihr Ursprung immer den `Authorization`-Header in Ursprungsanforderungen erhält, haben Sie folgende Möglichkeiten:

- Fügen Sie den `Authorization`-Header mithilfe einer Cache-Richtlinie zum Cache-Schlüssel hinzu. Alle Header im Cache-Schlüssel werden automatisch in Ursprungsanforderungen eingeschlossen. Weitere Informationen finden Sie unter Steuern des Cache-Schlüssels mit einer Richtlinie.
- Füge den `Authorization` Header einzeln in einer Quell-Anforderungsrichtlinie hinzu. Weitere Informationen finden Sie unter Steuern von Ursprungsanforderungen anhand einer Richtlinie.

- Verwenden Sie eine Quellanforderungsrichtlinie, die alle Viewer-Header an den Ursprung weiterleitet. CloudFront bietet eine verwaltete Quellanforderungsrichtlinie für diesen Anwendungsfall mit dem Namen. Managed-AllViewer Weitere Informationen finden Sie unter Verwenden verwalteter Ursprungsanforderungsrichtlinien.

> **Important**
>
> Wenn Sie den `Authorization` Header an Ihren Ursprung weiterleiten, ohne ihn in den Cache-Schlüssel aufzunehmen, stellen Sie sicher, dass Ihr Ursprung bei der Zugriffskontrolle für zwischengespeicherte Inhalte nicht auf den `Authorization` Header angewiesen ist. Wenn der `Authorization` Header nicht Teil des Cache-Schlüssels ist, CloudFront kann er sowohl autorisierten als auch nicht autorisierten Zuschauern dieselbe zwischengespeicherte Antwort senden. Nehmen Sie den `Authorization` Header entweder mithilfe einer Cache-Richtlinie in den Cache-Schlüssel auf oder deaktivieren Sie das Caching vollständig für Ursprünge, für die eine Autorisierungsverarbeitung auf der Originalseite erforderlich ist.

Wie CloudFront verarbeitet Teilanfragen für ein Objekt (Bereichs-GETs)

Bei großen Objekten kann der Viewer (Webbrowser oder anderer Client) mehrere `GET`-Anforderungen stellen und verwendet den `Range`-Anforderungs-Header, um das Objekt in kleineren Teilen herunterzuladen. Diese Anfragen für Bereiche von Bytes, manchmal auch als `Range GET`-Anfragen bezeichnet, steigern die Effizienz von Teil-Downloads und die Wiederherstellung nach teilweise fehlgeschlagenen Übertragungen.

Wenn es eine `Range GET` Anfrage CloudFront erhält, überprüft es den Cache an dem Edge-Standort, an dem die Anfrage empfangen wurde. Wenn der Cache an dieser Edge-Position bereits das gesamte Objekt oder den angeforderten Teil des Objekts enthält, wird CloudFront sofort der angeforderte Bereich aus dem Cache bereitgestellt.

Wenn der Cache den angeforderten Bereich nicht enthält, CloudFront leitet die Anfrage an den Ursprung weiter. (Zur Leistungsoptimierung wird CloudFront möglicherweise ein größerer Bereich angefordert, als der Client in der angefordert hat`Range GET`.) Der nächste Schritt hängt davon ab, ob der Ursprung `Range GET`-Anfragen unterstützt:

- Wenn der Ursprung **`Range GET`** Anfragen unterstützt — Er gibt den angeforderten Bereich zurück. CloudFront stellt den angeforderten Bereich bereit und speichert ihn auch für zukünftige Anfragen im Cache. (Amazon S3 unterstützt `Range GET`-Anforderungen, ebenso wie viele HTTP-Server.)
- Wenn der Ursprung keine **`Range GET`** Anfragen unterstützt, wird das gesamte Objekt zurückgegeben. CloudFront bedient die aktuelle Anfrage, indem das gesamte Objekt gesendet und gleichzeitig für zukünftige Anfragen zwischengespeichert wird. Nachdem CloudFront das gesamte Objekt in einem Edge-Cache zwischengespeichert wurde, reagiert es auf neue `Range GET` Anfragen, indem es den angeforderten Bereich bereitstellt.

In beiden Fällen CloudFront beginnt die Bereitstellung des angeforderten Bereichs oder Objekts an den Endbenutzer, sobald das erste Byte vom Ursprung eintrifft.

> **Note**
>
> Wenn der Betrachter eine `Range GET` Anfrage stellt und der Ursprung CloudFront zurückkehrt`Transfer-Encoding: chunked`, wird das gesamte Objekt an den Betrachter zurückgegeben, und nicht der angeforderte Bereich.

CloudFront folgt im Allgemeinen der RFC-Spezifikation für den `Range` Header. Wenn Ihre `Range`-Header die folgenden Anforderungen jedoch nicht erfüllen, gibt CloudFront HTTP-Statuscode `200` mit dem vollständigen Objekt anstelle von Statuscode `206` mit den angefragten Bereichen zurück:

- Die Bereiche müssen in aufsteigender Reihenfolge aufgeführt sein. Beispielweise ist `100-200,300-400` gültig, `300-400,100-200` hingegen nicht.
- Die Bereiche dürfen sich nicht überschneiden. Beispielsweise ist `100-200,150-250` nicht gültig.
- Alle Bereichsspezifikationen müssen gültig sein. Beispielsweise können Sie keinen negativen Wert als Teil eines Bereichs angeben.

Weitere Informationen zum `Range`-Anforderungs-Header finden Sie im Abschnitt zu Bereichsanforderungen in RFC 7233 oder im Abschnitt zum Bereich in den MDN-Webdokumenten.

Verwenden von Bereichsanforderungen zum Zwischenspeichern großer Objekte

Wenn das Caching aktiviert ist, wird CloudFront kein Objekt abgerufen oder zwischengespeichert, das größer als 50 GB ist. Wenn ein Ursprung angibt, dass das Objekt größer als diese Größe ist (im `Content-Length` Antwort-Header), CloudFront wird die Verbindung zum Ursprung geschlossen und eine Fehlermeldung wird an den Viewer zurückgegeben. (Bei deaktiviertem Caching CloudFront kann ein Objekt, das größer als diese Größe ist, vom Ursprung abgerufen und an den Viewer weitergegeben werden. Das Objekt CloudFront wird jedoch nicht zwischengespeichert.)

Bei Bereichsanforderungen können CloudFront Sie jedoch ein Objekt zwischenspeichern, das größer als die maximal zwischenspeicherbare Dateigröße ist.

Example Beispiel

1. Gehen wir von einem Ursprung mit einem Objekt mit 100 GB aus. Wenn das Caching aktiviert ist, CloudFront wird ein Objekt dieser Größe nicht abgerufen oder zwischengespeichert. Der Viewer kann jedoch mehrere Bereichsanforderungen senden, um dieses Objekt in Teilen abzurufen, wobei die Teile jeweils kleiner als 50 GB sind.
2. Der Viewer kann das Objekt in 20-GB-Teilen anfordern, indem er eine Anforderung mit dem Header `Range: bytes=0-21474836480` sendet, um den ersten Teil abzurufen, eine weitere Anforderung mit dem Header `Range: bytes=21474836481-42949672960`, um den nächsten Teil abzurufen und so weiter.
3. Nachdem der Viewer alle Teile empfangen hat, kann er sie kombinieren, um das ursprüngliche Objekt mit 100 GB zu erstellen.
4. In diesem Fall CloudFront speichert jedes der 20-GB-Teile des Objekts im Cache und kann auf nachfolgende Anfragen für denselben Teil aus dem Cache antworten.

Bei einer Bereichsanforderung für ein komprimiertes Objekt basiert der angeforderte Bytebereich auf der komprimierten Größe und nicht auf der Originalgröße des Objekts. Weitere Informationen zur Komprimierung von Dateien finden Sie unter Bereitstellen von komprimierten Dateien.

Wie CloudFront verarbeitet HTTP 3xx-Statuscodes von Ihrem Ursprung

Wenn Sie CloudFront ein Objekt von Ihrem Amazon S3-Bucket oder Ihrem benutzerdefinierten Ursprungsserver anfordern, gibt Ihr Ursprungsserver manchmal einen HTTP 3xx-Statuscode zurück. Dies weist in der Regel auf einen der folgenden Umstände hin:

- Die URL des Objekts hat sich geändert (z. B. Statuscodes 301, 302, 307 oder 308).
- Das Objekt hat sich seit der letzten CloudFront Anforderung nicht geändert (Statuscode 304)

CloudFront speichert 3xx-Antworten entsprechend den Einstellungen in Ihrer CloudFront Distribution und den Headern in der Antwort im Cache. CloudFront speichert 307- und 308-Antworten nur dann im Cache, wenn Sie den `Cache-Control` Header in die Antworten vom Ursprung aufnehmen. Weitere Informationen finden Sie unter Verwalten der Dauer, die Inhalte im Cache bleiben (Ablauf).

Wenn dein Absender einen Weiterleitungsstatuscode zurückgibt (z. B. 301 oder 307), folgt er der Weiterleitung CloudFront nicht. CloudFront leitet die 301- oder 307-Antwort an den Zuschauer weiter, der der Weiterleitung folgen kann, indem er eine neue Anfrage sendet.

Wie CloudFront verarbeitet die HTTP 4xx- und 5xx-Statuscodes von Ihrem Ursprung

Wenn Sie CloudFront ein Objekt von Ihrem Amazon S3-Bucket oder Ihrem benutzerdefinierten Ursprungsserver anfordern, gibt Ihr Ursprungsserver manchmal einen HTTP 4xx- oder 5xx-Statuscode zurück, der darauf hinweist, dass ein Fehler aufgetreten ist. CloudFront Das Verhalten hängt ab von:

- ob Sie benutzerdefinierte Fehlerseiten konfiguriert haben
- Ob Sie konfiguriert haben, wie lange Sie Fehlerantworten von Ihrem Ursprung zwischenspeichern CloudFront möchten (Mindest-TTL für das Zwischenspeichern von Fehlern)
- dem Statuscode
- Bei 5xx-Statuscodes, ob sich das angeforderte Objekt derzeit im Edge-Cache befindet CloudFront
- bei einigen 4xx-Statuscodes, ob der Ursprung einen `Cache-Control max-age-` oder `Cache-Control s-maxage`-Header zurückgibt

CloudFront speichert Antworten auf `GET` und `HEAD` Anfragen immer im Cache. Sie können auch so konfigurieren CloudFront , dass Antworten auf `OPTIONS` Anfragen zwischengespeichert werden. CloudFront speichert Antworten auf Anfragen, die die anderen Methoden verwenden, nicht im Cache.

Wenn der Absender nicht antwortet, kommt es bei der CloudFront Anfrage an den Ursprung zu einem Timeout, was als HTTP 5xx-Fehler vom Ursprung betrachtet wird, obwohl der Absender nicht mit diesem Fehler geantwortet hat. In diesem Szenario werden CloudFront weiterhin zwischengespeicherte Inhalte bereitgestellt. Weitere Informationen finden Sie unter Ursprung nicht verfügbar.

Wenn Sie die Protokollierung aktiviert haben, werden die Ergebnisse unabhängig vom HTTP-Statuscode in die Protokolle CloudFront geschrieben.

Weitere Informationen zu Funktionen und Optionen, die sich auf die von Ihnen zurückgegebene Fehlermeldung beziehen CloudFront, finden Sie unter den folgenden Links:

- Informationen zu Einstellungen für benutzerdefinierte Fehlerseiten in der CloudFront Konsole finden Sie unterBenutzerdefinierte Fehlerseiten und Zwischenspeicherung von Fehlern.
- Hinweise zum Fehler beim Zwischenspeichern der minimalen TTL in der CloudFront Konsole finden Sie unter. Mindest-TTL für die Zwischenspeicherung von Fehlern (Sekunden)
- Eine Liste der HTTP-Statuscodes, die CloudFront zwischengespeichert werden, finden Sie unter. HTTP-Statuscodes 4xx und 5xx, die zwischengespeichert werden CloudFront

Themen

- Wie CloudFront verarbeitet Fehler, wenn Sie benutzerdefinierte Fehlerseiten konfiguriert haben
- Wie CloudFront verarbeitet Fehler, wenn Sie keine benutzerdefinierten Fehlerseiten konfiguriert haben
- HTTP-Statuscodes 4xx und 5xx, die zwischengespeichert werden CloudFront

Wie CloudFront verarbeitet Fehler, wenn Sie benutzerdefinierte Fehlerseiten konfiguriert haben

Wenn Sie benutzerdefinierte Fehlerseiten konfiguriert haben, hängt CloudFront das Verhalten davon ab, ob sich das angeforderte Objekt im Edge-Cache befindet.

Das angeforderte Objekt ist nicht im Edge-Cache vorhanden

CloudFront versucht weiterhin, das angeforderte Objekt von Ihrem Ursprung abzurufen, wenn alle der folgenden Bedingungen zutreffen:

- Ein Viewer fordert ein Objekt an.
- Das Objekt ist nicht im Edge-Cache vorhanden.
- Ihr Ursprungsserver gibt einen HTTP-Statuscode 4xx oder 5xx zurück und eines der Folgenden ist wahr:
  - Ihr Ursprungsserver gibt einen HTTP-Statuscode 5xx anstelle eines Statuscodes 304 (nicht geändert) oder eine aktualisierte Version des Objekts zurück.
  - Ihr Ursprungs-Server gibt einen HTTP-Statuscode 4xx zurück, der nicht durch einen Cache-Control-Header eingeschränkt und in der folgenden Statuscodeliste enthalten is: HTTP-Statuscodes 4xx und 5xx, die zwischengespeichert werden CloudFront.
  - Ihr Ursprungsserver gibt einen HTTP-Statuscode 4xx mit einem `Cache-Control max-age`- oder `Cache-Control s-maxage`-Header zurück und der Statuscode ist in der folgenden Statuscodeliste enthalten: HTTP 4xx-Statuscodes, die basierend auf CloudFront Headern zwischengespeichert werden `Cache-Control`.

CloudFront macht Folgendes:

1. CloudFront Überprüft im CloudFront Edge-Cache, der die Viewer-Anfrage erhalten hat, Ihre Verteilungskonfiguration und ruft den Pfad der benutzerdefinierten Fehlerseite ab, die dem Statuscode entspricht, den Ihr Ursprung zurückgegeben hat.
2. CloudFront findet das erste Cache-Verhalten in Ihrer Distribution, das ein Pfadmuster hat, das dem Pfad der benutzerdefinierten Fehlerseite entspricht.
3. Die CloudFront Edge-Position sendet eine Anforderung für die benutzerdefinierte Fehlerseite an den Ursprung, der im Cache-Verhalten angegeben ist.
4. Der Ursprungsserver gibt die benutzerdefinierte Fehlerseite an den Edge-Standort zurück.
5. CloudFront gibt die benutzerdefinierte Fehlerseite an den Viewer zurück, der die Anforderung gestellt hat, und speichert außerdem die benutzerdefinierte Fehlerseite für maximal Folgendes im Cache:
   - Der Zeitraum, der durch die Mindest-TTL für die Zwischenspeicherung von Fehlern angegeben ist (standardmäßig zehn Sekunden)

- Der Zeitraum, der durch einen `Cache-Control max-age-` oder einen `Cache-Control s-maxage`-Header angegeben ist, der vom Ursprungsserver zurückgegeben wird, wenn die erste Anfrage den Fehler generiert hat

6. CloudFront Versuchen Sie nach Ablauf der (in Schritt 5 festgelegten) Cache-Zeit erneut, das angeforderte Objekt abzurufen, indem Sie eine weitere Anfrage an Ihren Ursprung weiterleiten. CloudFront wiederholt den Vorgang in Intervallen, die durch die Mindest-TTL für den Fehler beim Zwischenspeichern festgelegt sind.

> **Note**
>
> Wenn Sie für dieselbe benutzerdefinierte Fehlerseite auch ein Cacheverhalten konfiguriert haben, wird stattdessen das Cache-Verhalten TTL CloudFront verwendet. Gehen Sie in diesem CloudFront Fall für die Schritte 5 und 6 wie folgt vor:
>
> - After CloudFront gibt die benutzerdefinierte Fehlerseite an den Viewer zurück, der die Anfrage gestellt hat, und CloudFront überprüft das Cacheverhalten TTL (Sie legen beispielsweise die Standard-TTL auf 5 Sekunden fest). CloudFront speichert dann die benutzerdefinierte Fehlerseite bis zu diesem Höchstwert im Cache.
> - CloudFront Ruft nach Ablauf von 5 Sekunden die benutzerdefinierte Fehlerseite erneut vom Ursprung ab. CloudFront versucht es weiterhin in Intervallen, die durch das Cache-Verhalten (TTL) festgelegt sind.
>
> Weitere Informationen finden Sie unter TTL-Einstellungen für das Cacheverhalten.

Das angeforderte Objekt ist im Edge-Cache vorhanden

CloudFront bedient weiterhin das Objekt, das sich derzeit im Edge-Cache befindet, wenn alle der folgenden Bedingungen zutreffen:

- Ein Viewer fordert ein Objekt an.
- Das Objekt ist im Edge-Cache vorhanden, aber es ist abgelaufen.
- Ihr Ursprungsserver gibt einen HTTP-Statuscode 5xx anstelle eines Statuscodes 304 (nicht geändert) oder eine aktualisierte Version des Objekts zurück.

CloudFront macht Folgendes:

1. Wenn Ihr Ursprung einen 5xx-Statuscode zurückgibt, CloudFront wird das Objekt bereitgestellt, obwohl es abgelaufen ist. Während der Dauer des Fehlers beim Caching reagiert TTL CloudFront weiterhin auf Viewer-Anfragen, indem es das Objekt aus dem Edge-Cache bereitstellt.

   Wenn Ihr Ursprungsserver einen 4xx-Statuscode zurückgibt, sendet CloudFront den Statuscode anstelle des angeforderten Objekts an den Betrachter.
2. CloudFront Versuchen Sie erneut, das angeforderte Objekt abzurufen, indem Sie eine weitere Anfrage an Ihren Ursprung weiterleiten, nachdem der Fehler beim Zwischenspeichern der TTL abgelaufen ist. Beachten Sie, dass das Objekt, wenn es nicht häufig angefordert wird, CloudFront möglicherweise aus dem Edge-Cache entfernt wird, während Ihr Originalserver immer noch 5xx-Antworten zurückgibt. Hinweise dazu, wie lange Objekte in CloudFront Edge-Caches bleiben, finden Sie unter. Verwalten der Dauer, die Inhalte im Cache bleiben (Ablauf)

Wie CloudFront verarbeitet Fehler, wenn Sie keine benutzerdefinierten Fehlerseiten konfiguriert haben

Wenn Sie keine benutzerdefinierten Fehlerseiten konfiguriert haben, hängt CloudFront das Verhalten davon ab, ob sich das angeforderte Objekt im Edge-Cache befindet.

Themen

- Das angeforderte Objekt ist nicht im Edge-Cache vorhanden
- Das angeforderte Objekt ist im Edge-Cache vorhanden

Das angeforderte Objekt ist nicht im Edge-Cache vorhanden

CloudFront versucht weiterhin, das angeforderte Objekt von Ihrem Ursprung abzurufen, wenn alle der folgenden Bedingungen zutreffen:

- Ein Viewer fordert ein Objekt an.
- Das Objekt ist nicht im Edge-Cache vorhanden.
- Ihr Ursprungsserver gibt einen HTTP-Statuscode 4xx oder 5xx zurück und eines der Folgenden ist wahr:
  - Ihr Ursprungsserver gibt einen HTTP-Statuscode 5xx anstelle eines Statuscodes 304 (nicht geändert) oder eine aktualisierte Version des Objekts zurück.

- Ihr Ursprungs-Server gibt einen HTTP-Statuscode 4xx zurück, der nicht durch einen Cache-Control-Header eingeschränkt und in der folgenden Statuscodeliste enthalten is: HTTP-Statuscodes 4xx und 5xx, die zwischengespeichert werden CloudFront
- Ihr Ursprungsserver gibt einen HTTP-Statuscode 4xx mit einem `Cache-Control max-age`- oder `Cache-Control s-maxage`-Header zurück und der Statuscode ist in der folgenden Statuscodeliste enthalten: HTTP 4xx-Statuscodes, die basierend auf CloudFront Headern zwischengespeichert werden `Cache-Control`.

CloudFront macht Folgendes:

1. CloudFront gibt den 4xx- oder 5xx-Statuscode an den Viewer zurück und speichert auch den Statuscode im Edge-Cache, der die Anforderung für maximal Folgendes erhalten hat:
   - Der Zeitraum, der durch die Mindest-TTL für die Zwischenspeicherung von Fehlern angegeben ist (standardmäßig zehn Sekunden)
   - Der Zeitraum, der durch einen `Cache-Control max-age`- oder einen `Cache-Control s-maxage`-Header angegeben ist, der vom Ursprungsserver zurückgegeben wird, wenn die erste Anfrage den Fehler generiert hat
2. Für die Dauer der Zwischenspeicherung (in Schritt 1 bestimmt) reagiert CloudFront auf nachfolgende Viewer-Anfragen für dasselbe Objekte mit den zwischengespeicherten Statuscodes 4xx und 5xx.
3. CloudFront Versuchen Sie nach Ablauf der (in Schritt 1 festgelegten) Cache-Zeit erneut, das angeforderte Objekt abzurufen, indem Sie eine weitere Anfrage an Ihren Ursprung weiterleiten. CloudFront wiederholt den Vorgang in Intervallen, die durch die Mindest-TTL für den Fehler beim Zwischenspeichern festgelegt sind.

Das angeforderte Objekt ist im Edge-Cache vorhanden

CloudFront bedient weiterhin das Objekt, das sich derzeit im Edge-Cache befindet, wenn alle der folgenden Bedingungen zutreffen:

- Ein Viewer fordert ein Objekt an.
- Das Objekt ist im Edge-Cache vorhanden, aber es ist abgelaufen. Das bedeutet, dass das Objekt veraltet ist.
- Ihr Ursprungsserver gibt einen HTTP-Statuscode 5xx anstelle eines Statuscodes 304 (nicht geändert) oder eine aktualisierte Version des Objekts zurück.

CloudFront macht Folgendes:

1. Wenn Ihr Ursprung einen 5xx-Fehlercode zurückgibt, CloudFront wird das Objekt bereitgestellt, obwohl es abgelaufen ist. Während der Dauer des Fehlers beim Zwischenspeichern der TTL-Mindestwert (standardmäßig 10 Sekunden) reagiert CloudFront weiterhin auf Viewer-Anfragen, indem er das Objekt aus dem Edge-Cache bereitstellt.

   Wenn Ihr Ursprungsserver einen 4xx-Statuscode zurückgibt, sendet CloudFront den Statuscode anstelle des angeforderten Objekts an den Betrachter.

2. CloudFront Versuchen Sie erneut, das angeforderte Objekt abzurufen, indem Sie eine weitere Anfrage an Ihren Ursprung weiterleiten, nachdem die Mindest-TTL für den Fehler beim Zwischenspeichern abgelaufen ist. Wenn das Objekt nicht häufig angefordert wird, löschen Sie es CloudFront möglicherweise aus dem Edge-Cache, während Ihr Original-Server immer noch 5xx-Antworten zurückgibt. Weitere Informationen finden Sie unter Verwalten der Dauer, die Inhalte im Cache bleiben (Ablauf).

> **Tip**
>
> - Wenn Sie die Direktive `stale-if-error` oder `Stale-While-Revalidate` konfigurieren, können Sie angeben, wie lange die veralteten Objekte im Edge-Cache verfügbar sein sollen. Auf diese Weise können Sie weiterhin Inhalte für Ihre Zuschauer bereitstellen, auch wenn Ihr Ursprungsserver nicht verfügbar ist. Weitere Informationen finden Sie unter Bereitstellung veralteter (abgelaufener) Inhalte.
> - CloudFront stellt nur Objekte bereit, die bis zum angegebenen ??? TTL-Maximalwert veraltet sind. Nach Ablauf dieser Dauer ist das Objekt nicht mehr im Edge-Cache verfügbar.

HTTP-Statuscodes 4xx und 5xx, die zwischengespeichert werden CloudFront

CloudFront speichert die von Ihrem Absender zurückgegebenen HTTP 4xx- und 5xx-Statuscodes im Cache, abhängig vom spezifischen Statuscode, der zurückgegeben wird, und davon, ob Ihr Ursprung bestimmte Header in der Antwort zurückgibt.

CloudFront speichert die folgenden HTTP 4xx- und 5xx-Statuscodes, die von Ihrem Absender zurückgegeben wurden, im Cache. Wenn Sie eine benutzerdefinierte Fehlerseite für einen HTTP-Statuscode konfiguriert haben, wird die benutzerdefinierte Fehlerseite CloudFront zwischengespeichert.

> **Note**
>
> Wenn Sie die Richtlinie für CachingDisabled verwalteten Cache verwenden, CloudFront werden diese Statuscodes oder benutzerdefinierten Fehlerseiten nicht zwischengespeichert.

HTTP-Statuscode	Description
404	Not Found
414	Request-URI Zu groß
500	Internal Server Error
501	Nicht implementiert
502	Bad Gateway
503	Service nicht verfügbar
504	Gateway Time-out

HTTP 4xx-Statuscodes, die basierend auf CloudFront Headern zwischengespeichert werden **`Cache-Control`**

CloudFront speichert nur die folgenden HTTP 4xx-Statuscodes, die von Ihrem Ursprung zurückgegeben werden, im Cache, wenn Ihr Ursprung einen OR-Header zurückgibt. `Cache-`

`Control max-age` `Cache-Control s-maxage` Wenn Sie eine benutzerdefinierte Fehlerseite für einen dieser HTTP-Statuscodes konfiguriert haben — und Ihr Ursprung einen der Cache-Control-Header zurückgibt —, wird die benutzerdefinierte Fehlerseite CloudFront zwischengespeichert.

HTTP-Statuscode	Description
400	Inkorrekte Anfrage
403	Forbidden
405	Method Not Allowed
412[1]	Vorbedingung fehlgeschlagen
415[1]	Unsupported Media Type (Nicht unterstützter Medientyp)

[1] unterstützt CloudFront nicht die Erstellung benutzerdefinierter Fehlerseiten für diese HTTP-Statuscodes.

Erstellen von benutzerdefinierten Fehlerantworten

Wenn ein Objekt, über das Sie bereitstellen, aus irgendeinem Grund nicht verfügbar CloudFront ist, gibt Ihr Webserver in der Regel einen entsprechenden HTTP-Statuscode zurück, CloudFront um darauf hinzuweisen. Wenn ein Betrachter beispielsweise eine ungültige URL anfordert, gibt Ihr Webserver einen HTTP-Statuscode 404 (Not Found) CloudFront zurück und gibt diesen Statuscode dann an den Betrachter zurück. CloudFront Anstatt diese Standardfehlerantwort zu verwenden, können Sie eine benutzerdefinierte Antwort erstellen, die zum Viewer CloudFront zurückkehrt.

Wenn Sie die Rückgabe einer benutzerdefinierten Fehlerseite für einen HTTP-Statuscode konfigurieren CloudFront , die benutzerdefinierte Fehlerseite jedoch nicht verfügbar ist, wird der Statuscode, den Sie von der Quelle CloudFront erhalten haben und die benutzerdefinierten Fehlerseiten enthalten, an den Betrachter CloudFront zurückgegeben. Nehmen wir zum Beispiel an, Ihr benutzerdefinierter Ursprung gibt einen Statuscode 500 zurück und Sie haben konfiguriert

CloudFront , dass eine benutzerdefinierte Fehlerseite für einen 500-Statuscode aus einem Amazon S3-Bucket abgerufen wird. Jemand hat jedoch versehentlich die benutzerdefinierte Fehlerseite aus Ihrem Amazon S3 S3-Bucket gelöscht. CloudFront gibt einen HTTP-404-Statuscode (Nicht gefunden) an den Betrachter zurück, der das Objekt angefordert hat.

Wenn eine benutzerdefinierte Fehlerseite an einen Betrachter CloudFront zurückgegeben wird, zahlen Sie die CloudFront Standardgebühren für die benutzerdefinierte Fehlerseite, nicht die Gebühren für das angeforderte Objekt. Weitere Informationen zu CloudFront Gebühren finden Sie unter CloudFrontAmazon-Preise.

Themen

- Konfigurieren des Fehlerantwortverhaltens
- Erstellen einer benutzerdefinierten Fehlerseite für bestimmte HTTP-Statuscodes
- Speichern von Objekten und benutzerdefinierten Fehlerseiten an anderen Speicherorten
- Ändern Sie die Antwortcodes, die zurückgegeben wurden von CloudFront
- Steuern Sie, wie lange Fehler CloudFront zwischengespeichert werden

Konfigurieren des Fehlerantwortverhaltens

Sie haben mehrere Optionen, um zu verwalten, wie CloudFront auf einen Fehler reagiert wird. Um benutzerdefinierte Fehlerantworten zu konfigurieren, können Sie die CloudFront Konsole, die CloudFront API oder verwenden CloudFormation. Unabhängig davon, wie Sie die Konfiguration aktualisieren, sollten Sie die folgenden Tipps und Empfehlungen beachten:

- Speichern Sie Ihre benutzerdefinierten Fehlerseiten an einem Ort, auf den zugegriffen werden kann CloudFront. Wir empfehlen Ihnen, sie in einem Amazon-S3-Bucket zu speichern und sie nicht am selben Ort wie den Rest der Inhalte Ihrer Website oder Anwendung zu speichern. Wenn Sie die benutzerdefinierten Fehlerseiten auf demselben Ursprung wie Ihre Website oder Anwendung speichern und der Ursprung beginnt, 5xx-Fehler zurückzugeben, CloudFront können Sie die benutzerdefinierten Fehlerseiten nicht abrufen, da der Ursprungsserver nicht verfügbar ist. Weitere Informationen finden Sie unter Speichern von Objekten und benutzerdefinierten Fehlerseiten an anderen Speicherorten.
- Stellen Sie sicher, dass dieser Benutzer berechtigt CloudFront ist, Ihre benutzerdefinierten Fehlerseiten abzurufen. Wenn die benutzerdefinierten Fehlerseiten in Amazon S3 gespeichert sind, müssen die Seiten öffentlich zugänglich sein oder Sie müssen eine CloudFront Origin

Access Control (OAC) konfigurieren. Wenn die benutzerdefinierten Fehlerseiten in einem benutzerdefinierten Ursprung gespeichert sind, müssen die Seiten öffentlich zugänglich sein.

- (Optional) Konfigurieren Sie Ihren Ursprung so, dass ein `Cache-Control`- oder `Expires`-Header zusammen mit den benutzerdefinierten Fehlerseiten hinzugefügt wird, wenn Sie möchten. Sie können auch die Einstellung Minimum TTL für Error Caching verwenden, um zu steuern, wie lange die benutzerdefinierten Fehlerseiten CloudFront zwischengespeichert werden. Weitere Informationen finden Sie unter Steuern Sie, wie lange Fehler CloudFront zwischengespeichert werden.

Konfigurieren benutzerdefinierter Fehlerantworten

Um benutzerdefinierte Fehlerantworten in der CloudFront Konsole zu konfigurieren, benötigen Sie eine Distribution. CloudFront In der Konsole stehen die Konfigurationseinstellungen für benutzerdefinierte Fehlerantworten nur für vorhandene Verteilungen zur Verfügung. Informationen zum Erstellen einer Verteilung finden Sie unter Beginnen Sie mit einer CloudFront Standarddistribution.

Console

Konfigurieren benutzerdefinierter Fehlerantworten (Konsole)

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Seite Verteilungen in der CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home#distributions.
2. Wählen Sie in der Liste der Verteilungen die zu aktualisierende Verteilung aus.
3. Wählen Sie die Registerkarte Fehlerseiten und wählen Sie dann Benutzerdefinierte Fehlerantwort erstellen.
4. Geben Sie die entsprechenden Werte ein. Weitere Informationen finden Sie unter Benutzerdefinierte Fehlerseiten und Zwischenspeicherung von Fehlern.
5. Nachdem Sie die gewünschten Werte eingegeben haben, wählen Sie Erstellen.

CloudFront API or CloudFormation

Um benutzerdefinierte Fehlerantworten mit der CloudFront API oder zu konfigurieren CloudFormation, verwenden Sie den `CustomErrorResponse` Typ in einer Distribution. Weitere Informationen finden Sie hier:

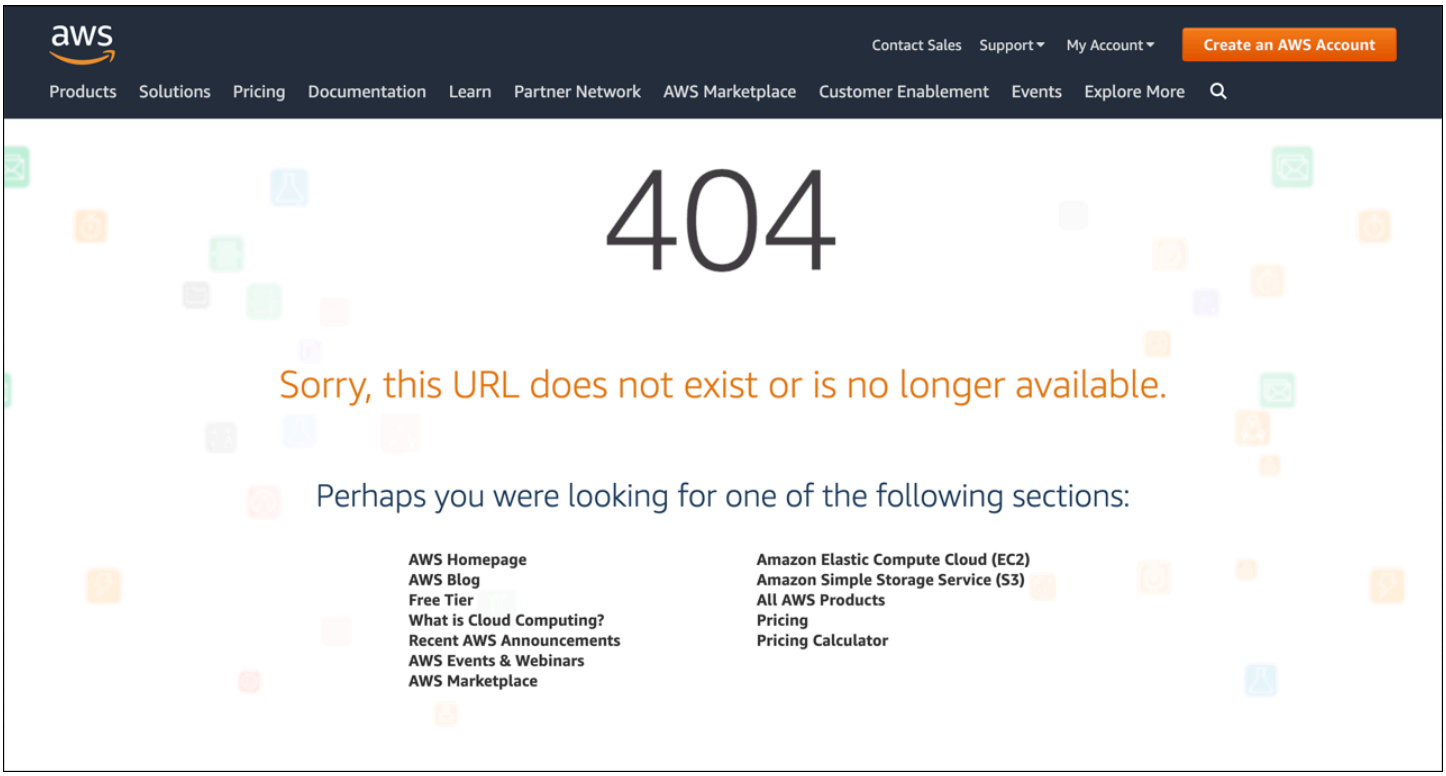
- AWS::CloudFront: :Distribution CustomErrorResponse im AWS CloudFormation Benutzerhandbuch
- CustomErrorResponsein der Amazon CloudFront API-Referenz

Erstellen einer benutzerdefinierten Fehlerseite für bestimmte HTTP-Statuscodes

Wenn Sie statt der Standardmeldung lieber eine benutzerdefinierte Fehlermeldung anzeigen möchten, z. B. eine Seite, die dieselbe Formatierung wie der Rest Ihrer Website verwendet, können Sie ein Objekt (z. B. eine HTML-Datei), das Ihre benutzerdefinierte Fehlermeldung enthält, an den Viewer CloudFront zurückgeben lassen.

Um die Datei, die Sie zurückgeben möchten, und die Fehler, für die die Datei zurückgegeben werden soll, anzugeben, aktualisieren Sie Ihre CloudFront Distribution, sodass diese Werte angegeben werden. Weitere Informationen finden Sie unter Konfigurieren des Fehlerantwortverhaltens.

Zum Beispiel ist das Folgende eine benutzerdefinierte Fehlerseite:



Sie können ein anderes Objekt für jeden unterstützten HTTP-Statuscode festlegen oder Sie können dasselbe Objekt für alle unterstützten Statuscodes verwenden. Es ist möglich, benutzerdefinierte Fehlerseiten für einige Statuscodes festzulegen und für andere nicht.

Die Objekte, über die Sie bereitstellen, CloudFront können aus verschiedenen Gründen nicht verfügbar sein. Diese gliedern sich in zwei große Kategorien:

- Client-Fehler weisen auf ein Problem mit der Anfrage hin. Beispielsweise ist ein Objekt mit dem angegebenen Namen nicht verfügbar oder der Benutzer verfügt nicht über die erforderlichen Berechtigungen, um ein Objekt in Ihrem Amazon S3-Bucket abzurufen. Wenn ein Client-Fehler auftritt, gibt der Ursprung einen HTTP-Statuscode im Bereich 4xx bis CloudFront zurück.
- Server-Fehler weisen auf ein Problem mit dem Ursprungs-Server hin. Beispielsweise ist der HTTP-Server ausgelastet oder nicht verfügbar. Wenn ein Serverfehler auftritt, gibt Ihr Ursprungsserver entweder einen HTTP-Statuscode im Bereich 5xx zurück oder CloudFront er erhält für einen bestimmten Zeitraum keine Antwort von Ihrem Ursprungsserver und nimmt den Statuscode 504 an (Gateway Timeout). CloudFront

Zu den HTTP-Statuscodes, für die eine benutzerdefinierte Fehlerseite zurückgegeben werden CloudFront kann, gehören die folgenden:

- 400, 403, 404, 405, 414, 416
- 500, 501, 502, 503, 504

Hinweise

- Wenn CloudFront erkannt wird, dass die Anfrage möglicherweise unsicher ist, wird anstelle einer benutzerdefinierten Fehlerseite ein 400-Fehler (Bad Request) CloudFront zurückgegeben.
- Sie können eine benutzerdefinierte Fehlerseite für den HTTP-Statuscode 416 (Requested Range Not Satisfiable) erstellen und Sie können den HTTP-Statuscode ändern, der den Zuschauern CloudFront zurückgegeben wird, wenn Ihr Absender den Statuscode 416 an zurückgibt. CloudFront Weitere Informationen finden Sie unter Ändern Sie die Antwortcodes, die zurückgegeben wurden von CloudFront. Status-Code 416-Antworten werden jedoch CloudFront nicht zwischengespeichert, sodass dieser auch dann nicht verwendet wird, wenn Sie für den Statuscode 416 einen Wert für Error Caching Minimum TTL angeben. CloudFront
- In einigen Fällen CloudFront gibt es keine benutzerdefinierte Fehlerseite für den HTTP-Statuscode 503 zurück, selbst wenn Sie dies konfigurieren CloudFront . Wenn der CloudFront Fehlercode `Capacity Exceeded` oder ist`Limit Exceeded`, wird ein 503-

Statuscode an den Viewer CloudFront zurückgegeben, ohne Ihre benutzerdefinierte Fehlerseite zu verwenden.

- Wenn Sie eine benutzerdefinierte Fehlerseite erstellt haben, CloudFront wird `Connection: keep-alive` für die folgenden Antwortcodes `Connection: close` oder zurückgegeben:
  - CloudFront gibt `Connection: close` für Statuscodes zurück: 400, 405, 414, 416, 500, 501
  - CloudFront gibt `Connection: keep-alive` für Statuscodes zurück: 403, 404, 502, 503, 504

Eine ausführliche Erläuterung, wie CloudFront mit Fehlerantworten von Ihrer Quelle umgegangen wird, finden Sie unterWie CloudFront verarbeitet die HTTP 4xx- und 5xx-Statuscodes von Ihrem Ursprung.

Speichern von Objekten und benutzerdefinierten Fehlerseiten an anderen Speicherorten

Wenn Sie Ihre Objekte und Ihre benutzerdefinierten Fehlerseiten an verschiedenen Orten speichern möchten, muss Ihre Verteilung ein Cache-Verhalten mit den folgenden Eigenschaften enthalten:

- Der Wert von Path Pattern stimmt mit dem Pfad zu Ihren benutzerdefinierten Fehlermeldungen überein. Angenommen, Sie haben benutzerdefinierte Fehlerseiten für 4xx-Fehler in einem Amazon S3-Bucket in einem Verzeichnis mit dem Namen gespeicher `/4xx-errors`. Ihre Verteilung muss ein Cache-Verhalten beinhalten, für welches das Pfadmuster Anfragen für Ihre benutzerdefinierten Fehlerseiten an diesen Ort weiterleitet, z. B, `/4xx-errors/*`.
- Der Wert von Origin legt den Wert von Origin ID für den Ursprung fest, der Ihre benutzerdefinierten Fehlerseiten enthält.

Weitere Informationen finden Sie unter Einstellungen für das Cache-Verhalten.

Ändern Sie die Antwortcodes, die zurückgegeben wurden von CloudFront

Sie können so konfigurieren CloudFront , dass dem Betrachter ein anderer HTTP-Statuscode zurückgegeben wird als der, den CloudFront er vom Absender erhalten hat. Wenn Ihr Absender beispielsweise den Statuscode 500 an zurückgibtCloudFront, CloudFront möchten Sie

möglicherweise eine benutzerdefinierte Fehlerseite und den Statuscode 200 (OK) an den Betrachter zurückgeben. Es gibt eine Vielzahl von Gründen, warum Sie dem Betrachter möglicherweise einen Statuscode zurückgeben CloudFront möchten, der sich von dem unterscheidet, zu dem Ihr Ursprung zurückgekehrt istCloudFront:

- Einige Internetgeräte (beispielsweise einige Firewalls und Unternehmens-Proxys) fangen HTTP 4xx- und 5xx-Statuscodes ab und verhindern, dass die Antwort an den Betrachter zurückgegeben wird. Wenn Sie in diesem Szenario `200` ersetzen, wird die Antwort nicht abgefangen.
- Wenn Sie nicht zwischen verschiedenen Client- oder Serverfehlern unterscheiden möchten, können Sie `400` oder `500` als den Wert angeben, der für alle 4xx- oder 5xx-Statuscodes CloudFront zurückgegeben wird.
- Möglicherweise möchten Sie einen `200`-Statuscode (OK) und eine statische Website zurückgeben, sodass Ihre Kunden nicht wissen, dass die Website nicht verfügbar ist.

Wenn Sie CloudFront Standardprotokolle aktivieren und so konfigurierenCloudFront , dass der HTTP-Statuscode in der Antwort geändert wird, enthält der Wert der `sc-status` Spalte in den Protokollen den von Ihnen angegebenen Statuscode. Der Wert der `x-edge-result-type`-Spalte wird jedoch nicht beeinflusst. Sie enthält den Ergebnistyp der Antwort vom Ursprung. Nehmen wir zum Beispiel an, Sie konfigurieren CloudFront , dass der Statuscode von `200` an den Betrachter zurückgegeben wird, wenn der Ursprung `404` (Not Found) zu zurückkehrt CloudFront. Wenn der Ursprung auf eine Anfrage mit dem Statuscode `404` antwortet, ist der Wert in der Spalte `sc-status` im Protokoll `200`, der Wert in der Spalte `x-edge-result-type` jedoch `Error`.

Sie können so konfigurieren CloudFront , dass jeder der folgenden HTTP-Statuscodes zusammen mit einer benutzerdefinierten Fehlerseite zurückgegeben wird:

- 200
- 400, 403, 404, 405, 414, 416
- 500, 501, 502, 503, 504

Steuern Sie, wie lange Fehler CloudFront zwischengespeichert werden

CloudFront speichert Fehlerantworten für eine Standarddauer von 10 Sekunden im Cache. CloudFront sendet dann die nächste Anfrage für das Objekt an Ihre Quelle, um zu überprüfen, ob das Problem, das den Fehler verursacht hat, behoben wurde und das angeforderte Objekt verfügbar ist.

Sie können für jeden 4xx- und 5xx-Statuscode, der zwischengespeichert wird, die Dauer des Fehler-Cachings — die Mindest-TTL für das Fehler-Caching angeben. CloudFront (Weitere Informationen finden Sie unter HTTP-Statuscodes 4xx und 5xx, die zwischengespeichert werden CloudFront.) Wenn Sie eine Dauer angeben, beachten Sie Folgendes:

- Wenn Sie eine kurze Dauer für das Zwischenspeichern von Fehlern angeben, werden mehr Anfragen an Ihren Ursprung weitergeleitet, CloudFront als wenn Sie eine längere Dauer angeben. Bei 5xx-Fehlern verschlimmert dies möglicherweise das Problem, das ursprünglich dazu geführt hat, dass Ihr Ursprung einen Fehler zurückgibt.
- Wenn Ihr Absender einen Fehler für ein Objekt zurückgibt, CloudFront beantwortet er Anfragen für das Objekt entweder mit der Fehlerantwort oder mit Ihrer benutzerdefinierten Fehlerseite, bis die Dauer der Fehlerzwischenspeicherung abgelaufen ist. Wenn Sie eine lange Dauer für das Zwischenspeichern von Fehlern angeben, reagiert das CloudFront Objekt möglicherweise noch lange auf Anfragen mit einer Fehlerantwort oder Ihrer benutzerdefinierten Fehlerseite, nachdem das Objekt wieder verfügbar ist.

> **Note**
>
> Sie können eine benutzerdefinierte Fehlerseite für HTTP-Statuscode 416 (Requested Range Not Satisfiable) erstellen und Sie können den HTTP-Statuscode ändern, den CloudFront an Betrachter zurückgibt, wenn Ihr Ursprung einen Statuscode 416 an CloudFront zurückgibt. (Weitere Informationen finden Sie unter Ändern Sie die Antwortcodes, die zurückgegeben wurden von CloudFront.) Status-Code 416-Antworten werden jedoch CloudFront nicht zwischengespeichert. Selbst wenn Sie einen Wert für Error Caching Minimum TTL für den Statuscode 416 angeben, wird dieser Wert nicht verwendet. CloudFront

Wenn Sie steuern möchten, wie lange Fehler für einzelne Objekte CloudFront zwischengespeichert werden, können Sie Ihren Ursprungsserver so konfigurieren, dass er der Fehlerantwort für dieses Objekt den entsprechenden Header hinzufügt.

Wenn der Ursprung eine `Cache-Control: s-maxage` OR-Direktive `Cache-Control: max-age` oder einen `Expires` Header hinzufügt, werden Fehlerantworten CloudFront zwischengespeichert, je nachdem, welcher Wert im Header oder der Mindest-TTL für Error Caching gilt.

> **Note**
>
> Beachten Sie, dass die Werte für `Cache-Control: max-age` und `Cache-Control: s-maxage` nicht größer als der Wert für Maximum TTL sein können, der für das Cache-Verhalten festgelegt wurde, für das die Fehlerseite abgerufen wird.

Wenn der Ursprung eine `Cache-Control: private` Direktive `Cache-Control: no-storeCache-Control: no-cache`, oder für die Fehlercodes 404, 410, 414 oder 501 hinzufügt, wird die CloudFront Fehlerantwort nicht zwischengespeichert. CloudFront Ignoriert bei allen anderen Fehlercodes die `private` Direktiven `no-storeno-cache`, und und speichert die Fehlerantwort für den Wert Error Caching Minimum TTL zwischen.

Wenn der Ursprung weitere **`Cache-Control`** Direktiven hinzufügt oder keine Header hinzufügt, werden die Fehlerantworten für den Wert von Error CloudFront Caching Minimum TTL zwischengespeichert.

Wenn die Ablaufzeit für einen 4xx- oder 5xx-Statuscode für ein Objekt länger ist, als Sie warten möchten, und das Objekt wieder verfügbar ist, können Sie den zwischengespeicherten Fehlercode mit der URL des angefragten Objekts aufheben. Wenn Ihr Ursprung eine Fehlermeldung für mehrere Objekte zurückgibt, müssen Sie die Gültigkeit aller Objekte einzeln aufheben. Weitere Informationen zur Aufhebung der Gültigkeit von Objekten finden Sie unter Aufheben der Gültigkeit von Dateien zum Entfernen von Inhalten.

Wenn Sie das Caching für einen S3-Bucket-Ursprung aktiviert haben und in Ihrer CloudFront Distribution einen Fehler beim Zwischenspeichern einer Mindest-TTL von 0 Sekunden konfigurieren, wird immer noch ein Fehler beim Zwischenspeichern einer Mindest-TTL von 1 Sekunde für Fehler mit S3-Ursprung angezeigt. CloudFront tut dies, um Ihren Ursprung vor DDoS-Angriffen zu schützen. Dies gilt nicht für andere Arten von Ursprüngen.

Inhalte hinzufügen, entfernen oder ersetzen, die CloudFront verbreiten

In diesem Abschnitt wird erklärt, wie Sie sicherstellen CloudFront können, dass Sie auf die Inhalte zugreifen können, die Ihren Zuschauern angezeigt werden sollen, wie Sie die Objekte auf Ihrer Website oder in Ihrer Anwendung angeben und wie Sie Inhalte entfernen oder ersetzen.

Themen

- Inhalte hinzufügen und darauf zugreifen, die CloudFront verbreitet werden
- Verwenden Sie die Dateiversionierung, um Inhalte einer CloudFront Distribution zu aktualisieren oder zu entfernen
- Passen Sie das URL-Format für Dateien in an CloudFront
- Angeben eines Standardstammobjekts
- Aufheben der Gültigkeit von Dateien zum Entfernen von Inhalten
- Bereitstellen von komprimierten Dateien

Inhalte hinzufügen und darauf zugreifen, die CloudFront verbreitet werden

Wenn Sie Inhalte (Objekte) verteilen CloudFront möchten, fügen Sie Dateien zu einem der Ursprünge hinzu, die Sie für die Verteilung angegeben haben, und Sie stellen einen CloudFront Link zu den Dateien bereit. Ein CloudFront Edge-Standort ruft die neuen Dateien erst von einem Ursprung ab, wenn der Edge-Standort Viewer-Anfragen dafür erhält. Weitere Informationen finden Sie unter Wie liefert Inhalte CloudFront.

Wenn Sie eine Datei hinzufügen, die Sie verteilen CloudFront möchten, stellen Sie sicher, dass Sie sie zu einem der Amazon S3-Buckets hinzufügen, die in Ihrer Distribution angegeben sind, oder, für einen benutzerdefinierten Ursprung, zu einem Verzeichnis in der angegebenen Domain. Überprüfen Sie darüber hinaus, ob das Pfadmuster im entsprechenden Cache-Verhalten Anfragen an den richtigen Ursprung sendet.

Nehmen wir beispielsweise an, dass das Pfadmuster für ein Cache-Verhalten is `*.html`. Wenn Sie keine anderen Cache-Verhaltensweisen für das Weiterleiten von Anfragen an diesen Ursprungs-Server konfiguriert haben, leitet CloudFront nur `*.html`-Dateien weiter. In diesem Szenario verteilt

CloudFront beispielsweise niemals JPG-Dateien, die Sie auf den Ursprungs-Server hochladen, da Sie kein Cache-Verhalten erstellt haben, das JPG-Dateien enthält.

CloudFront Server bestimmen nicht den MIME-Typ für die Objekte, die sie bereitstellen. Wenn Sie eine Datei auf Ihren Ursprungs-Server hochladen, sollten Sie das `Content-Type`-Header-Feld dafür festlegen.

Verwenden Sie die Dateiversionierung, um Inhalte einer CloudFront Distribution zu aktualisieren oder zu entfernen

Um vorhandene Inhalte zu aktualisieren, die CloudFront für die Verteilung an Sie eingerichtet sind, empfehlen wir, in Datei- oder Ordnernamen eine Versionskennung zu verwenden. Auf diese Weise haben Sie die Kontrolle über die Verwaltung der bereitgestellten CloudFront Inhalte.

Aktualisieren vorhandener Dateien mithilfe von versionierten Dateinamen

Wenn Sie vorhandene Dateien in einer CloudFront Distribution aktualisieren, empfehlen wir Ihnen, eine Art Versionskennung entweder in Ihre Dateinamen oder in Ihre Verzeichnisnamen aufzunehmen, um eine bessere Kontrolle über Ihre Inhalte zu haben. Bei dieser Kennung kann es sich um einen Datum-Zeitstempel, eine fortlaufende Nummer oder eine andere Methode der Unterscheidung zwei Versionen desselben Objekts handeln.

Anstatt beispielsweise eine Grafikdatei mit image.jpg zu benennen, könnten Sie diese image_1.jpg nennen. Wenn Sie eine neue Version der Datei bereitstellen möchten, würden Sie die neue Datei mit image_2.jpg benennen und die Links in Ihrer Web-Anwendung oder auf Ihrer Website aktualisieren, damit sie auf image_2.jpg verweisen. Alternativ könnten Sie alle Grafiken in einem images_v1-Verzeichnis ablegen und, wenn Sie neue Versionen von einer oder mehreren Grafiken bereitstellen möchten, ein neues images_v2-Verzeichnis erstellen und Ihre Links aktualisieren, damit sie auf dieses Verzeichnis verweisen. Bei der Versionsverwaltung müssen Sie nicht warten, bis ein Objekt abläuft, bevor CloudFront eine neue Version davon bereitgestellt wird, und Sie müssen nicht für die Ungültigerklärung eines Objekts bezahlen.

Auch wenn Sie Ihre Dateien versionieren, sollten Sie ein Ablaufdatum festlegen. Weitere Informationen finden Sie unter Verwalten der Dauer, die Inhalte im Cache bleiben (Ablauf).

> **Note**
>
> Das Festlegen von versionsgesteuerten Dateinamen oder Verzeichnisnamen steht nicht mit dem Amazon S3-Objekt-Versioning im Zusammenhang.

Inhalte entfernen, damit sie CloudFront nicht verteilt werden

Sie können Dateien aus Ihrem Ursprungs-Server entfernen, die nicht länger in Ihrer CloudFront-Verteilung enthalten sein sollen. Zeigt Zuschauern jedoch CloudFront weiterhin Inhalte aus dem Edge-Cache an, bis die Dateien ablaufen.

Wenn Sie eine Datei direkt entfernen möchten, müssen Sie einen der folgenden Schritte ausführen:

- Verwenden der Dateiversionierung. Wenn Sie die Versionsverwaltung verwenden, haben verschiedene Versionen einer Datei unterschiedliche Namen, die Sie in Ihrer CloudFront Distribution verwenden können, um zu ändern, welche Datei an die Zuschauer zurückgegeben wird. Weitere Informationen finden Sie unter Aktualisieren vorhandener Dateien mithilfe von versionierten Dateinamen.
- Aufheben der Gültigkeit der Datei. Weitere Informationen finden Sie unter Aufheben der Gültigkeit von Dateien zum Entfernen von Inhalten.

Passen Sie das URL-Format für Dateien in an CloudFront

Nachdem Sie Ihren Ursprung mit den Objekten (Inhalten) eingerichtet haben, die Sie Ihren Zuschauern anbieten CloudFront möchten, müssen Sie die richtigen URLs verwenden, um auf diese Objekte in Ihrer Website oder im Anwendungscode zu verweisen, damit sie CloudFront bereitgestellt werden können.

Für den Domänennamen, den Sie in den URLs für Objekte auf Ihren Webseiten oder in Ihrer Webanwendung verwenden, haben Sie folgende Optionen:

- Der Domainname, der z. `d111111abcdef8.cloudfront.net` B. CloudFront automatisch zugewiesen wird, wenn Sie eine Distribution erstellen
- Ihr eigener Domänenname, z. B. `example.com`

Sie können beispielsweise eine der folgenden URLs verwenden, um die Datei zurückzugeben `image.jpg`:

`https://d111111abcdef8.cloudfront.net/images/image.jpg`

`https://example.com/images/image.jpg`

Sie verwenden dasselbe URL-Format, unabhängig davon, ob Sie den Inhalt in Amazon S3-Buckets oder an einem benutzerdefinierten Ursprung wie Ihren eigenen Webservern speichern.

> **Note**
>
> Das URL-Format richtet sich zum Teil nach dem Wert, den Sie für Origin Path in Ihrer Verteilung festlegen. Dieser Wert gibt CloudFront den obersten Verzeichnispfad für Ihre Objekte an. Weitere Informationen zum Einrichten eines Ursprungspfades beim Erstellen einer Verteilung finden Sie unter Ursprungspfad.

Weitere Informationen zu URL-Formaten finden Sie in den folgenden Abschnitten.

Verwenden Ihres eigenen Domainnamens (example.com)

Anstatt den Standard-Domainnamen zu verwenden, der Ihnen CloudFront zugewiesen wird, wenn Sie eine Distribution erstellen, können Sie einen alternativen Domainnamen hinzufügen, mit dem Sie einfacher arbeiten können, wie `example.com` Indem Sie Ihren eigenen Domainnamen mit einrichten CloudFront, können Sie eine URL wie diese für Objekte in Ihrer Distribution verwenden:

`https://example.com/images/image.jpg`

Wenn Sie HTTPS zwischen Zuschauern und verwenden möchten CloudFront, finden Sie weitere Informationen unterVerwenden von alternativen Domainnamen und HTTPS.

Verwenden eines abschließenden Schrägstrichs (/) in URLs

Wenn Sie URLs für Verzeichnisse in Ihrer CloudFront Distribution angeben, wählen Sie, ob Sie entweder immer einen abschließenden Schrägstrich verwenden oder niemals einen abschließenden Schrägstrich verwenden möchten. Wählen Sie z. B. nur eines der folgenden Formate für alle Ihre URLs aus:

`https://d111111abcdef8.cloudfront.net/images/`

```
https://d111111abcdef8.cloudfront.net/images
```

Warum ist das wichtig?

Beide Formate eignen sich für Links zu CloudFront Objekten, aber wenn Sie konsistent sind, können Sie Probleme vermeiden, wenn Sie ein Verzeichnis später für ungültig erklären möchten. CloudFront speichert URLs genau so, wie sie definiert sind, einschließlich nachstehender Schrägstriche. Wenn Ihr Format also inkonsistent ist, müssen Sie Verzeichnis-URLs mit und ohne Schrägstrich für ungültig erklären, um sicherzustellen, dass das Verzeichnis dadurch entfernt wird. CloudFront

Es ist lästig, beide URL-Formate deaktivieren zu müssen und kann zu zusätzlichen Kosten führen. Wenn Sie nämlich die Anzahl der Aufhebungen verdoppeln müssen, um beide Typen von URLs abzudecken, überschreiten Sie möglicherweise die für den Monat maximal zulässige Anzahl an kostenlosen Aufhebungen. Tritt dieser Fall ein, müssen Sie für alle Invalidierungen bezahlen, auch wenn in CloudFront jeweils nur ein Format für jede Verzeichnis-URL vorhanden ist.

Erstellen signierter URLs für eingeschränkte Inhalte

Wenn Sie den Zugriff auf bestimmte Inhalte einschränken möchten, können Sie signierte URLs erstellen. Wenn Sie beispielsweise Ihren Inhalt nur authentifizierten Benutzern zur Verfügung stellen möchten, können Sie URLs erstellen, die nur für einen bestimmten Zeitraum gültig sind der nur von einer bestimmten IP-Adresse aus zugänglich sind. Weitere Informationen finden Sie unter Bereitstellen privater Inhalte mit signierten URLs und signierten Cookies.

Angeben eines Standardstammobjekts

Sie können so konfigurieren CloudFront , dass ein bestimmtes Objekt (das Standard-Stammobjekt) zurückgegeben wird, wenn ein Benutzer (Betrachter) die Root-URL für Ihre Distribution anfordert, anstatt ein Objekt in Ihrer Distribution anzufordern. Durch die Verwendung eines Standardstammobjekt können Sie vermeiden, dass die Inhalte Ihrer Distribution preisgegeben werden.

Inhalt

- So geben Sie ein Standardstammobjekt an
- So funktioniert das Standardstammobjekt
- Wie CloudFront funktioniert das, wenn Sie kein Root-Objekt definieren

So geben Sie ein Standardstammobjekt an

Um zu verhindern, dass die Inhalte Ihrer Distribution offengelegt werden oder dass ein Fehler zurückgegeben wird, geben Sie ein Standardstammobjekt für Ihre Distribution an. Sie können den genauen Dateinamen oder den Pfad zur Datei angeben. Wenn es sich bei Ihrem Stammobjekt beispielsweise um eine `index.html`-Datei handelt, können Sie diesen Dateinamen angeben. Wenn sich Ihre `index.html`-Datei in einem anderen Ordner befindet, geben Sie stattdessen den Pfad an, z. B. *`exampleFolderName`*`/index.html`. Wenn Sie einen Pfad zum Standardstammobjekt festlegen, geben Viewer-Anforderungen an die Stamm-URL der Distribution die angegebene Datei aus diesem Pfad zurück. Sie können einen Dateipfad verwenden, um Ihre Inhalte flexibler am Ursprung zu organisieren, da sich Ihr Standardstammobjekt in einem Ordner statt auf Stammebene befinden kann.

So geben Sie ein Standardstammobjekt für Ihre Verteilung an

1. Laden Sie das Standardstammobjekt auf den Ursprung hoch, auf den Ihre Verteilung zeigt.

   Der Dateityp kann jeder von CloudFront unterstützte Typ sein. Eine Liste der Einschränkungen in Bezug auf den Dateinamen finden Sie im `DefaultRootObject` Element in DistributionConfig der CloudFront Amazon-API-Referenz.

   > **Note**
   >
   > Wenn der Dateiname des Standard-Stammobjekts zu lang ist oder ein ungültiges Zeichen enthält, wird der Fehler CloudFront zurückgegeben`HTTP 400 Bad Request - InvalidDefaultRootObject`. Außerdem wird der Code für 10 Sekunden CloudFront zwischengespeichert (standardmäßig) und die Ergebnisse werden in die Zugriffsprotokolle geschrieben.

2. Vergewissern Sie sich, dass die Berechtigungen für das Objekt CloudFront mindestens Lesezugriff gewähren.

   Weitere Informationen zu Amazon-S3-Berechtigungen finden Sie unter Identity and Access Management in Amazon S3 im Amazon-Simple-Storage-Service-Benutzerhandbuch.

3. Aktualisieren Sie Ihre Distribution, sodass sie auf das Standard-Stammobjekt verweist, indem Sie die CloudFront Konsole oder die CloudFront API verwenden.

   So geben Sie mithilfe der CloudFront Konsole ein Standard-Root-Objekt an:

a. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home.

b. Wählen Sie in der Liste der Verteilungen im oberen Bereich die zu aktualisierende Verteilung aus.

c. Wählen Sie im Bereich Einstellungen auf der Registerkarte Allgemein die Option Bearbeiten aus.

d. Geben Sie im Dialogfeld Einstellungen bearbeiten im Feld Standardstammobjekt den Dateinamen oder den Pfad des Standardstammobjekts ein.

> **Tip**
>
> Die Zeichenfolge darf nicht mit einem Schrägstrich (/) beginnen. Geben Sie nur den Objektnamen oder den Pfad zum Objekt an. Verwenden Sie beispielsweise `index.html` oder *`exampleFolderName`*`/index.html`. Die Angabe von **`/`***`exampleFolderName`*`/index.html` oder **`/`**`index.html` kann zu dem Fehler 403 „Zugriff verweigert“ führen.

e. Wählen Sie Änderungen speichern aus.

Um Ihre Konfiguration mithilfe der CloudFront API zu aktualisieren, geben Sie einen Wert für das `DefaultRootObject` Element in Ihrer Distribution an. Informationen zur Verwendung der CloudFront API zur Angabe eines Standard-Stammobjekts finden Sie UpdateDistribution in der CloudFront Amazon-API-Referenz.

4. Vergewissern Sie sich, dass Sie das Standardstammobjekt aktiviert haben, indem Ihre Stamm-URL anfragen. Wenn Ihr Standardstammobjekt im Browser nicht angezeigt wird, führen Sie die folgenden Schritte aus:

   a. Vergewissern Sie sich, dass Ihre Distribution vollständig bereitgestellt ist, indem Sie den Status Ihrer Distribution in der CloudFront Konsole einsehen.

   b. Wiederholen Sie die Schritte 2 und 3, um sicherzustellen, dass Sie die richtigen Berechtigungen erteilt und die Konfiguration Ihrer Verteilung ordnungsgemäß aktualisiert haben, um das Standardstammobjekt anzugeben.

So funktioniert das Standardstammobjekt

Angenommen, die folgende Anfrage zeigt beispielsweise auf das Objekt `image.jpg`:

`https://d111111abcdef8.cloudfront.net/image.jpg`

Im Gegensatz dazu zeigt die folgende Anfrage auf die Stamm-URL derselben Verteilung anstatt auf ein bestimmtes Objekt, wie im ersten Beispiel:

`https://d111111abcdef8.cloudfront.net/`

Wenn Sie ein Standardstammobjekt definieren, gibt eine Endbenutzeranfrage, welche den Stamm Ihrer Verteilung aufruft, das Standardstammobjekt zurück. Wenn Sie beispielsweise die Datei `index.html` als Standardstammobjekt bestimmen, gibt eine Anfrage für:

`https://d111111abcdef8.cloudfront.net/`

Rückgabe:

`https://d111111abcdef8.cloudfront.net/index.html`

> **Note**
>
> CloudFront bestimmt nicht, ob eine URL mit mehreren abschließenden Schrägstrichen (`https://d111111abcdef8.cloudfront.net///`) gleichbedeutend ist mit. `https://d111111abcdef8.cloudfront.net/` Ihr Ursprungsserver führt diesen Vergleich durch.

Wenn Sie ein Standardstammobjekt definieren, gibt eine Endbenutzeranfrage für ein Unterverzeichnis Ihrer Verteilung nicht das Standardstammobjekt zurück. Nehmen wir zum Beispiel an, `index.html` es handelt sich um Ihr Standard-Root-Objekt, das eine Endbenutzeranforderung für das `install` Verzeichnis in Ihrer Distribution CloudFront erhält: CloudFront

`https://d111111abcdef8.cloudfront.net/install/`

CloudFront gibt das Standard-Stammobjekt nicht zurück, auch wenn eine Kopie von im `install` Verzeichnis `index.html` erscheint. Wenn Sie jedoch einen Pfad zu Ihrem Standard-Stammobjekt angegeben haben, gibt (`install/index.html`) CloudFront das Standard-Stammobjekt für Endbenutzeranfragen für das `install` Verzeichnis zurück

Wenn Sie Ihre Verteilung so konfigurieren, dass alle HTTP-Methoden, die CloudFront unterstützt, zulässig sind, gilt das Standardstammobjekt für alle Methoden. Wenn Ihr Standard-Root-Objekt

beispielsweise index.php ist und Sie Ihre Anwendung schreiben, um eine `POST` Anfrage an das Stammverzeichnis Ihrer Domain (https://example.com) zu senden, CloudFront sendet die Anfrage an https://example.com/index.php.

Das Verhalten von CloudFront Standard-Root-Objekten unterscheidet sich vom Verhalten von Amazon S3-Indexdokumenten. Wenn Sie einen Amazon S3-Bucket als Website konfigurieren und das Indexdokument angeben, gibt Amazon S3 das Indexdokument auch dann zurück, wenn ein Benutzer ein Unterverzeichnis im Bucket anfragt. (Eine Kopie des Indexdokuments muss in allen Unterverzeichnissen angezeigt werden.) Weitere Informationen zum Konfigurieren von Amazon-S3-Buckets als Websites und zu Indexdokumenten finden Sie im Kapitel Hosting von Websites auf Amazon S3 im Benutzerhandbuch zu Amazon Simple Storage Service.

> **Important**
>
> Denken Sie daran, dass ein Standard-Stammobjekt nur für Ihre CloudFront Distribution gilt. Sie müssen noch die Sicherheit für Ihren Ursprung verwalten. Wenn Sie beispielsweise einen Amazon S3-Ursprung verwenden, müssen Sie Ihre Amazon S3-Bucket-ACLs trotzdem entsprechend einrichten, um sicherzustellen, dass Sie die gewünschte Zugriffsebene für Ihren Bucket haben.

Wie CloudFront funktioniert das, wenn Sie kein Root-Objekt definieren

Wenn Sie kein Standardstammobjekt definieren, werden Anfragen für den Stamm Ihrer Verteilung an Ihren Ursprungs-Server weitergeleitet. Wenn Sie einen Amazon S3-Ursprung verwenden, kann Folgendes zurückgegeben werden:

- Eine Liste der Inhalte Ihres Amazon S3-Buckets — Unter einer der folgenden Bedingungen sind die Inhalte Ihrer Herkunft für jeden sichtbar, der auf Ihre Distribution CloudFront zugreift:
  - Ihr Bucket ist nicht ordnungsgemäß konfiguriert.
  - Die Amazon S3-Berechtigungen für den Bucket, der Ihrer Distribution zugeordnet ist, und für die Objekte im Bucket gewähren allen Zugriff.
  - Ein Endbenutzer greift mit der Stamm-URL Ihres Ursprungs auf den Ursprung zu.
- Eine Liste der privaten Inhalte Ihrer Herkunft — Wenn Sie Ihren Ursprung als private Distribution konfigurieren (nur Sie und Sie CloudFront haben Zugriff), sind die Inhalte des Amazon S3-Buckets, der Ihrer Distribution zugeordnet ist, für jeden sichtbar, der über die Anmeldeinformationen für den Zugriff auf Ihre Distribution verfügt CloudFront. In diesem Fall können Benutzer nicht über die

Stamm-URL Ihres Ursprungs auf Ihre Inhalte zugreifen. Weitere Informationen zum Verteilen von privaten Inhalten finden Sie unter the section called “Beschränken von Inhalten mit signierten URLs und signierten Cookies”.

- `Error 403 Forbidden`— CloudFront gibt diesen Fehler zurück, wenn die Berechtigungen für den Amazon S3-Bucket, der Ihrer Distribution zugeordnet ist, oder die Berechtigungen für die Objekte in diesem Bucket den Zugriff für CloudFront und für alle verweigern.

Aufheben der Gültigkeit von Dateien zum Entfernen von Inhalten

Wenn Sie eine Datei vor ihrem Ablauf aus CloudFront Edge-Caches entfernen müssen, können Sie einen der folgenden Schritte ausführen:

- Heben Sie die Gültigkeit der Datei in Edge-Caches auf. Wenn ein Viewer die Datei das nächste Mal anfordert, CloudFront kehrt er zum Ursprung zurück, um die neueste Version der Datei abzurufen.
- Verwenden Sie Datei-Versioning, um eine andere Version der Datei mit einem anderen Namen bereitzustellen. Weitere Informationen finden Sie unter Aktualisieren vorhandener Dateien mithilfe von versionierten Dateinamen.

Themen

- Auswählen zwischen der Aufhebung der Gültigkeit von Dateien und der Verwendung von versionierten Dateinamen
- Ermitteln, welche Pfade ungültig gemacht werden sollen
- Was Sie zur Aufhebung der Gültigkeit von Pfaden wissen müssen
- Aufhebung der Gültigkeit durch Cache-Tags
- Was Sie zur Aufhebung der Gültigkeit von Tags wissen müssen
- Aufheben der Gültigkeit von Dateien
- Kosten der Aufhebung der Gültigkeit von Dateien

Auswählen zwischen der Aufhebung der Gültigkeit von Dateien und der Verwendung von versionierten Dateinamen

Um die Versionen von Dateien zu steuern, die von Ihrer Verteilung bereitgestellt werden, können Sie entweder die Gültigkeit der Dateien aufheben oder diesen versionierte Dateinamen zuweisen.

Wenn Sie Ihre Dateien häufig aktualisieren möchten, empfehlen wir, hauptsächlich Datei-Versioning zu verwenden. Die Gründe dafür sind folgende:

- Mit Versioning können Sie steuern, welche Datei von einer Anfrage zurückgegeben wird, auch wenn der Benutzer über eine Version verfügt, die entweder lokal oder hinter einem Unternehmens-Caching-Proxy zwischengespeichert ist. Wenn Sie die Gültigkeit der Datei aufheben, kann der Benutzer möglicherweise noch die alte Version aus diesen Caches anzeigen, bis sie abläuft.
- CloudFront Die Zugriffsprotokolle enthalten die Namen Ihrer Dateien, sodass die Versionierung die Analyse der Ergebnisse von Dateiänderungen erleichtert.
- Versioning bietet eine Möglichkeit, verschiedenen Benutzern verschiedene Versionen von Dateien bereitzustellen.
- Versioning vereinfacht den Wechsel zwischen verschiedenen Dateirevisionen.
- Versioning ist kostengünstiger. Sie müssen immer noch für die CloudFront Übertragung neuer Versionen Ihrer Dateien an Edge-Standorte bezahlen, aber Sie müssen nicht für ungültig werdende Dateien bezahlen.

Weitere Informationen zu Datei-Versioning finden Sie unter Aktualisieren vorhandener Dateien mithilfe von versionierten Dateinamen.

Ermitteln, welche Pfade ungültig gemacht werden sollen

Wenn Sie die Gültigkeit mehrerer Dateien aufheben möchten, wie beispielsweise aller Dateien in einem Verzeichnis oder aller Dateien, deren Namen mit den gleichen Zeichen beginnen, können Sie den Platzhalter * am Ende des Aufhebungspfads einfügen. Weitere Informationen zur Verwendung des Platzhalters * finden Sie unter Invalidation paths.

Um die Gültigkeit von Dateien aufzuheben, können Sie entweder den Pfad für einzelne Dateien festlegen oder einen Pfad, der mit dem Platzhalter * endet und für ein oder mehrere Dateien gelten könnte, wie in den folgenden Beispielen gezeigt:

- `/images/image1.jpg`
- `/images/image*`
- `/images/*`

Wenn Sie die Gültigkeit ausgewählter Dateien aufheben möchten, Ihre Benutzer jedoch nicht unbedingt auf alle Dateien in Ihrem Ursprung zugreifen, können Sie ermitteln, welche Dateien

Betrachter von CloudFront abgefragt haben, und nur die Gültigkeit dieser Dateien aufheben. Um festzustellen, welche Dateien von den Viewern angefordert wurden, aktivieren Sie die CloudFront Zugriffsprotokollierung. Weitere Informationen zu Zugriffsprotokollen finden Sie unter Zugriffsprotokolle (Standardprotokolle).

Was Sie zur Aufhebung der Gültigkeit von Pfaden wissen müssen

Wenn Sie eine Datei angeben, die ungültig gemacht werden soll, beachten Sie die folgenden Informationen:

Groß- und Kleinschreibung

Bei Aufhebungspfaden wird zwischen Groß- und Kleinschreibung unterschieden. `/images/Image.jpg` und `/images/image.jpg` geben beispielsweise zwei verschiedene Dateien an.

Änderung der URI mithilfe einer Lambda-Funktion

Wenn Ihre CloudFront Distribution bei Viewer-Anforderungsereignissen eine Lambda-Funktion auslöst und die Funktion den URI der angeforderten Datei ändert, empfehlen wir, beide URIs ungültig zu machen, um die Datei aus den Edge-Caches zu entfernen: CloudFront

- Der URI in der Viewer-Anfrage
- Die URI, nachdem sie von der Funktion geändert wurde

Example Beispiel

Angenommen, Ihre Lambda-Funktion ändert den URI für eine Datei von diesem:

`https://d111111abcdef8.cloudfront.net/index.html`

in einen URI, der ein Sprachverzeichnis enthält:

`https://d111111abcdef8.cloudfront.net/en/index.html`

Um die Datei ungültig zu machen, müssen Sie die folgenden Pfade angeben:

- `/index.html`
- `/en/index.html`

Weitere Informationen finden Sie unter Invalidation paths.

Standardstammobjekt

Um die Gültigkeit des Standardstammobjekts (Datei) aufzuheben, geben Sie den Pfad auf die Weise an, auf die Sie auch den Pfad für alle anderen Dateien angeben. Weitere Informationen finden Sie unter So funktioniert das Standardstammobjekt.

Weiterleiten von Cookies

Wenn Sie so konfiguriert haben CloudFront , dass Cookies an Ihren Ursprung weitergeleitet werden, können CloudFront Edge-Caches mehrere Versionen der Datei enthalten. Wenn Sie eine Datei für ungültig erklären, wird jede zwischengespeicherte Version der Datei CloudFront ungültig, unabhängig von den zugehörigen Cookies. Sie können nicht auf der Grundlage der zugehörigen Cookies selektiv die Gültigkeit einiger Versionen aufheben und anderer Versionen nicht. Weitere Informationen finden Sie unter Zwischenspeichern von Inhalten auf der Grundlage von Cookies.

Weiterleiten von Headern

Wenn Sie so konfiguriert haben CloudFront , dass eine Liste von Headern an Ihren Ursprung weitergeleitet und basierend auf den Werten der Header zwischengespeichert wird, können CloudFront Edge-Caches mehrere Versionen der Datei enthalten. Wenn Sie die Gültigkeit einer Datei aufheben, hebt CloudFront die Gültigkeit aller zwischengespeicherten Versionen der Datei auf, unabhängig von den Header-Werten. Sie können nicht auf der Grundlage der Header-Werte selektiv die Gültigkeit einiger Versionen aufheben und anderer Versionen nicht. (Wenn Sie so konfigurieren CloudFront , dass alle Header an Ihren Ursprung weitergeleitet werden, werden Ihre Dateien CloudFront nicht zwischengespeichert.) Weitere Informationen finden Sie unter Zwischenspeichern von Inhalten auf der Grundlage von Anforderungsheadern.

Weiterleiten von Abfragezeichenfolgen

Wenn Sie CloudFront die Weiterleitung von Abfragezeichenfolgen an Ihren Ursprung konfiguriert haben, müssen Sie die Abfragezeichenfolgen bei der Ungültigerklärung von Dateien einbeziehen, wie in den folgenden Beispielen gezeigt:

- `/images/image.jpg?parameter1=a`
- `/images/image.jpg?parameter1=b`

Wenn Client-Anfragen fünf verschiedene Abfragezeichenfolgen für dieselbe Datei enthalten, können Sie entweder die Gültigkeit der Datei fünfmal aufheben, einmal für jede Abfragezeichenfolgen, oder den Platzhalter * im Aufhebungspfad verwenden wie im folgenden Beispiel gezeigt:

```
/images/image.jpg*
```

Weitere Informationen zur Verwendung von Platzhaltern im Aufhebungspfad finden Sie unter Invalidation paths.

Weitere Informationen zu Abfragezeichenfolgen finden Sie unter Zwischenspeichern von Inhalten auf der Grundlage von Abfragezeichenfolgeparametern.

Um zu ermitteln, welche Abfragezeichenfolgen verwendet werden, können Sie die CloudFront-Protokollierung aktivieren. Weitere Informationen finden Sie unter Zugriffsprotokolle (Standardprotokolle).

Maximal zulässig

Weitere Informationen zur maximalen Anzahl der zulässigen Gültigkeitsaufhebungen finden Sie unter Kontingente für Aufhebungen.

Microsoft Smooth Streaming-Dateien

Sie können die Gültigkeit von Mediendateien im Format Microsoft Smooth Streaming nicht aufheben, wenn Smooth Streaming für das entsprechende Cacheverhalten aktiviert ist.

Non-ASCII oder unsichere Zeichen im Pfad

Wenn der Pfad Nicht-ASCII-Zeichen oder unsichere Zeichen enthält, wie in RFC 1738 definiert, diese Zeichen. URL-encode Verwenden Sie URL-encode keine anderen Zeichen im Pfad, da sonst die alte Version der CloudFront aktualisierten Datei nicht ungültig wird.

> **Important**
>
> Verwende das ~ Zeichen nicht in deinem Pfad. CloudFront unterstützt dieses Zeichen nicht für Ungültigerklärungen, unabhängig davon, ob es das ist URL-encoded oder nicht.

Aufhebungspfade

Der Pfad ist der relative Pfad in Bezug auf die Verteilung. Um beispielsweise die Gültigkeit der Datei unter `https://d111111abcdef8.cloudfront.net/images/image2.jpg` aufzuheben, würden Sie `/images/image2.jpg` angeben.

> **Note**
>
> In der CloudFront-Konsole können Sie den vorangestellten Schrägstrich im Pfad wie folgt weglassen: `images/image2.jpg`. Wenn Sie die CloudFront API direkt verwenden, müssen die Invalidierungspfade mit einem führenden Schrägstrich beginnen.

Mit dem Platzhalter * können Sie auch die Gültigkeit mehrerer Dateien gleichzeitig aufheben. Der Platzhalter *, der 0 oder mehr Zeichen ersetzt, muss das letzte Zeichen im Aufhebungspfad sein.

> **Important**
>
> Zur Verwendung von Platzhaltern (*) bei der Aufhebung der Gültigkeit müssen Sie den Platzhalter am Ende des Pfads platzieren. Sternchen (*), die an einer anderen Stelle eingefügt werden, werden als wörtliche Zeichenübereinstimmung behandelt und nicht als Aufhebung der Gültigkeit eines Platzhalters.

Wenn Sie das AWS Command Line Interface (AWS CLI) verwenden, um Dateien für ungültig zu erklären, und Sie einen Pfad angeben, der den * Platzhalter enthält, müssen Sie den Pfad wie in Anführungszeichen (") setzen. `"/*"`

Die maximale Länge eines Pfads beträgt 4 000 Zeichen.

Example Beispiel: Aufhebungspfade

- Um alle Dateien in einem Verzeichnis, all seine Unterverzeichnisse und alle Dateien in dem Verzeichnis und den Unterverzeichnissen für ungültig zu erklären:

  `/directory-path/*`
- So heben Sie die Gültigkeit eines Verzeichnisses, aller Unterverzeichnisse und aller Dateien im Verzeichnis und den Unterverzeichnissen auf:

  `/directory-path*`
- So heben Sie die Gültigkeit aller Dateien auf, die denselben Namen, aber verschiedene Dateinamenerweiterungen haben, wie beispielsweise logo.jpg, logo.png und logo.gif:

  `/directory-path/file-name.*`

- So heben Sie die Gültigkeit aller Dateien in einem Verzeichnis auf, deren Dateinamen mit denselben Zeichen beginnen (beispielsweise alle Dateien für ein Video im HLS-Format), unabhängig von der Dateinamenerweiterung:

  */directory-path/initial-characters-in-file-name**

- Wenn Sie CloudFront den Cache auf der Grundlage von Abfragezeichenfolgenparametern konfigurieren und jede Version einer Datei für ungültig erklären möchten, gehen Sie wie folgt vor:

  */directory-path/file-name.file-name-extension**

- So heben Sie die Gültigkeit aller Dateien in einer Distribution auf:

  ```
  /*
  ```

  Weitere Informationen zum AUfheben der Gültigkeit von Dateien, wenn Sie eine Lambda-Funktion zum Ändern der URI verwenden, finden Sie unter Changing the URI Using a Lambda Function.

  Wenn es sich bei dem Aufhebungspfad um ein Verzeichnis handelt und wenn Sie über keine Standard-Methode zum Angeben von Verzeichnissen verfügen – mit oder ohne einen abschließenden Schrägstrich (/) –, empfehlen wir, die Gültigkeit des Verzeichnisses sowohl mit als auch ohne abschließenden Schrägstrich aufzuheben, beispielsweise `/images` und `/images/`.

Signierte URLs

  Wenn Sie signierte URLs verwenden, heben Sie die Gültigkeit einer Datei auf, indem Sie nur den Teil der URL vor dem Fragezeichen (?) einfügen.

Aufhebung der Gültigkeit durch Cache-Tags

Amazon CloudFront unterstützt Tag-Invalidierungen, sodass Sie zwischengespeicherte Objekte auf der Grundlage semantischer Tags und nicht anhand von URL-Pfaden für ungültig erklären können. Auf diese Weise können Sie die Cache-Invalidierung flexibel steuern, ohne dass Ihre URL-Struktur Ihrer Invalidierungsstrategie entsprechen muss.

Bevor Sie Inhalte anhand von Cache-Tags für ungültig erklären, lesen Sie die Anforderungen und Überlegungen unter. Was Sie zur Aufhebung der Gültigkeit von Tags wissen müssen

So funktioniert die Tag-Invalidierung

1. Konfiguriere deine Distribution: Füge deiner Distribution eine `CacheTagConfig` hinzu, die den HTTP-Header-Namen angibt, den dein Ursprung für die Rückgabe von Cache-Tags verwendet.
2. Markieren Sie Objekte an Ihrem Ursprung: Wenn Sie ein Objekt zurückgeben, das Sie mit Tags zwischenspeichern möchten, konfigurieren Sie Ihren Ursprung so, dass der angegebene Header in HTTP-Antworten mit kommagetrennten Tag-Werten enthalten ist.

```
HTTP/1.1 200 OK
Content-Type: text/html
x-amz-meta-cache-tag: product:electronics, category:tv, brand:example
Cache-Control: max-age=3600
```

> **Note**
>
> Für S3-Ursprünge können Sie Cache-Tags als Metadaten an Ihre S3-Objekte anhängen. Sie können einen Metadateneintrag mit einem Schlüssel Ihrer Wahl (Beispiel: `cache-tag`) und einer kommagetrennten Liste von Tags als Wert hinzufügen (Beispiel:`product:electronics, category:tv, brand:example`).
> S3 zeigt Objektmetadaten als Antwort-Header an, denen ein Präfix `x-amz-meta-<Key>` vorangestellt ist, sodass ein Metadatenschlüssel von als Header zurückgegeben `cache-tag` würde. `x-amz-meta-cache-tag` Sie können das `HeaderName` in your `CacheTagConfig` to setzen, `x-amz-meta-cache-tag` um das Senden von Ungültigerklärungen an diese Tags zu ermöglichen.

> **Note**
>
> Alternativ können Sie Ihrem Cache-Verhalten eine Lambda @Edge -Funktion der Ursprungsantwort hinzufügen, um Cache-Tag-Header hinzuzufügen. Wenn Sie Lambda @Edge verwenden, muss der Header-Name nicht dem Format folgen. `x-amz-meta-<Key>`

3. Nach Tag ungültig machen: Verwenden Sie die `CreateInvalidation` API mit dem # Präfix, um alle Objekte, die mit dem angegebenen Tag zwischengespeichert wurden, für ungültig zu erklären.

```
aws cloudfront create-invalidation \
  --distribution-id distribution_ID \
  --paths "#product:electronics"
```

Dadurch werden ALLE zwischengespeicherten Objekte ungültig, die das Tag enthalten `product:electronics`, unabhängig von ihrem URL-Pfad.

Anforderungen an das Tag-Format

Header-Wert der Origin-Antwort:

- Die Tags sind im Header-Wert durch Kommas getrennt.
- Leerzeichen am Anfang und Ende der einzelnen Tags werden gekürzt.
- Beide `tag1,tag2,tag3` und `tag1, tag2, tag3` sind gültig und gleichwertig.

Einzelne Tag-Werte:

- Sichtbare ASCII-Zeichen (33—126)
- Keine Steuerzeichen, Leerzeichen oder Kommas
- Case-insensitive
- Maximal 256 Zeichen pro Tag
- Maximal 50 Tags pro Objekt (zusätzliche Tags werden ignoriert)

Konfiguration einer Verteilung für Tag-Invalidierungen

Um Tag-Invalidierungen zu aktivieren, fügen Sie Ihrer Distributionskonfiguration eine `CacheTagConfig` hinzu:

```
aws cloudfront create-distribution \
  --distribution-config '{
    "CallerReference": "my-distribution",
    "CacheTagConfig": {
      "HeaderName": "x-amz-meta-cache-tag"
    },
    "DefaultCacheBehavior": {
      "TargetOriginId": "myOrigin",
```

```
      "ViewerProtocolPolicy": "redirect-to-https",
      "CachePolicyId": "658327ea-f89d-4fab-a63d-7e88639e58f6"
    },
    "Origins": { ... },
    "Enabled": true,
    "Comment": "Distribution with tag support"
  }'
```

Tag-Invalidierungen erstellen

Verwenden Sie das # Präfix im `--paths` Parameter, um Tags anzugeben:

```
# Invalidate all objects tagged with "user1"
aws cloudfront create-invalidation \
  --distribution-id distribution_ID \
  --paths "#user1"

# Invalidate objects matching any of multiple tags (OR logic)
aws cloudfront create-invalidation \
  --distribution-id distribution_ID \
  --paths "#user1" "#product-category:electronics"

# Mix path and tag invalidations in one batch
aws cloudfront create-invalidation \
  --distribution-id distribution_ID \
  --paths "/index.html" "#user1" "/images/*" "#product-category:electronics"
```

Überprüfung des Invalidierungsstatus

Wird verwendet`GetInvalidation`, um den Status zu überprüfen. Die Antwort umfasst sowohl Pfad- als auch Tag-Elemente:

```
aws cloudfront get-invalidation \
  --distribution-id distribution_ID \
  --id invalidation_ID
```

Wichtige Überlegungen

- Opt-in erforderlich: Die Tag-Invalidierung funktioniert nur bei Distributionen, die konfiguriert wurden`CacheTagConfig`. Distributionen ohne diese Konfiguration ignorieren Cache-Tag-Header aus dem Ursprung.

- Grenzwerte für die Tag-Verarbeitung: CloudFront verarbeitet bis zu 50 Tags pro zwischengespeichertem Objekt. Wenn eine Quellantwort mehr als 50 Tags enthält, werden die zusätzlichen Tags, die das Limit überschreiten, nicht gespeichert.
- Änderung des Headernamens: Tag-Invalidierungen werden anhand der aktuellen `CacheTagConfig` Konfiguration abgeglichen. Wenn Sie den `HeaderName` in ändern`CacheTagConfig`, werden Ungültigerklärungen für Objekte, die mit Tags unter dem alten Header-Namen zwischengespeichert wurden, nicht mehr ausgewertet. Wenn Sie den Header-Namen ändern müssen, beginnen Sie, sowohl die neuen als auch die alten Cache-Tag-Header zusammen mit Ihren Objekten zurückzugeben, und führen Sie dann eine Pfadinvalidierung durch (z. B.`/*`) oder machen Sie vorhandene Tags ungültig, bevor Sie den Header-Namen ändern, um zu vermeiden, dass veraltete Inhalte bereitgestellt werden. Sobald das erledigt ist, können Sie das Senden der alten Cache-Tags-Header mit Ihren Objekten beenden.
- Entfernen CacheTagConfig: Wenn Sie `CacheTagConfig` aus einer Distribution entfernen, wird das Extrahieren von Tags aus den ursprünglichen Antworten CloudFront beendet. Bestehende zwischengespeicherte Objekte mit Tags werden normal bereitgestellt, bis sie ablaufen oder anhand des Pfads für ungültig erklärt werden.
- Abwärtskompatibilität: Bestehende Pfad- und Platzhalterinvalidierungen funktionieren weiterhin unverändert. Tag-Invalidierungen sind additiv — Sie können beide Methoden in derselben Distribution verwenden.
- Distribution-Mandanten: Die Invalidierung von Tags wird mithilfe der API auch für Distributionsmandanten unterstützt. `CreateInvalidationForDistributionTenant`

Beispielanwendungsfälle

Beispiel 1: E-commerce Produktkatalog

Eine E-Commerce-Website kennzeichnet zwischengespeicherte Produktseiten mit Produkt, Kategorie und Marke:

```
x-amz-meta-cache-tag: category:electronics, brand:acme, product:12345
```

Wenn ACME ihr Branding aktualisiert, machen Sie alle ACME-Produkte gleichzeitig ungültig:

```
aws cloudfront create-invalidation \
  --distribution-id distribution_ID \
  --paths "#brand:acme"
```

Beispiel 2: Plattform für Inhalte User-generated

Eine Plattform kennzeichnet zwischengespeicherte Inhalte mit der Benutzer-ID des Eigentümers:

```
x-amz-meta-cache-tag: user:12345, content-type:image
```

Wenn ein Nutzer sein Konto schließt, mache seinen gesamten Inhalt ungültig, nachdem du ihn vom Ursprung entfernt hast:

```
aws cloudfront create-invalidation \
  --distribution-id distribution_ID \
  --paths "#user:12345"
```

Beispiel 3: Versionierung von Inhalten

Ein CMS kennzeichnet Inhalte mit Versions-Identifikatoren:

```
x-amz-meta-cache-tag: version:v2, template:homepage
```

Wenn Sie eine neue Version bereitstellen, machen Sie den gesamten v2-Inhalt ungültig:

```
aws cloudfront create-invalidation \
  --distribution-id distribution_ID \
  --paths "#version:v2"
```

Was Sie zur Aufhebung der Gültigkeit von Tags wissen müssen

Wenn Sie ein Tag angeben, das ungültig gemacht werden soll, beachten Sie die folgenden Informationen:

Opt-in erforderlich

Die Tag-Invalidierung funktioniert nur bei Distributionen, die konfiguriert wurden`CacheTagConfig`. Distributionen ohne diese Konfiguration ignorieren Cache-Tag-Header aus dem Ursprung. Weitere Informationen finden Sie unter Aufhebung der Gültigkeit durch Cache-Tags.

Präfix für die Invalidierung von Tags

Elemente für die Invalidierung von Tags müssen mit dem # Zeichen beginnen. Um beispielsweise alle zwischengespeicherten Objekte, die mit gekennzeichnet sind, ungültig zu

machen`product:electronics`, geben Sie dies `#product:electronics` in der Liste an. `Paths.Items`

Groß-/Kleinschreibung

Bei Tag-Werten wird nicht zwischen Groß- und Kleinschreibung unterschieden. Zum Beispiel `#Product:Electronics` und `#product:electronics` beziehen sich auf dasselbe Tag.

Tag-Format

Jeder Tag-Wert darf nur sichtbare ASCII-Zeichen (33—126) ohne Kommas enthalten. Die maximale Länge eines Tags beträgt 256 Zeichen.

Keine Unterstützung für Platzhalter

Im Gegensatz zu Pfadinvalidierungen unterstützen Tag-Invalidierungselemente den Platzhalter nicht. * Jedes Element mit # Präfix entspricht dem exakten Tag-Wert.

Maximale Anzahl von Tags pro zwischengespeichertem Objekt

CloudFront verarbeitet bis zu 50 Tags pro zwischengespeichertem Objekt. Wenn eine Quellantwort mehr als 50 Tags im konfigurierten Header enthält, werden die zusätzlichen Tags, die das Limit überschreiten, nicht gespeichert und können nicht für die Invalidierung verwendet werden.

Weiterleitung von Cookies, Headern und Abfragezeichenfolgen

Wenn Sie nach Tag ungültig machen, wird jede zwischengespeicherte Variante jedes Objekts CloudFront ungültig, das das angegebene Tag trägt, unabhängig von den zugehörigen Cookies, Headern oder Abfragezeichenfolgenparametern. Sie können einige Varianten nicht selektiv für ungültig erklären und andere nicht.

Gemischte Ungültigerklärungen von Pfaden und Tags

Sie können in einer Anfrage sowohl Pfadelemente (z. B.`/images/*`) als auch Tag-Elemente (z. B.`#brand:acme`) in denselben Invalidierungsstapel aufnehmen. `CreateInvalidation` Jedes Element zählt als ein Invalidierungspfad zur Anrechnung Ihrer Grenzwerte.

Maximum erlaubt

Für Tag-Invalidierungselemente gelten dieselben Grenzwerte für die gleichzeitige Invalidierung wie für Pfadelemente. Jedes Tag-Element zählt als ein Invalidierungspfad. Weitere Informationen finden Sie unter Kontingente für Aufhebungen.

Den Header-Namen ändern

Wenn Sie den `HeaderName` In ändern`CacheTagConfig`, werden Ungültigerklärungen für Tags, die unter dem alten Header-Namen extrahiert wurden, nicht mehr ausgewertet. Bevor Sie den Header-Namen ändern:

1. Beginne, sowohl den alten als auch den neuen Cache-Tag-Header von deinem Ursprung zurückzugeben.
2. Führen Sie eine Pfadinvalidierung durch (z. B.`/*`) oder machen Sie vorhandene Tags ungültig.
3. `CacheTagConfig`Mit dem Neuen aktualisieren. `HeaderName`
4. Hören Sie auf, den alten Header von Ihrem Ursprung zurückzugeben.

Vertrieb (Mieter)

Die Invalidierung von Tags wird auch für Distributionsmandanten mithilfe der `CreateInvalidationForDistributionTenant` API unterstützt.

Aufheben der Gültigkeit von Dateien

Sie können die CloudFront Konsole verwenden, um eine Invalidierung zu erstellen und auszuführen, eine Liste der zuvor eingereichten Invalidierungen anzuzeigen und detaillierte Informationen zu einer einzelnen Invalidierung anzuzeigen. Sie können auch eine vorhandene Aufhebung kopieren, die Liste der Dateipfade bearbeiten und die bearbeiteten Aufhebungen ausführen. Sie können diese Aufhebungen nicht aus der Liste entfernen.

Inhalt

- Aufheben der Gültigkeit von Dateien
- Kopieren, Bearbeiten und erneutes Ausführen einer vorhandenen Gültigkeitsaufhebung
- Stornieren von Aufhebungen
- Auflisten von Aufhebungen
- Anzeigen von Informationen zu einer Aufhebung

Aufheben der Gültigkeit von Dateien

Gehen Sie wie folgt vor, um Dateien mithilfe der CloudFront Konsole für ungültig zu erklären.

Console

So heben Sie die Gültigkeit von Dateien auf (Konsole)

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home.
2. Wählen Sie die Verteilung aus, für die Sie Dateien ungültig machen möchten.
3. Wählen Sie die Registerkarte Invalidations aus.
4. Wählen Sie Gültigkeitsaufhebung erstellen aus.
5. Geben Sie für die Dateien, die Sie ungültig machen möchten, einen Aufhebungspfad pro Zeile ein. Weitere Informationen zur Angabe von Aufhebungspfaden finden Sie unter Was Sie zur Aufhebung der Gültigkeit von Pfaden wissen müssen.

   > **Important**
   >
   > Gehen Sie bei der Angabe der Dateipfade vorsichtig vor. Sie können einen Aufhebungsantrag nicht abbrechen, wenn Sie ihn begonnen haben.

6. Wählen Sie Gültigkeitsaufhebung erstellen aus.

CloudFront API

Weitere Informationen zur Ungültigerklärung von Objekten und zum Anzeigen von Informationen zu Invalidierungen finden Sie in den folgenden Themen in der CloudFront Amazon-API-Referenz:

- CreateInvalidation
- ListInvalidations
- GetInvalidation

> **Note**
>
> Wenn Sie das AWS Command Line Interface (AWS CLI) verwenden, um Dateien für ungültig zu erklären, und Sie einen Pfad angeben, der den * Platzhalter enthält, müssen Sie den Pfad in Anführungszeichen (") setzen, wie im folgenden Beispiel:

```
aws cloudfront create-invalidation --distribution-id distribution_ID --paths
 "/*"
```

Kopieren, Bearbeiten und erneutes Ausführen einer vorhandenen Gültigkeitsaufhebung

Sie können eine zuvor erstellte Aufhebung kopieren, die Liste der Aufhebungspfade aktualisieren und die aktualisierte Aufhebung ausführen. Sie können nicht eine vorhandene Gültigkeitsaufhebung kopieren, die Aufhebungspfade aktualisieren und dann die aktualisierte Aufhebung speichern, ohne sie auszuführen.

> **Important**
>
> Wenn Sie eine Invalidierung kopieren, die noch im Gange ist, aktualisieren Sie die Liste der Invalidierungspfade und führen Sie dann die aktualisierte Invalidierung aus. Die kopierte Invalidierung CloudFront wird nicht gestoppt oder gelöscht. Wenn im Original und in der Kopie Pfade für die Invalidierung vorkommen, CloudFront wird zweimal versucht, die Dateien für ungültig zu erklären. Beide Invalidierungen werden dann auf Ihre maximale Anzahl an kostenlosen Invalidierungen für den Monat angerechnet. Wenn Sie die maximale Anzahl der kostenlosen Aufhebungen bereits erreicht haben, fallen Gebühren für beide Aufhebungen der einzelnen Dateien an. Weitere Informationen finden Sie unter Kontingente für Aufhebungen.

So kopieren, bearbeiten und führen Sie eine vorhandene Aufhebung erneut aus

1. Melden Sie sich bei der an und öffnen Sie die Konsole unter AWS-Managementkonsole . CloudFront https://console.aws.amazon.com/cloudfront/v4/home
2. Wählen Sie die Verteilung mit der Aufhebung aus, die Sie kopieren möchten.
3. Wählen Sie die Registerkarte Invalidations aus.
4. Wählen Sie die Verteilung aus, die Sie kopieren möchten.

   Wenn Sie sich nicht sicher sind, welche Aufhebung Sie kopieren möchten, können Sie eine Aufhebung und anschließend Details anzeigen auswählen, um detaillierte Informationen zu dieser Aufhebung anzuzeigen.
5. Wählen Sie In neue kopieren aus.

6. Aktualisieren Sie die Liste der Aufhebungspfade, sofern zutreffend.
7. Wählen Sie Gültigkeitsaufhebung erstellen aus.

Stornieren von Aufhebungen

Wenn Sie eine Invalidierungsanforderung an senden CloudFront, CloudFront leitet die Anforderung innerhalb weniger Sekunden an alle Edge-Standorte weiter, und jeder Edge-Standort beginnt sofort mit der Verarbeitung der Invalidierung. Daher gibt es keine Möglichkeit, eine Aufhebung nach der Übermittlung zu stornieren.

Auflisten von Aufhebungen

Mithilfe der Konsole können Sie eine Liste der letzten 100 Invalidierungen anzeigen, die Sie für eine Distribution erstellt und ausgeführt haben. CloudFront Wenn Sie eine Liste mit mehr als 100 Aufhebungen anzeigen möchten, verwenden Sie die API-Operation `ListInvalidations`. Weitere Informationen finden Sie ListInvalidations in der CloudFront Amazon-API-Referenz.

So listen Sie Aufhebungen auf

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home.
2. Wählen Sie die Verteilung aus, für die Sie eine Liste der Aufhebungen anzeigen möchten.
3. Wählen Sie die Registerkarte Invalidations aus.

> **Note**
>
> Sie können diese Aufhebungen nicht aus der Liste entfernen.

Anzeigen von Informationen zu einer Aufhebung

Sie können detaillierte Informationen zu einer Aufhebung anzeigen, einschließlich Verteilungs-ID, Aufhebungs-ID, Status der Aufhebung, Datum und Uhrzeit der Erstellung der Aufhebung sowie eine vollständige Liste der Aufhebungspfade.

So zeigen Sie Informationen zu einer Aufhebung auf

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home.
2. Wählen Sie die Verteilung mit der Aufhebung aus, zu der Sie detaillierte Informationen anzeigen möchten.
3. Wählen Sie die Registerkarte Invalidations aus.
4. Wählen Sie die entsprechende Aufhebungs-ID aus und klicken Sie dann auf Details anzeigen.

Kosten der Aufhebung der Gültigkeit von Dateien

Die ersten 1 000 pro Monat übermittelten Aufhebungspfade sind kostenlos. Alle weiteren Aufhebungspfade (über 1 000) in einem Monat sind kostenpflichtig. Ein Aufhebungspfad kann für eine einzelne Datei (z. B. `/images/logo.jpg`) oder für mehrere Dateien (z. B. `/images/*`) gelten. Ein Pfad, der den `*` Platzhalter enthält, gilt als ein Pfad, auch wenn dadurch Tausende CloudFront von Dateien ungültig werden.

Das Maximum von 1 000 kostenlosen Aufhebungspfaden pro Monat gilt für die Gesamtanzahl an Aufhebungspfaden auf allen Distributionen, die Sie mit einem AWS-Konto erstellen. Angenommen, Sie verwenden den, um drei Verteilungen AWS-Konto `john@example.com` zu erstellen, und Sie senden 600 Invalidierungspfade für jede Verteilung in einem bestimmten Monat ein (insgesamt 1.800 Invalidierungspfade). In diesem Fall AWS wird Ihnen die Differenz zwischen der Gesamtzahl der Invalidierungspfade und dem Limit von 1000 freien Mitteln in Rechnung gestellt. In diesem Beispiel AWS würde ich Ihnen 800 Invalidierungspfade in diesem Monat in Rechnung stellen.

Die Gebühr für das Übermitteln eines Aufhebungspfads ist dieselbe, unabhängig von der Anzahl der Dateien, die Sie ungültig machen: eine einzelne Datei (`/images/logo.jpg`) oder alle Dateien, die mit einer Verteilung verknüpft sind (`/*`). Da die Gebühren für Ihre Aufhebungsanforderung pro Pfad berechnet werden, wird jeder Pfad für Abrechnungszwecke auch dann einzeln gezählt, wenn Sie mehrere Pfade in einer einzigen Anforderung zusammenfassen.

Weitere Informationen zu Preisen für Ungültigerklärungen finden Sie unter Amazon CloudFront Pricing. Weitere Informationen zur Aufhebungspfaden finden Sie unter Invalidation paths.

Invalidierungspfade mit Tags werden auf die gleiche Summe von 1.000 kostenlosen Invalidierungspfaden pro Monat angerechnet. Wenn Sie beispielsweise in einem Monat 500 Pfad-Invalidierungen und 600 Tag-Invalidierungen einreichen (also insgesamt 1.100

Invalidierungspfade), werden Ihnen 100 Invalidierungspfade in Rechnung gestellt. Ein einzelnes Tag-Invalidierungselement (z. B. `#product:electronics`) zählt als ein Pfad, unabhängig davon, wie viele zwischengespeicherte Objekte es ungültig macht — genau wie eine Pfadinvalidierung mit Platzhaltern (z. B.) zählt als ein Pfad. `/*`

Weitere Hinweise zur Tag-Invalidierung finden Sie unter. Aufhebung der Gültigkeit durch Cache-Tags

Bereitstellen von komprimierten Dateien

Wenn angeforderte Objekte komprimiert werden, können Downloads schneller sein, da die Objekte kleiner sind – in einigen Fällen weniger als ein Viertel der Größe des Originals. Schnellere Downloads können zu einem schnelleren Rendern von Webseiten für Ihre Zuschauer führen, insbesondere für JavaScript und CSS-Dateien. Darüber hinaus basieren die Kosten für die CloudFront Datenübertragung auf der Gesamtmenge der bereitgestellten Daten. Die Bereitstellung komprimierter Objekte kann kostengünstiger sein als die Bereitstellung unkomprimierter Objekte.

Themen

- Konfigurieren Sie CloudFront , um Objekte zu komprimieren
- Wie funktioniert die CloudFront Komprimierung
- Bedingungen für die Komprimierung
- Dateitypen, die komprimiert werden CloudFront
- ETag-Header-Konvertierung

Konfigurieren Sie CloudFront , um Objekte zu komprimieren

Um das Komprimieren von Objekten CloudFront zu konfigurieren, aktualisieren Sie das Cache-Verhalten, das Sie für die komprimierten Objekte bereitstellen möchten.

So konfigurieren Sie CloudFront die Komprimierung von Objekten (Konsole)

1. Melden Sie sich bei der CloudFront-Konsole an.
2. Wählen Sie Ihre Distribution und dann das zu bearbeitende Verhalten aus.
3. Setzen Sie Objekte automatisch komprimieren auf Ja.
4. Verwenden Sie eine Cache-Richtlinie, um Cache-Einstellungen anzugeben, und aktivieren Sie die Komprimierungsformate Gzip und Brotli.

Hinweise

- Für die Brotli-Komprimierung müssen Sie Cache-Richtlinien verwenden. Brotli unterstützt keine Legacy-Cache-Einstellungen.
- Um die Komprimierung mithilfe CloudFormationder CloudFrontAPI zu aktivieren, setzen Sie die `EnableAcceptEncodingBrotli` Parameter `CompressEnableAcceptEncodingGzip`, auf`true`.

Informationen zur CloudFront Komprimierung von Objekten finden Sie im folgenden Abschnitt.

Wie funktioniert die CloudFront Komprimierung

1. Ein Viewer fordert ein Objekt an. Der Viewer fügt den HTTP-Header `Accept-Encoding` in die Anforderung ein. Die Header-Werte fügen `gzip`, `br` oder beides ein. Dadurch wird angezeigt, dass der Viewer komprimierte Objekte unterstützt. Wenn der Viewer sowohl Gzip als auch Brotli unterstützt, CloudFront verwendet er Brotli.

   Note

   Die Webbrowser Chrome und Firefox unterstützen die Brotli-Komprimierung nur, wenn die Anforderung über HTTPS gesendet wird. Sie unterstützen Brotli für HTTP-Anforderungen nicht.

2. Sucht am Edge-Standort im Cache CloudFront nach einer komprimierten Kopie des angeforderten Objekts.
3. Führt je nachdem, ob sich das komprimierte Objekt im Cache befindet oder CloudFront nicht, eine der folgenden Aktionen aus:
   - Wenn sich das komprimierte Objekt bereits im Cache befindet, wird das Objekt an den Viewer CloudFront gesendet und die verbleibenden Schritte übersprungen.
   - Wenn sich das komprimierte Objekt nicht im Cache befindet, CloudFront leitet die Anfrage an den Ursprung weiter.

   Note

   Wenn sich bereits eine unkomprimierte Kopie des Objekts im Cache befindet, wird sie CloudFront möglicherweise an den Betrachter gesendet, ohne die Anfrage an den

Ursprung weiterzuleiten. Dies kann beispielsweise passieren, wenn die Komprimierung CloudFront zuvor übersprungen wurde. In diesem Fall wird das unkomprimierte Objekt CloudFront zwischengespeichert und solange bereitgestellt, bis das Objekt abläuft, entfernt oder ungültig gemacht wird.

4. Wenn der Ursprung ein komprimiertes Objekt zurückgibt (wie durch den `Content-Encoding` Header in der HTTP-Antwort angegeben), CloudFront sendet er das komprimierte Objekt an den Viewer, fügt es dem Cache hinzu und überspringt die verbleibenden Schritte. CloudFront komprimiert das Objekt nicht erneut.
5. Wenn der Ursprung ein unkomprimiertes Objekt CloudFront ohne den `Content-Encoding` Header in der HTTP-Antwort zurückgibt, CloudFront wird dann bestimmt, ob das Objekt komprimiert werden kann. Weitere Informationen finden Sie unter Bedingungen für die Komprimierung.
6. Wenn das Objekt komprimiert werden kann, wird es CloudFront komprimiert, an den Viewer gesendet und anschließend dem Cache hinzugefügt.
7. Wenn es nachfolgende Viewer-Anfragen für dasselbe Objekt CloudFront gibt, wird die erste zwischengespeicherte Version zurückgegeben. Wenn ein Viewer beispielsweise ein bestimmtes zwischengespeichertes Objekt anfordert, das Gzip-Komprimierung verwendet, und der Viewer das Gzip-Format akzeptiert, geben nachfolgende Anforderungen an dasselbe Objekt immer die Gzip-Version zurück, auch wenn der Viewer sowohl Brotli als auch Gzip akzeptiert.

Einige benutzerdefinierte Ursprünge können auch Objekte komprimieren. Ihr Origin ist möglicherweise in der Lage, Objekte zu komprimieren, die CloudFront nicht komprimiert werden. Weitere Informationen finden Sie unter Dateitypen, die komprimiert werden CloudFront.

Bedingungen für die Komprimierung

Die folgende Liste enthält weitere Informationen zu Szenarien, in denen Objekte CloudFront nicht komprimiert werden.

Anforderung verwendet HTTP 1.0

Wenn eine Anfrage HTTP 1.0 CloudFront verwendet, CloudFront wird der `Accept-Encoding` Header entfernt und das Objekt in der Antwort nicht komprimiert.

`Accept-Encoding`-Header der Anforderung

Wenn der `Accept-Encoding` Header in der Viewer-Anfrage fehlt oder wenn er keinen Wert enthält `gzip` oder `br` keinen Wert enthält, CloudFront wird das Objekt in der Antwort nicht komprimiert. Wenn der `Accept-Encoding` Header zusätzliche Werte enthält`deflate`, z. B. CloudFront werden diese entfernt, bevor die Anfrage an den Ursprung weitergeleitet wird.

Wenn für die Komprimierung von Objekten konfiguriert CloudFront ist, nimmt er den `Accept-Encoding` Header automatisch in den Cache-Schlüssel und in Ursprungsanfragen auf.

Der Inhalt wird bereits zwischengespeichert, wenn Sie das Komprimieren von CloudFront Objekten konfigurieren

CloudFront komprimiert Objekte, wenn sie vom Ursprung abgerufen werden. Wenn Sie CloudFront das Komprimieren von Objekten konfigurieren, werden Objekte, die bereits an Kantenpositionen zwischengespeichert sind, CloudFront nicht komprimiert. Wenn ein zwischengespeichertes Objekt an einem Edge-Standort abläuft und eine weitere Anforderung für das Objekt an Ihren Ursprung CloudFront weiterleitet, wird das Objekt außerdem CloudFront nicht komprimiert, wenn Ihr Ursprung den HTTP-Statuscode 304 zurückgibt. Das bedeutet, dass Edge-Standort bereits über die neueste Version des Objekts verfügt. Wenn Sie Objekte komprimieren CloudFront möchten, die bereits an Edge-Standorten zwischengespeichert sind, müssen Sie diese Objekte für ungültig erklären. Weitere Informationen finden Sie unter Aufheben der Gültigkeit von Dateien zum Entfernen von Inhalten.

Ursprung ist bereits für die Kompression von Objekten konfiguriert

Wenn Sie CloudFront die Konfiguration so konfigurieren, dass Objekte komprimiert werden und der Ursprung auch Objekte komprimiert, sollte der Ursprung einen Header enthalten. `Content-Encoding` Dieser Header weist darauf hin CloudFront , dass das Objekt bereits komprimiert ist. Wenn eine Antwort von einem Ursprung den `Content-Encoding` Header enthält, wird das Objekt CloudFront nicht komprimiert, unabhängig vom Wert des Headers. CloudFrontsendet die Antwort an den Betrachter und speichert das Objekt an der Edge-Position im Cache.

Dateitypen, die CloudFront komprimiert werden

Eine vollständige Liste finden Sie hier: Dateitypen, die komprimiert werden CloudFront.

Größe der Objekte, die CloudFront komprimiert werden

CloudFront komprimiert Objekte mit einer Größe zwischen 1.000 Byte und 10.000.000 Byte.

`Content-Length`-Header

Der Ursprung muss in der Antwort einen `Content-Length` Header enthalten, CloudFront anhand dessen bestimmt wird, ob die Größe des Objekts in dem Bereich liegt, der komprimiert wird. CloudFront Wenn der `Content-Length` Header fehlt, einen ungültigen Wert enthält oder einen Wert außerhalb des Größenbereichs für die CloudFront Komprimierung enthält, wird CloudFront das Objekt nicht komprimiert. Weitere Informationen zur Verarbeitung großer Objekte, die den Größenbereich überschreiten können, finden Sie unterWie CloudFront verarbeitet Teilanfragen für ein Objekt (Bereichs-GETs). CloudFront

Den HTTP-Statuscode der Antwort.

CloudFront komprimiert Objekte nur, wenn der HTTP-Statuscode der Antwort `200`403, oder `404` lautet.

Antwort hat keinen Körper

Wenn die HTTP-Antwort vom Ursprung keinen Hauptteil hat, gibt es nichts, was komprimiert werden CloudFront könnte.

`ETag`-Header

CloudFront ändert manchmal den `ETag` Header in der HTTP-Antwort, wenn Objekte komprimiert werden. Weitere Informationen finden Sie unter the section called “`ETag`-Header-Konvertierung”.

CloudFront überspringt die Komprimierung

CloudFront komprimiert Objekte nach bestem Wissen. In seltenen Fällen CloudFront überspringt die Komprimierung eines Objekts bei hoher CloudFront Verkehrsbelastung. CloudFront trifft diese Entscheidung auf der Grundlage einer Vielzahl von Faktoren, einschließlich der Hostkapazität. Wenn die Komprimierung für ein Objekt CloudFront übersprungen wird, wird das unkomprimierte Objekt zwischengespeichert und Benutzern weiterhin zur Verfügung gestellt, bis das Objekt abläuft, entfernt oder ungültig gemacht wird.

Dateitypen, die komprimiert werden CloudFront

Wenn Sie CloudFront das Komprimieren von Objekten konfigurieren, werden CloudFront nur Objekte komprimiert, die einen der folgenden Werte im `Content-Type` Antwortheader haben:

- `application/dash+xml`
- `application/eot`

- application/font
- application/font-sfnt
- application/javascript
- application/json
- application/opentype
- application/otf
- application/pdf
- application/pkcs7-mime
- application/protobuf
- application/rss+xml
- application/truetype
- application/ttf
- application/vnd.apple.mpegurl
- application/vnd.mapbox-vector-tile
- application/vnd.ms-fontobject
- application/wasm
- application/xhtml+xml
- application/xml
- application/x-font-opentype
- application/x-font-truetype
- application/x-font-ttf
- application/x-httpd-cgi
- application/x-javascript
- application/x-mpegurl
- application/x-opentype
- application/x-otf
- application/x-perl
- application/x-ttf
- font/eot

- font/opentype
- font/otf
- font/ttf
- image/svg+xml
- text/css
- text/csv
- text/html
- text/javascript
- text/js
- text/plain
- text/richtext
- text/tab-separated-values
- text/xml
- text/x-component
- text/x-java-source
- text/x-script
- vnd.apple.mpegurl

ETag-Header-Konvertierung

Wenn das unkomprimierte Objekt aus dem Ursprung einen gültigen, starken ETag HTTP-Header enthält und das Objekt CloudFront komprimiert, wandelt es CloudFront auch den starken ETag Header-Wert in einen schwachen um und gibt den schwachen ETag ETag Wert an den Viewer zurück. Viewer können den schwachen ETag-Wert speichern und ihn verwenden, um bedingte Anforderungen mit dem HTTP-Header If-None-Match zu senden. Auf diese Weise können Betrachter und der Ursprung die komprimierten und unkomprimierten Versionen eines Objekts als semantisch gleichwertig behandeln, wodurch unnötige Datenübertragungen vermieden werden. CloudFront

Ein gültiger, starker ETag-Headerwert beginnt und endet mit einem doppelten Anführungszeichen (''). Um den starken ETag-Wert in einen schwachen Wert zu konvertieren, fügt CloudFront die Zeichen W/ am Anfang des starken ETag-Werts hinzu.

Wenn das Objekt aus dem Ursprung einen schwachen `ETag` Header-Wert (einen Wert, der mit den Zeichen beginnt`W/`) enthält, CloudFront wird dieser Wert nicht geändert und er wird so an den Betrachter zurückgegeben, wie er vom Ursprung empfangen wurde.

Wenn das Objekt aus dem Ursprung einen ungültigen `ETag` Header-Wert enthält (der Wert beginnt nicht mit '' oder mit`W/`), CloudFront wird der `ETag` Header entfernt und das Objekt ohne den `ETag` Antwort-Header an den Viewer zurückgegeben.

Weitere Informationen finden Sie auf den folgenden Seiten in den MDN-Webdokumenten:

- Direktiven (`ETag`-HTTP-Header)
- Schwache Validierung (bedingte HTTP-Anforderungen)
- If-None-Match-HTTP-Header

Verwenden Sie AWS WAF Schutzmaßnahmen

Sie können es verwenden AWS WAF, um Ihre CloudFront Distributionen und Ursprungsserver zu schützen. AWS WAF ist eine Webanwendungs-Firewall, die zum Schutz Ihrer Webanwendungen und APIs beiträgt, indem sie Anfragen blockiert, bevor sie Ihre Server erreichen. Weitere Informationen finden Sie unter Beschleunigen und schützen Sie Ihre Websites mithilfe CloudFront AWS WAF und Richtlinien für die Implementierung AWS WAF.

Um den AWS WAF Schutz zu aktivieren, können Sie:

- Verwenden Sie den One-Choose-Schutz in der CloudFront Konsole. One-choose Protection erstellt eine AWS WAF Web Access Control List (Web ACL), konfiguriert Regeln zum Schutz Ihrer Server vor gängigen Internet-Bedrohungen und hängt die Web-ACL für Sie an die CloudFront Distribution an. Bei den Themen in diesem Abschnitt wird von der Verwendung von Schutzmaßnahmen ausgegangen, die nur einmal ausgewählt werden können.
- Verwenden Sie eine vorkonfigurierte Web-ACL (Zugriffskontrollliste), die Sie in der AWS WAF Konsole oder mithilfe der APIs erstellen. AWS WAF Weitere Informationen finden Sie unter Web Access Control Lists (ACLs) im AWS WAF Developer Guide und AssociateWeb ACL in der API-Referenz AWS WAF

Sie können aktivieren, AWS WAF wenn Sie:

- Eine Distribution erstellen
- Verwenden Sie das Sicherheits-Dashboard, um die Sicherheitseinstellungen einer vorhandenen Distribution zu bearbeiten.

Wenn Sie den One-Chose-Schutz verwenden, AWS wird ein empfohlener Satz von Schutzmaßnahmen CloudFront angewendet, der:

- Blockieren von IP-Adressen auf der Grundlage interner Bedrohungsinformationen von Amazon zu potenziellen Bedrohungen.
- Schutz vor den häufigsten Sicherheitslücken in Webanwendungen, wie in den OWASP Top 10 beschrieben.
- Schutz vor böswilligen Akteuren, die Anwendungsschwachstellen entdecken.

> **Important**
>
> Sie müssen AWS WAF diese Option aktivieren, wenn Sie Sicherheitsmetriken im Sicherheits-Dashboard anzeigen möchten. CloudFront Ohne AWS WAF diese Option können Sie das Sicherheits-Dashboard nur verwenden, um CloudFront geografische Einschränkungen zu aktivieren AWS WAF oder zu konfigurieren. Weitere Informationen zum Dashboard finden Sie unter Verwalten AWS WAF Sicherheitsvorkehrungen im CloudFront Sicherheits-Dashboard in diesem Abschnitt.

Themen

- Aktivieren AWS WAF für Distributionen
- Verwalten AWS WAF Sicherheitsvorkehrungen im CloudFront Sicherheits-Dashboard
- Festlegen der Ratenbegrenzung
- Deaktivieren AWS WAF Sicherheitsvorkehrungen

Aktivieren AWS WAF für Distributionen

Sie können sie aktivieren, AWS WAF wenn Sie eine Verteilung erstellen, oder Sie können den Sicherheitsschutz für eine vorhandene Zugriffskontrollliste (ACL) aktivieren.

Wenn Sie diese Option AWS WAF für Ihre CloudFront Distribution aktivieren, können Sie auch die Bot-Steuerung aktivieren und den Sicherheitsschutz nach Bot-Kategorien konfigurieren.

Themen

- Aktivieren AWS WAF für eine neue Distribution
- Verwenden einer vorhandenen Web-ACL
- Aktivieren von Bot Control
- Konfigurieren des Schutzes nach Bot-Kategorie

Aktivieren AWS WAF für eine neue Distribution

Das folgende Verfahren zeigt Ihnen, wie Sie die Option aktivieren AWS WAF , wenn Sie eine neue CloudFront Distribution erstellen.

Um zu aktivieren AWS WAF für eine neue Distribution

1. Öffnen Sie die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home.
2. Wählen Sie im Navigationsbereich Distributionen und dann Distribution erstellen aus.
3. Folgen Sie bei Bedarf den Schritten unter Eine Distribution erstellen.
4. Wählen Sie im Abschnitt Web Application Firewall die Option Bearbeiten und dann Sicherheitsschutzmaßnahmen aktivieren aus.
5. Füllen Sie die folgenden Felder aus:

   - Überwachungsmodus verwenden – Sie können den Überwachungsmodus aktivieren, wenn Sie zunächst Daten sammeln möchten, um zu testen, wie der Schutz funktioniert. Wenn der Überwachungsmodus aktiviert ist, werden Anfragen nicht blockiert, wenn der Schutz aktiv ist. Stattdessen sammelt der Überwachungsmodus Daten über Anforderungen, die bei aktiviertem Schutz blockiert werden würden. Wenn Sie bereit sind, mit der Blockierung zu beginnen, können Sie sie auf der Seite Sicherheit aktivieren.
   - Zusätzliche Schutzmaßnahmen – Wählen Sie die Optionen aus, die Sie aktivieren möchten. Wenn Sie die Ratenbegrenzung aktivieren, finden Sie weitere Informationen unter the section called “Festlegen der Ratenbegrenzung”.
   - Preisschätzung — Sie können den Abschnitt öffnen, um ein Feld anzuzeigen, in das Sie eine andere Zahl eingeben können, requests/month und eine neue Schätzung wird angezeigt.
6. Überprüfen Sie die verbleibenden Distributionseinstellungen und wählen Sie dann Distribution erstellen aus.

Nachdem Sie eine Verteilung erstellt haben, CloudFront wird ein Sicherheits-Dashboard erstellt. Sie können dieses Dashboard verwenden, um AWS WAF zu deaktivieren oder zu aktivieren. Wenn Sie es AWS WAF noch nicht aktiviert haben, bleiben die Diagramme und Grafiken im Dashboard leer.

Verwenden einer vorhandenen Web-ACL

Wenn Sie bereits eine Web-ACL haben, können Sie diese anstelle des von angebotenen Schutzes verwenden AWS WAF.

Um eine bestehende zu verwenden AWS WAF configuration

1. Öffnen Sie die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home.
2. Führen Sie eine der folgenden Aktionen aus:

a. Wählen Sie Distribution erstellen aus und folgen Sie den Schritten unter Eine Distribution erstellen. Kehren Sie dann zu diesem Thema zurück.

b. Wählen Sie eine bestehende Konfiguration und anschließend die Registerkarte Sicherheit aus.

3. Wählen Sie im Abschnitt Web Application Firewall (WAF) die Option Bearbeiten und dann Sicherheitsschutzmaßnahmen aktivieren aus.
4. Wählen Sie Bestehende WAF-Konfiguration verwenden. Diese Option wird nur angezeigt, wenn Sie Web-ACLs konfiguriert haben.
5. Wählen Sie Ihre vorhandene Web-ACL aus der Tabelle Wählen Sie eine Web-ACL aus.
6. Überprüfen Sie die übrigen Distributionseinstellungen und wählen Sie dann Distribution erstellen aus.

Aktivieren von Bot Control

Wenn Sie AWS WAF die Option für Ihre CloudFront Distribution aktivieren, können Sie Bot-Anfragen für einen bestimmten Zeitraum im Sicherheits-Dashboard in der CloudFront Konsole einsehen. Sie können hier auch Bot Control aktivieren oder deaktivieren.

Es fallen Gebühren an, wenn Sie Bot Control aktivieren. Das Sicherheitsdashboard bietet einen Kostenvoranschlag.

Wenn Sie Bot Control aktivieren, zeigt das Sicherheitsdashboard den Bot-Datenverkehr nach den einzelnen Bot-Typen und Bot-Kategorien an. Wenn Sie Bot Control deaktivieren, wird der Bot-Datenverkehr basierend auf dem Anforderungs-Sampling angezeigt.

So aktivieren Sie Bot Control:

1. Öffnen Sie die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home.
2. Wählen Sie im Navigationsbereich die Option Distributionen und dann die Distribution, die Sie ändern möchten.
3. Wählen Sie die Registerkarte Sicherheit aus.
4. Scrollen Sie nach unten zum Abschnitt Bot-Anfragen für einen bestimmten Zeitraum und wählen Sie Bot Control aktivieren aus.
5. Aktivieren Sie im Dialogfeld Bot Control unter Konfiguration das Kontrollkästchen Bot Control für gängige Bots aktivieren.

6. Wählen Sie Änderungen speichern aus.

Konfigurieren des Schutzes nach Bot-Kategorie

Wenn Sie Bot Control aktivieren, können Sie konfigurieren, wie jeder nicht verifizierte Bot je nach Bot-Kategorie behandelt wird. Sie können beispielsweise einen HTTP-Bibliotheks-Bot in den Überwachungsmodus versetzen und einem Link Checker eine Challenge zuweisen.

> **Note**
>
> Bots, von denen bekannt ist, dass AWS sie häufig und überprüfbar sind, wie z. B. bekannte Suchmaschinen-Crawler, unterliegen nicht den hier festgelegten Aktionen. Die Bot Control bestätigt, dass validierte Bots aus der Quelle stammen, aus der sie zu stammen vorgeben, bevor sie als verifiziert markiert werden.

So konfigurieren Sie den Schutz für eine Bot-Kategorie

1. Öffnen Sie die CloudFront Konsole unter. https://console.aws.amazon.com/cloudfront/v4/home
2. Wählen Sie im Navigationsbereich die Option Distributionen und dann die Distribution, die Sie ändern möchten.
3. Wählen Sie die Registerkarte Sicherheit aus.
4. Zeigen Sie im Diagramm Anforderungen nach Bot-Kategorie auf eines der Elemente in der Spalte Nicht verifizierte Bot-Aktion und wählen Sie das Stiftsymbol zur Bearbeitung aus.
5. Öffnen Sie die Ergebnisliste und wählen Sie eine der nachstehenden Optionen aus:
   - Blockieren
   - Sobald Sie die Details auf dieser Seite überprüft haben, klicken Sie auf
   - Überwachungsmodus
   - CAPTCHA
   - Challenge
6. Wählen Sie das Häkchen neben der Liste aus, um Ihre Änderung zu speichern.

Verwalten AWS WAF Sicherheitsvorkehrungen im CloudFront Sicherheits-Dashboard

CloudFront erstellt ein Sicherheits-Dashboard für jede Ihrer Distributionen. Sie verwenden die Dashboards in der CloudFront Konsole. Mithilfe der Dashboards können Sie die allgemeinen Sicherheitsvorkehrungen für Ihre Webanwendungen AWS WAF gemeinsam an einem zentralen Ort überwachen und verwalten. CloudFront Die Dashboards enthalten die folgenden Aufgaben und Daten:

- Sicherheitskonfiguration — Sie können Schutzmaßnahmen aktivieren und deaktivieren und sich alle anwendungsspezifischen AWS WAF Schutzmaßnahmen wie Schutzmaßnahmen anzeigen lassen. WordPress
- Sicherheitstrends – Dazu zählen zulässige und blockierte Anforderungen, Challenge- und CAPTCHA-Anforderungen sowie die häufigsten Angriffsarten. Sie können die Traffic-Zahlen und ihre Veränderung im Laufe der Zeit sehen. Wenn beispielsweise alle Anfragen um 3 % zunehmen, die zulässigen Anfragen jedoch um 14 % steigen, bedeutet das, dass Sie im aktuellen Zeitraum einen größeren Teil Ihres Traffics zugelassen haben.
- Bot-Anforderungen – Sie können sehen, wie viel Datenverkehr von Bots stammt, um welche Arten von Bots (verifiziert oder nicht verifiziert) es sich dabei handelt und wie sich die prozentualen Zuweisungen der Bot-Typen (verifiziert oder nicht verifiziert) im Laufe der Zeit ändern. Weitere Informationen zum Aktivieren von Bot Control finden Sie unter Aktivieren von Bot Control.
- Anforderungsprotokolle – Protokolldaten können dabei helfen, Fragen zu Sicherheitstrends oder Bot-Anforderungen zu beantworten. Sie können Ihre Protokolle durchsuchen, ohne Abfragen schreiben zu müssen, und anhand von Aggregationsdiagrammen feststellen, ob ein gefilterter Satz von Protokollen hauptsächlich auf einer Teilmenge von HTTP-Methoden, IP-Adressen, URI-Pfaden oder Ländern basiert. Sie können den Mauszeiger über Werte in den Diagrammen bewegen und IP-Adressen und Länder blockieren. Weitere Informationen finden Sie unter Aktivieren AWS WAF Protokolle.
- Verwaltung geografischer Beschränkungen — CloudFront und AWS WAF Bereitstellung von Funktionen für geografische Beschränkungen. CloudFront stellt geografische Einschränkungen kostenlos zur Verfügung, aber Metriken für CloudFront geografische Einschränkungen werden nicht im Sicherheits-Dashboard angezeigt. Um die Anforderungsmetriken für Anfragen aus geblockten Ländern zu sehen, müssen Sie AWS WAF geografische Einschränkungen verwenden. Bewegen Sie dazu den Mauszeiger über eine Länderleiste im Sicherheits-Dashboard und

blockieren Sie das Land. Weitere Informationen finden Sie unter Verwenden Sie CloudFront geografische Einschränkungen.

- Die Option Blockieren ist möglicherweise nicht verfügbar, wenn Sie zuvor außerhalb der CloudFront Konsole eine benutzerdefinierte AWS WAF Regel zum Blockieren von Ländern erstellt haben.

Themen

- Voraussetzungen
- Aktivieren AWS WAF Protokolle

Voraussetzungen

Sie müssen AWS WAF diese Option aktivieren, wenn Sie Sicherheitsmetriken im CloudFront Sicherheits-Dashboard anzeigen möchten. Wenn Sie diese Option nicht aktivieren AWS WAF, können Sie das Sicherheits-Dashboard nur verwenden, um CloudFront geografische Einschränkungen zu aktivieren AWS WAF oder zu konfigurieren.

Weitere Informationen zur Aktivierung finden AWS WAF Sie unterAktivieren AWS WAF für Distributionen.

Aktivieren AWS WAF Protokolle

AWS WAF Protokolldaten können Ihnen helfen, bestimmte Verkehrsmuster zu isolieren. Protokolle können Ihnen beispielsweise zeigen, woher bestimmter Traffic stammt oder was er bewirkt.

Wenn Sie die AWS WAF Protokollierung für aktivieren CloudWatch, fragt das CloudFront Security Dashboard die Erkenntnisse aus den CloudWatch Protokollen ab, aggregiert sie und zeigt sie an. Wir berechnen keine Gebühren für die Nutzung des Sicherheits-Dashboards, aber die CloudWatch Preise gelten für Protokolle, die über das Dashboard abgefragt werden. Weitere Informationen finden Sie unter CloudWatch Amazon-Preise.

So aktivieren Sie die Protokollierung:

1. Geben Sie Ihr erwartetes Anforderungsvolumen in das requests/month Feld Anzahl an, um die Kosten für die Aktivierung von Protokollen abzuschätzen.
2. Wählen Sie das Kontrollkästchen AWS WAF Protokolle aktivieren aus.
3. Wählen Sie Enable (Aktivieren) aus.

CloudFront erstellt eine CloudWatch Protokollgruppe und aktualisiert Ihre AWS WAF Konfiguration, um mit der Protokollierung zu beginnen CloudWatch. Bei der erstmaligen Verwendung kann es einige Minuten dauern, bis die Protokolldaten angezeigt werden. Im Abschnitt Anfragen der Diagrammliste sind alle Anfragen aufgeführt. In den Balkendiagrammen unter den einzelnen Anfragen werden die Daten nach HTTP-Methode, den wichtigsten URI-Pfaden, den wichtigsten IP-Adressen und den wichtigsten Ländern zusammengefasst. Die Diagramme können Ihnen dabei helfen, Muster zu finden. Beispielsweise können Sie eine unverhältnismäßig hohe Anzahl an Anfragen von einer einzigen IP-Adresse oder von Daten aus einem Land sehen, die Sie zuvor nicht in Ihren Protokollen gesehen hatten. Sie können Anfragen nach Land, Host-Header und anderen Attributen filtern, um unerwünschten Datenverkehr zu finden. Nachdem Sie den Traffic identifiziert haben, bewegen Sie den Mauszeiger über eine einzelne Anfrage oder ein Diagrammelement und blockieren Sie eine IP-Adresse oder ein Land.

> **Note**
>
> Die angezeigten Metriken basieren auf Web-ACL. Wenn Sie dieselbe Web-ACL mehreren Verteilungen zuordnen, werden Ihnen daher alle Metriken für Ihre Web-ACL angezeigt, nicht nur die AWS WAF Anfragen, die für diese Verteilung verarbeitet werden.

Festlegen der Ratenbegrenzung

Die Ratenbegrenzung ist eine der Empfehlungen, die Sie möglicherweise bei der Konfiguration von Sicherheitsschutzvorrichtungen erhalten.

CloudFront aktiviert im Monitormodus immer die Ratenbegrenzung. Wenn der Monitormodus aktiviert ist, werden Messwerte CloudFront erfasst, die Ihnen mitteilen, ob, wie oft und um wie viel die Rate, die Sie im Feld Ratenbegrenzung konfiguriert haben, überschritten wurde.

Nachdem Sie die Verteilung gespeichert haben, CloudFront beginnt die Erfassung von Daten auf der Grundlage der Zahl im Feld Ratenbegrenzung.

Sie können die Einstellungen zur Ratenbegrenzung im Abschnitt Sicherheit — Web Application Firewall (WAF) auf der Registerkarte Sicherheit jeder CloudFront Distribution aktivieren oder verwalten.

> **Note**
>
> Die Option zur Ratenbegrenzung wird in der CloudFront Konsole nur angezeigt, wenn Sie für Ihre Distribution einen benutzerdefinierten Ursprung angegeben haben, der nicht von S3 stammt. Andernfalls werden Ihnen nur die für die Distribution aktivierten Kernschutzmaßnahmen angezeigt. Weitere Informationen über Ursprungstypen erhalten Sie unter Verwenden Sie bei Verteilungen verschiedene Ursprünge CloudFront.

So legen Sie die Ratenbegrenzung fest

1. Öffnen Sie die CloudFront Konsole unter. https://console.aws.amazon.com/cloudfront/v4/home
2. Wählen Sie im Navigationsbereich die Option Distributionen und dann die Distribution, die Sie ändern möchten.
3. Wählen Sie die Registerkarte Sicherheit aus.
4. Wählen Sie im Abschnitt Sicherheit – Web Application Firewall (WAF) die Option Bearbeiten aus.
5. Wählen Sie unter Zusätzliche Schutzmaßnahmen die Option Ratenbegrenzung aus. Sie können die Ratenbegrenzung wahlweise ändern. Wenn Sie die Rate angepasst haben, wählen Sie Änderungen speichern aus.
6. Im Abschnitt Sicherheit — Web Application Firewall (WAF) können Sie neben Ratenbegrenzung den Monitormodus und dann Blockierung aktivieren auswählen, um den Monitormodus zu deaktivieren. CloudFront beginnt, Anfragen zu blockieren, die das angegebene Ratenlimit überschreiten.

Weitere Informationen zur Aktivierung AWS WAF und Ratenbegrenzung finden Sie im Blogbeitrag Introducing CloudFront Security Dashboard, einem Blogbeitrag zu Unified CDN und Security Experience.

Deaktivieren AWS WAF Sicherheitsvorkehrungen

Wenn Ihre Distribution keine AWS WAF Sicherheitsvorkehrungen benötigt, können Sie diese Funktion mithilfe der CloudFront Konsole deaktivieren.

Wenn Sie zuvor den AWS WAF Schutz aktiviert und keine vorhandene WAF-Konfiguration ausgewählt haben (auch bekannt als One-Chose-Schutz), wurde CloudFront automatisch eine

Web-ACL für Sie erstellt. Bei Web-ACLs, die auf diese Weise erstellt wurden, trennt die CloudFront Konsole die Zuordnung der Ressource und löscht die Web-ACL.

Das Trennen einer Web-ACL ist nicht dasselbe wie das Löschen. Durch das Trennen der Zuordnung wird die Web-ACL aus Ihrer Distribution entfernt, jedoch nicht aus Ihrem AWS-Konto gelöscht. Weitere Informationen finden Sie unter Zuordnen oder Aufheben der Zuordnung einer Web-ACL zu einer AWS Ressource im AWS WAF, AWS Firewall Manager und Developer Guide. AWS Shield Advanced

Gehen Sie wie folgt vor, um AWS WAF Schutzmaßnahmen zu deaktivieren und die Web-ACL von Ihrer Distribution zu trennen.

Um es zu deaktivieren AWS WAF Sicherheitsvorkehrungen in CloudFront

1. Öffnen Sie die CloudFront Konsole unter. https://console.aws.amazon.com/cloudfront/v4/home
2. Wählen Sie im Navigationsbereich die Option Distributionen und dann die Distribution, die Sie ändern möchten.
3. Wählen Sie die Registerkarte Sicherheit und dann Bearbeiten aus.
4. Wählen Sie im Abschnitt Web Application Firewall (WAF) die Option AWS WAF Schutz deaktivieren aus.
5. Wählen Sie Änderungen speichern aus.

Hinweise

- Wenn Sie den AWS WAF Sicherheitsschutz deaktiviert haben und die Web-ACL trotzdem von Ihrem löschen möchten AWS-Konto, können Sie sie manuell löschen. Gehen Sie wie folgt vor, um eine Web-ACL zu löschen. In der AWS WAF & Shield-Konsole müssen Sie auf der Seite Web-ACLs die Liste Global (CloudFront) auswählen, um die Web-ACLs zu finden.
- Wenn Sie eine Distribution von der CloudFront Konsole löschen, CloudFront wird versucht, auch die Web-ACL zu löschen, sofern Sie den One-Chose-Schutz gewählt haben. Dies ist das optimale Szenario und kann nicht garantiert werden. Weitere Informationen finden Sie unter Löschen einer -Verteilung.

Konfigurieren des sicheren Zugriffs und Beschränken des Zugriffs auf Inhalte

CloudFront bietet mehrere Optionen zum Schutz der bereitgestellten Inhalte. Im Folgenden finden Sie einige Methoden, mit denen Sie CloudFront den Zugriff auf Inhalte sichern und einschränken können:

- Konfigurieren von HTTPS-Verbindungen.
- Verhindern, dass Benutzer an bestimmten geografischen Standorten auf Inhalte zugreifen
- Erfordern Sie, dass Benutzer mithilfe CloudFront signierter URLs oder signierter Cookies auf Inhalte zugreifen
- Einrichten der Verschlüsselung auf Feldebene für bestimmte Inhaltsfelder
- Verwenden Sie AWS WAF , um den Zugriff auf Ihre Inhalte zu kontrollieren

Sie sollten auch eine DDoS-resilient Architektur für Ihre Infrastruktur und Anwendungen implementieren. Weitere Informationen finden Sie unter AWS – Best Practices für DDoS-Resilienz.

Weitere Informationen finden Sie unter:

- Sichern Sie Ihre Inhaltsbereitstellung mit CloudFront
- SIEM auf Amazon Service OpenSearch

Themen

- Verwenden Sie HTTPS mit CloudFront
- Verwenden von alternativen Domainnamen und HTTPS
- Gegenseitige TLS-Authentifizierung mit CloudFront (Viewer mTLS)
- Origin Mutual TLS mit CloudFront
- Bereitstellen privater Inhalte mit signierten URLs und signierten Cookies
- Beschränken Sie den Zugriff auf ein AWS Ursprungsserver
- Beschränken des Zugriffs auf Application Load Balancer
- Einschränken der geografischen Distribution Ihrer Inhalte
- Vertrauliche Daten durch Verschlüsselung auf Feldebene schützen

Verwenden Sie HTTPS mit CloudFront

Sie können so konfigurieren CloudFront , dass Zuschauer HTTPS verwenden müssen, sodass Verbindungen bei der CloudFront Kommunikation mit Zuschauern verschlüsselt werden. Sie können auch so konfigurieren CloudFront , dass HTTPS mit Ihrem Absender verwendet wird, sodass Verbindungen bei der CloudFront Kommunikation mit Ihrem Absender verschlüsselt werden.

Wenn du so konfigurierst CloudFront , dass HTTPS sowohl für die Kommunikation mit Zuschauern als auch für die Kommunikation mit deinem Absender erforderlich ist, passiert Folgendes, wenn du CloudFront eine Anfrage erhältst:

1. Ein Zuschauer sendet eine HTTPS-Anfrage an CloudFront. Hier gibt es einige SSL/TLS Verhandlungen zwischen dem Zuschauer und CloudFront. Am Ende sendet der Viewer die Anfrage in einem verschlüsselten Format.
2. Wenn der CloudFront Edge-Standort eine zwischengespeicherte Antwort enthält, CloudFront verschlüsselt die Antwort und gibt sie an den Betrachter zurück, und der Betrachter entschlüsselt sie.
3. Wenn der CloudFront Edge-Standort keine zwischengespeicherte Antwort enthält, CloudFront führt er eine SSL/TLS Aushandlung mit Ihrem Ursprung durch und leitet die Anfrage nach Abschluss der Verhandlung in verschlüsselter Form an Ihren Ursprung weiter.
4. Ihr Absender entschlüsselt die Anfrage, verarbeitet sie (generiert eine Antwort), verschlüsselt die Antwort und sendet die Antwort an. CloudFront
5. CloudFront entschlüsselt die Antwort, verschlüsselt sie erneut und leitet sie an den Betrachter weiter. CloudFrontspeichert die Antwort auch am Edge-Standort im Cache, sodass sie bei der nächsten Anforderung verfügbar ist.
6. Der Viewer entschlüsselt die Antwort.

Der Vorgang funktioniert grundsätzlich auf die gleiche Weise, unabhängig davon, MediaStore ob es sich bei Ihrem Ursprung um einen Amazon S3-Bucket oder um einen benutzerdefinierten Ursprung wie einen HTTP/S Server handelt.

> **Note**
>
> Um Angriffe vom Typ SSL-Neuverhandlung abzuwehren, wird die Neuverhandlung von Zuschauer- und Ursprungsanfragen CloudFront nicht unterstützt.

Alternativ können Sie die gegenseitige Authentifizierung für Ihre Distribution aktivieren. CloudFront Weitere Informationen finden Sie unter Gegenseitige TLS-Authentifizierung mit CloudFront (Viewer mTLS).

Informationen dazu, wie Sie HTTPS zwischen Zuschauern und CloudFront Ihrem Absender sowie zwischen CloudFront und Ihrem Absender verlangen, finden Sie in den folgenden Themen.

Themen

- Erfordern Sie HTTPS für die Kommunikation zwischen Zuschauern und CloudFront
- Erfordern Sie HTTPS für die Kommunikation zwischen CloudFront und Ihrem benutzerdefinierten Ursprung
- Erfordern Sie HTTPS für die Kommunikation zwischen CloudFront und Ihrem Amazon S3-Ursprung
- Unterstützte Protokolle und Chiffren zwischen Viewern und CloudFront
- Unterstützte Protokolle und Chiffren zwischen CloudFront und dem Ursprung

Erfordern Sie HTTPS für die Kommunikation zwischen Zuschauern und CloudFront

Sie können ein oder mehrere Cache-Verhaltensweisen in Ihrer CloudFront Distribution so konfigurieren, dass HTTPS für die Kommunikation zwischen Zuschauern und erforderlich ist CloudFront. Sie können auch ein oder mehrere Cache-Verhalten so konfigurieren, dass sowohl HTTP als auch HTTPS zulässig sind, sodass HTTPS für einige Objekte CloudFront erforderlich ist, für andere jedoch nicht. Die Konfigurationsschritte variieren je nachdem, welchen Domänennamen Sie in Objekt-URLs verwenden:

- Wenn Sie den Domänennamen verwenden, der Ihrer Distribution CloudFront zugewiesen wurde, z. B. d111111abcdef8.cloudfront.net, ändern Sie die Viewer-Protokollrichtlinieneinstellung für ein oder mehrere Cacheverhalten so, dass HTTPS-Kommunikation erforderlich ist. CloudFront Stellt in dieser Konfiguration das Zertifikat bereit. SSL/TLS

  Informationen zum Ändern des Werts von Viewer Protocol Policy mithilfe der CloudFront Konsole finden Sie weiter unten in diesem Abschnitt.

  Informationen darüber, wie Sie die CloudFront API verwenden, um den Wert des `ViewerProtocolPolicy` Elements zu ändern, finden Sie UpdateDistribution in der CloudFront Amazon-API-Referenz.

- Wenn Sie einen eigenen Domänennamen, z. B. beispiel.com, verwenden, müssen Sie mehrere CloudFront-Einstellungen ändern. Sie müssen außerdem ein von AWS Certificate Manager (ACM) bereitgestelltes SSL/TLS Zertifikat verwenden oder ein Zertifikat von einer Drittanbieter-Zertifizierungsstelle in ACM oder den IAM-Zertifikatsspeicher importieren. Weitere Informationen finden Sie unter Verwenden von alternativen Domainnamen und HTTPS.

> **Note**
>
> Wenn Sie sicherstellen möchten, dass die Objekte, von denen die Zuschauer erhalten, verschlüsselt CloudFront wurden, als CloudFront sie von Ihrem Ursprung abgerufen wurden, verwenden Sie immer HTTPS zwischen CloudFront und Ihrem Ursprung. Wenn Sie kürzlich zwischen CloudFront und Ihrem Ursprung von HTTP zu HTTPS gewechselt haben, empfehlen wir Ihnen, Objekte an CloudFront Edge-Standorten für ungültig zu erklären. CloudFront gibt ein Objekt an einen Viewer zurück, unabhängig davon, ob das vom Viewer verwendete Protokoll (HTTP oder HTTPS) mit dem Protokoll übereinstimmt, CloudFront mit dem das Objekt abgerufen wurde. Weitere Informationen zum Entfernen oder Ersetzen von Objekten in einer Verteilung finden Sie unter Inhalte hinzufügen, entfernen oder ersetzen, die CloudFront verbreiten.

Erzwingen von HTTPS für Viewer

Gehen Sie wie folgt vor, um HTTPS zwischen Viewern und CloudFront für ein oder mehrere Cache-Verhaltensweisen vorzuschreiben.

Um CloudFront zu konfigurieren, dass HTTPS zwischen Viewern und erforderlich ist CloudFront

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home.
2. Wählen Sie im oberen Bereich der CloudFront Konsole die ID für die Distribution aus, die Sie aktualisieren möchten.
3. Wählen Sie auf der Registerkarte Verhaltensweisen das zu aktualisierende Cacheverhalten und anschließend Bearbeiten aus.
4. Geben Sie einen der folgenden Werte für Viewer-Protokollrichtlinie an:

Redirect HTTP to HTTPS

Viewer können beide Protokolle verwenden. HTTP `GET` und `HEAD` Anfragen werden automatisch zu HTTPS-Anfragen umgeleitet. CloudFront gibt den HTTP-Statuscode 301 (Moved Permanently) zusammen mit der neuen HTTPS-URL zurück. Der Viewer sendet die Anfrage dann erneut CloudFront unter Verwendung der HTTPS-URL.

> **Important**
>
> Wenn Sie`POST`,, `PUT` `DELETE``OPTIONS`, oder `PATCH` über HTTP mit einem HTTP-zu-HTTPS-Cache-Verhalten und einer Anforderungsprotokollversion von HTTP 1.1 oder höher senden, CloudFront leitet die Anfrage an eine HTTPS-Stelle mit dem HTTP-Statuscode 307 weiter (Temporäre Weiterleitung). Dies gewährleistet, dass die Anforderung erneut unter Verwendung derselben Methode und derselben Nutzdaten an den neuen Speicherort gesendet wird.
> Wenn Sie`POST`,, `PUT` `DELETE``OPTIONS`, oder `PATCH` Anfragen über HTTP an den HTTPS-Cache mit einer Version des Anforderungsprotokolls unter HTTP 1.1 senden, wird der HTTP-Statuscode 403 (Verboten) CloudFront zurückgegeben.

Wenn ein Viewer eine HTTP-Anforderung veranlasst, die an eine HTTPS-Adresse umgeleitet wird, berechnet CloudFront Gebühren für beide Anforderungen. Bei der HTTP-Anfrage fallen nur Gebühren für die Anfrage und für die Header an, die an den Viewer CloudFront zurückgegeben werden. Bezüglich der HTTPS-Anforderung fallen Gebühren für die Anforderung sowie für die vom Ursprung zurückgegebenen Header und das Objekt an, das vom Ursprung zurückgegeben wird.

Nur HTTPS

Viewer können auf Ihre Inhalte nur zugreifen, wenn sie HTTPS verwenden. Wenn ein Viewer eine HTTP-Anfrage anstelle einer HTTPS-Anfrage sendet, CloudFront gibt er den HTTP-Statuscode 403 (Forbidden) zurück und gibt das Objekt nicht zurück.

5. Wählen Sie Änderungen speichern aus.
6. Wiederholen Sie die Schritte 3 bis 5 für jedes weitere Cacheverhalten, für das Sie HTTPS zwischen Viewern und benötigen CloudFront.
7. Vergewissern Sie sich, dass folgende Punkte erfüllt sind, bevor Sie die aktualisierte Konfiguration in einer Produktionsumgebung verwenden:

- Das Pfadmuster eines jeden Cache-Verhaltens wird nur auf die Anforderungen angewendet, für die die Viewer HTTPS verwenden sollen.
- Die Cache-Verhaltensweisen werden in der Reihenfolge aufgeführt, in der Sie sie auswerten CloudFront möchten. Weitere Informationen finden Sie unter Pfadmuster.
- Die Cache-Verhaltensweisen leiten Anforderungen an die richtigen Ursprünge weiter.

Erfordern Sie HTTPS für die Kommunikation zwischen CloudFront und Ihrem benutzerdefinierten Ursprung

Sie können HTTPS für die Kommunikation zwischen CloudFront und Ihrem Absender benötigen.

> **Note**
>
> Wenn Ihr Ursprung ein Amazon S3-Bucket ist, der als Website-Endpunkt konfiguriert ist, können Sie nicht für die Verwendung von HTTPS mit Ihrem Ursprung konfigurieren CloudFront , da Amazon S3 HTTPS für Website-Endpunkte nicht unterstützt.

Um HTTPS zwischen CloudFront und Ihrem Ursprung zu verlangen, folgen Sie den Anweisungen in diesem Thema, um Folgendes zu tun:

1. Ändern Sie in Ihrer Verteilung die Einstellung Origin Protocol Policy (Ursprungsprotokollrichtlinie) für den Ursprung
2. Installieren Sie ein SSL/TLS Zertifikat auf Ihrem Ursprungsserver (dies ist nicht erforderlich, wenn Sie einen Amazon S3-Ursprungsserver oder bestimmte andere AWS Ursprünge verwenden).

Themen

- Erzwingen von HTTPS für benutzerdefinierte Ursprünge
- Installieren Sie ein SSL/TLS Zertifikat auf Ihrem benutzerdefinierten Ursprung

Erzwingen von HTTPS für benutzerdefinierte Ursprünge

Das folgende Verfahren erklärt, wie Sie die Verwendung von HTTPS für die Kommunikation mit einem Elastic Load Balancing Load Balancer, einer Amazon EC2-Instance oder einem anderen benutzerdefinierten Ursprung konfigurieren CloudFront . Informationen zur Verwendung der

CloudFront API zur Aktualisierung einer Distribution finden Sie UpdateDistribution in der CloudFront Amazon-API-Referenz.

So konfigurieren Sie CloudFront , dass HTTPS zwischen CloudFront und Ihrem benutzerdefinierten Ursprung erforderlich ist

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home.
2. Wählen Sie im oberen Bereich der CloudFront Konsole die ID für die Distribution aus, die Sie aktualisieren möchten.
3. Wählen Sie auf der Registerkarte Ursprünge den Ursprung aus, den Sie aktualisieren möchten, und wählen Sie dann Bearbeiten aus.
4. Aktualisieren Sie die folgenden Einstellungen:

   Ursprungsprotokollrichtlinien

   Ändern Sie die Einstellung von Origin Protocol Policy für die den betroffenen Ursprung in Ihrer Verteilung:

   - Nur HTTPS — CloudFront verwendet nur HTTPS, um mit deinem benutzerdefinierten Ursprung zu kommunizieren.
   - Match Viewer — CloudFront kommuniziert mit Ihrem benutzerdefinierten Ursprung über HTTP oder HTTPS, je nach Protokoll der Viewer-Anfrage. Wenn Sie beispielsweise Match Viewer für Origin Protocol Policy wählen und der Viewer HTTPS verwendet, um ein Objekt anzufordern CloudFront, verwendet er CloudFront auch HTTPS, um die Anfrage an Ihren Ursprung weiterzuleiten.

     Wählen Sie Match Viewer nur dann, wenn Sie Redirect HTTP to HTTPS oder HTTPS Only für Viewer Protocol Policy auswählen.

     CloudFront speichert das Objekt nur einmal im Cache, auch wenn Betrachter Anfragen sowohl mit HTTP- als auch mit HTTPS-Protokollen stellen.

   Origin SSL Protocols

   Wählen Sie die SSL-Protokolle für den ausgewählten Ursprung in Ihrer Verteilung aus. Das SSLv3-Protokoll ist weniger sicher. Daher empfehlen wir, SSLv3 nur zu verwenden, wenn Ihr Ursprung TLSv1 oder eine spätere Version nicht unterstützt. Der TLSv1-Handshake ist sowohl abwärts- als auch aufwärtskompatibel mit SSLv3, später jedoch nicht. TLSv1.1 Wenn Sie SSLv3 wählen, werden nur SSLv3-Handshake-Anfragen gesendet. CloudFront

5. Wählen Sie Änderungen speichern aus.
6. Wiederholen Sie die Schritte 3 bis 5 für jeden weiteren Ursprung, für den Sie HTTPS benötigen, und für Ihren benutzerdefinierten Ursprung. CloudFront
7. Vergewissern Sie sich, dass folgende Punkte erfüllt sind, bevor Sie die aktualisierte Konfiguration in einer Produktionsumgebung verwenden:

   - Das Pfadmuster eines jeden Cache-Verhaltens wird nur auf die Anforderungen angewendet, für die die Viewer HTTPS verwenden sollen.
   - Die Cache-Verhaltensweisen werden in der Reihenfolge aufgeführt, in der Sie sie auswerten CloudFront möchten. Weitere Informationen finden Sie unter Pfadmuster.
   - Die Cache-Verhaltensweisen leiten Anforderungen an die Ursprünge weiter, deren Origin Protocol Policy-Einstellung Sie geändert haben.

Installieren Sie ein SSL/TLS Zertifikat auf Ihrem benutzerdefinierten Ursprung

Sie können ein SSL/TLS Zertifikat aus den folgenden Quellen für Ihren benutzerdefinierten Ursprung verwenden:

- Wenn Ihr Ursprung ein Elastic Load Balancing-Load Balancer ist, können Sie ein Zertifikat verwenden, das von AWS Certificate Manager (ACM) bereitgestellt wird. Sie können auch ein Zertifikat verwenden, das von einer vertrauenswürdigen Zertifizierungsstelle signiert und in ACM importiert wurde.
- Für andere Ursprünge als Elastic Load Balancing Load Balancer müssen Sie ein Zertifikat verwenden, das von einer vertrauenswürdigen Drittanbieter-Zertifizierungsstelle (CA) signiert wurde, z. B. Comodo oder DigiCert Symantec.

Das vom Ursprungs zurückgegebene Zertifikat muss einen der folgenden Domänennamen enthalten:

- Der Domainname im Quelldomänenfeld des Ursprungs (das `DomainName` Feld in der CloudFront API).
- Den Domänennamen im `Host`-Header, wenn das Cache-Verhalten so konfiguriert ist, dass der `Host`-Header zum Ursprung weiterleitet.

Wenn HTTPS CloudFront verwendet, um mit Ihrem Ursprung zu kommunizieren, CloudFront überprüft, ob das Zertifikat von einer vertrauenswürdigen Zertifizierungsstelle ausgestellt wurde. CloudFront unterstützt dieselben Zertifizierungsstellen wie Mozilla. Die aktuelle Liste finden Sie

unter Mozilla-Liste der CA-Zertifikate. Sie können kein selbstsigniertes Zertifikat für die HTTPS-Kommunikation zwischen CloudFront und Ihrem Ursprung verwenden.

> **Important**
>
> Wenn der Ursprungsserver ein abgelaufenes Zertifikat, ein ungültiges Zertifikat oder ein selbstsigniertes Zertifikat zurückgibt oder wenn der Ursprungsserver die Zertifikatskette in der falschen Reihenfolge zurückgibt, CloudFront bricht er die TCP-Verbindung ab, gibt den HTTP-Statuscode 502 (Bad Gateway) an den Viewer zurück und setzt den `X-Cache` Header auf. `Error from cloudfront` Außerdem wird die TCP-Verbindung unterbrochen, wenn die gesamte Zertifikatkette, einschließlich des Zwischenzertifikats CloudFront , nicht vorhanden ist.

Erfordern Sie HTTPS für die Kommunikation zwischen CloudFront und Ihrem Amazon S3-Ursprung

Wenn Ihr Ursprung ein Amazon S3-Bucket ist, CloudFront hängen Ihre Optionen für die Verwendung von HTTPS für die Kommunikation mit dem Bucket davon ab, wie Sie den Bucket verwenden. Wenn Ihr Amazon S3-Bucket als Website-Endpunkt konfiguriert ist, können Sie nicht so konfigurieren CloudFront , dass HTTPS für die Kommunikation mit Ihrem Ursprung verwendet wird, da Amazon S3 in dieser Konfiguration keine HTTPS-Verbindungen unterstützt.

Wenn Ihr Ursprung ein Amazon S3-Bucket ist, der HTTPS-Kommunikation unterstützt, CloudFront leitet es Anfragen an S3 weiter, indem es das Protokoll verwendet, das die Zuschauer zum Senden der Anfragen verwendet haben. Die Standardeinstellung von Protokoll (nur benutzerdefinierte Ursprünge) ist Match Viewer (An Betrachter anpassen) und kann nicht geändert werden. Wenn Sie jedoch Origin Access Control (OAC) für Ihren Amazon S3-Ursprung aktivieren, hängt die zwischen Amazon S3 CloudFront und Amazon S3 verwendete Kommunikation von Ihren Einstellungen ab. Weitere Informationen finden Sie unter Erstellen einer neuen Ursprungszugriffssteuerung.

Wenn Sie HTTPS für die Kommunikation zwischen Amazon S3 CloudFront und Amazon S3 benötigen, müssen Sie den Wert der Viewer Protocol Policy so ändern, dass HTTP auf HTTPS oder Nur HTTPS umgeleitet wird. Das Verfahren weiter unten in diesem Abschnitt erklärt, wie Sie die CloudFront Konsole verwenden, um die Viewer Protocol Policy zu ändern. Informationen zur Verwendung der CloudFront API zur Aktualisierung des `ViewerProtocolPolicy` Elements für eine Distribution finden Sie UpdateDistribution in der CloudFront Amazon-API-Referenz.

Wenn Sie HTTPS mit einem Amazon S3-Bucket verwenden, der HTTPS-Kommunikation unterstützt, stellt Amazon S3 das SSL/TLS Zertifikat bereit, sodass Sie dies nicht tun müssen.

Erzwingen von HTTPS für einen Amazon-S3-Ursprung

Das folgende Verfahren zeigt Ihnen, wie Sie konfigurieren CloudFront , dass HTTPS für Ihren Amazon S3-Ursprung erforderlich ist.

So konfigurieren Sie CloudFront , dass HTTPS für Ihren Amazon S3-Ursprung erforderlich ist

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home.
2. Wählen Sie im oberen Bereich der CloudFront Konsole die ID für die Distribution aus, die Sie aktualisieren möchten.
3. Wählen Sie auf der Registerkarte Behaviors das zu aktualisierende Cache-Verhalten und anschließend Edit aus.
4. Geben Sie einen der folgenden Werte für Viewer Protocol Policy an:

   Redirect HTTP to HTTPS

   Zuschauer können beide Protokolle verwenden, aber HTTP-Anfragen werden automatisch zu HTTPS-Anfragen umgeleitet. CloudFront gibt den HTTP-Statuscode 301 (Moved Permanently) zusammen mit der neuen HTTPS-URL zurück. Der Viewer sendet die Anfrage dann erneut CloudFront unter Verwendung der HTTPS-URL.

   > **Important**
   >
   > CloudFront leitet`DELETE`,, `OPTIONS` `PATCH``POST`, oder `PUT` Anfragen nicht von HTTP zu HTTPS um. Wenn Sie ein Cacheverhalten so konfigurieren, dass es zu HTTPS umleitet, CloudFront reagiert auf HTTP `DELETE``OPTIONS`,`PATCH`,`POST`, oder `PUT` Anfragen für dieses Cacheverhalten mit dem HTTP-Statuscode 403 (Verboten).

   Wenn ein Viewer eine HTTP-Anforderung veranlasst, die an eine HTTPS-Adresse umgeleitet wird, berechnet CloudFront Gebühren für beide Anforderungen. Bei der HTTP-Anfrage fallen nur Gebühren für die Anfrage und für die Header an, die an den Viewer CloudFront zurückgegeben werden. Bezüglich der HTTPS-Anforderung fallen Gebühren für die

Anforderung sowie für die vom Ursprung zurückgegebenen Header und das Objekt an, das vom Ursprung zurückgegeben wird.

HTTPS Only

Viewer können auf Ihre Inhalte nur zugreifen, wenn sie HTTPS verwenden. Wenn ein Viewer eine HTTP-Anfrage anstelle einer HTTPS-Anfrage sendet, CloudFront gibt er den HTTP-Statuscode 403 (Forbidden) zurück und gibt das Objekt nicht zurück.

5. Wählen Sie Yes, Edit aus.
6. Wiederholen Sie die Schritte 3 bis 5 für jedes weitere Cacheverhalten, für das Sie HTTPS zwischen Viewern und CloudFront CloudFront und zwischen S3 benötigen.
7. Vergewissern Sie sich, dass folgende Punkte erfüllt sind, bevor Sie die aktualisierte Konfiguration in einer Produktionsumgebung verwenden:

   - Das Pfadmuster eines jeden Cache-Verhaltens wird nur auf die Anforderungen angewendet, für die die Viewer HTTPS verwenden sollen.
   - Die Cache-Verhaltensweisen werden in der Reihenfolge aufgeführt, in der Sie sie auswerten CloudFront möchten. Weitere Informationen finden Sie unter Pfadmuster.
   - Die Cache-Verhaltensweisen leiten Anforderungen an die richtigen Ursprünge weiter.

Unterstützte Protokolle und Chiffren zwischen Viewern und CloudFront

Wenn du HTTPS zwischen Zuschauern und deiner CloudFront Distribution benötigst, musst du eine Sicherheitsrichtlinie wählen, die die folgenden Einstellungen festlegt:

- Das SSL/TLS Mindestprotokoll, das für die Kommunikation mit Zuschauern CloudFront verwendet wird.
- Die Chiffren, die zur Verschlüsselung der Kommunikation mit Zuschauern verwendet werden CloudFront können.

Um eine Sicherheitsrichtlinie auszuwählen, geben Sie den anwendbaren Wert für Sicherheitsrichtlinie (SSL/TLS Mindestversion) an. In der folgenden Tabelle sind die Protokolle und Chiffren aufgeführt, die für jede Sicherheitsrichtlinie verwendet CloudFront werden können.

Ein Viewer muss mindestens eine der unterstützten Chiffren unterstützen, mit denen eine HTTPS-Verbindung hergestellt werden kann. CloudFront CloudFront wählt aus den vom Viewer unterstützten

Chiffren eine Chiffre in der aufgeführten Reihenfolge aus. Siehe auch OpenSSL-, s2n- und RFC-Verschlüsselungsnamen.

	Sicherheitsrichtlinie								
	SSLv3	TLSv1	TLSv1_6	TLSv1.016	TLSv1.018	TLSv1.019	TLSv1.021	TLSv1.025	TLSv1.3_2025
Unterstützte Protokolle SSL/TLS									
TLSv1.3	♦	♦	♦	♦	♦	♦	♦	♦	♦
TLSv1.2	♦	♦	♦	♦	♦	♦	♦	♦	
TLSv1.1	♦	♦	♦	♦					
TLSv1	♦	♦	♦						
SSLv3	♦								
Unterstützte TLSv1.3 Chiffren									
TLS_AES_128_GCM_SHA256	♦	♦	♦	♦	♦	♦	♦	♦	♦
TLS_AES_256_GCM_SHA384	♦	♦	♦	♦	♦	♦	♦	♦	♦
TLS_CHACHA20_POLY1305_SHA256	♦	♦	♦	♦	♦	♦	♦		♦
Unterstützte ECDSA-Verschlüsselungen									
ECDHE-ECDSA-AES128-GCM-SHA256	♦	♦	♦	♦	♦	♦	♦	♦	
ECDHE-ECDSA-AES128-SHA256	♦	♦	♦	♦	♦	♦			

	Sicherheitsrichtlinie								
	SSLv3	TLSv1	TLSv1_6	TLSv1.016	TLSv1.018	TLSv1.019	TLSv1.021	TLSv1.025	TLSv1.3_2025
ECDHE-ECDSA-AES128-SHA	♦	♦	♦	♦					
ECDHE-ECDSA-AES256-GCM-SHA384	♦	♦	♦	♦	♦	♦	♦	♦	
ECDHE-ECDSA-CHACHA20-POLY1305	♦	♦	♦	♦	♦	♦	♦		
ECDHE-ECDSA-AES256-SHA384	♦	♦	♦	♦	♦	♦			
ECDHE-ECDSA-AES256-SHA	♦	♦	♦	♦					
Unterstützte RSA-Verschlüsselungen									
ECDHE-RSA-AES128-GCM-SHA256	♦	♦	♦	♦	♦	♦	♦	♦	
ECDHE-RSA-AES128-SHA256	♦	♦	♦	♦	♦	♦			
ECDHE-RSA-AES128-SHA	♦	♦	♦	♦					
ECDHE-RSA-AES256-GCM-SHA384	♦	♦	♦	♦	♦	♦	♦	♦	

	Sicherheitsrichtlinie								
	SSLv3	TLSv1	TLSv1_6	TLSv1.016	TLSv1.018	TLSv1.019	TLSv1.021	TLSv1.025	TLSv1.3_2025
ECDHE-RSA-CHACHA20-POLY1305	♦	♦	♦	♦	♦	♦	♦		
ECDHE-RSA-AES256-SHA384	♦	♦	♦	♦	♦	♦			
ECDHE-RSA-AES256-SHA	♦	♦	♦	♦					
AES128-GCM-SHA256	♦	♦	♦	♦	♦				
AES256-GCM-SHA384	♦	♦	♦	♦	♦				
AES128-SHA256	♦	♦	♦	♦	♦				
AES256-SHA	♦	♦	♦	♦					
AES128-SHA	♦	♦	♦	♦					
DES-CBC3-SHA	♦	♦							
RC4-MD5	♦								

OpenSSL-, s2n- und RFC-Verschlüsselungsnamen

OpenSSL und s2n verwenden für Chiffren andere Namen als die TLS-Standards verwenden (RFC 2246, RFC 4346, RFC 5246, und RFC 8446). Die folgende Tabelle ordnet den OpenSSL-Namen und s2n Namen den RFC-Namen für jedes Verschlüsselungsverfahren zu.

CloudFront unterstützt sowohl den klassischen als auch den quantensicheren Schlüsselaustausch. Unterstützt für den klassischen Schlüsselaustausch, der elliptische Kurven verwendet, Folgendes: CloudFront

- `prime256v1`
- `X25519`
- `secp384r1`

Unterstützt für den quantensicheren Schlüsselaustausch Folgendes: CloudFront

- `X25519MLKEM768`
- `SecP256r1MLKEM768`

> **Note**
>
> Quantum-safe Der Schlüsselaustausch wird nur mit TLS 1.3 unterstützt. TLS 1.2 und frühere Versionen unterstützen keinen quantensicheren Schlüsselaustausch.

Weitere Informationen finden Sie unter den folgenden Themen:

- Post-Quantum Kryptografie
- Kryptographie-Algorithmen und AWS-Services
- Hybrider Schlüsselaustausch in TLS 1.3

Weitere Informationen zu den Zertifikatsanforderungen für finden Sie CloudFront unterAnforderungen für die Verwendung von SSL/TLS Zertifikaten mit CloudFront.

OpenSSL- und s2n-Chiffrenname	RFC-Verschlüsselungsname
Unterstützte TLSv1.3 Chiffren	
TLS_AES_128_GCM_SHA256	TLS_AES_128_GCM_SHA256
TLS_AES_256_GCM_SHA384	TLS_AES_256_GCM_SHA384
TLS_CHACHA20_POLY1305_SHA256	TLS_CHACHA20_POLY1305_SHA256
Unterstützte ECDSA-Verschlüsselungen	
ECDHE-ECDSA-AES128-GCM-SHA256	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256

OpenSSL- und s2n-Chiffrenname	RFC-Verschlüsselungsname
ECDHE-ECDSA-AES128-SHA256	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
ECDHE-ECDSA-AES128-SHA	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA
ECDHE-ECDSA-AES256-GCM-SHA384	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
ECDHE-ECDSA-CHACHA20-POLY1305	TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256
ECDHE-ECDSA-AES256-SHA384	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384
ECDHE-ECDSA-AES256-SHA	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA
Unterstützte RSA-Verschlüsselungen	
ECDHE-RSA-AES128-GCM-SHA256	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
ECDHE-RSA-AES128-SHA256	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
ECDHE-RSA-AES128-SHA	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA
ECDHE-RSA-AES256-GCM-SHA384	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
ECDHE-RSA-CHACHA20-POLY1305	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256
ECDHE-RSA-AES256-SHA384	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384

OpenSSL- und s2n-Chiffrenname	RFC-Verschlüsselungsname
ECDHE-RSA-AES256-SHA	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA
AES128-GCM-SHA256	TLS_RSA_WITH_AES_128_GCM_SHA256
AES256-GCM-SHA384	TLS_RSA_WITH_AES_256_GCM_SHA384
AES128-SHA256	TLS_RSA_WITH_AES_128_CBC_SHA256
AES256-SHA	TLS_RSA_WITH_AES_256_CBC_SHA
AES128-SHA	TLS_RSA_WITH_AES_128_CBC_SHA
DES-CBC3-SHA	TLS_RSA_WITH_3DES_EDE_CBC_SHA
RC4-MD5	TLS_RSA_WITH_RC4_128_MD5

Unterstützte Signaturschemata zwischen Zuschauern und CloudFront

CloudFront unterstützt die folgenden Signaturschemata für Verbindungen zwischen Zuschauern undCloudFront.

	Sicherheitsrichtlinie								
Signaturschemata	SSLv3	TLSv1	TLSv1_6	TLSv1.016	TLSv1.018	TLSv1.019	TLSv1.021	TLSv1.025	TLSv1.3_2025
TLS_SIGNATURE_SCHEME_RSA_PSS_PSS_SHA256	♦	♦	♦	♦	♦	♦	♦	♦	♦
TLS_SIGNATURE_SCHEME_RSA_PSS_PSS_SHA384	♦	♦	♦	♦	♦	♦	♦	♦	♦

	Sicherheitsrichtlinie								
Signaturschemata	SSLv3	TLSv1	TLSv1_6	TLSv1.016	TLSv1.018	TLSv1.019	TLSv1.021	TLSv1.025	TLSv1.3_2025
TLS_SIGNATURE_SCHEME_RSA_PSS_PSS_SHA512	♦	♦	♦	♦	♦	♦	♦	♦	♦
TLS_SIGNATURE_SCHEME_RSA_PSS_RSAE_SHA256	♦	♦	♦	♦	♦	♦	♦	♦	♦
TLS_SIGNATURE_SCHEME_RSA_PSS_RSAE_SHA384	♦	♦	♦	♦	♦	♦	♦	♦	♦
TLS_SIGNATURE_SCHEME_RSA_PSS_RSAE_SHA512	♦	♦	♦	♦	♦	♦	♦	♦	♦
TLS_SIGNATURE_SCHEME_RSA_PKCS1_SHA256	♦	♦	♦	♦	♦	♦	♦	♦	♦
TLS_SIGNATURE_SCHEME_RSA_PKCS1_SHA384	♦	♦	♦	♦	♦	♦	♦	♦	♦

	Sicherheitsrichtlinie								
Signaturschemata	SSLv3	TLSv1	TLSv1_6	TLSv1.016	TLSv1.018	TLSv1.019	TLSv1.021	TLSv1.025	TLSv1.3_2025
TLS_SIGNATURE_SCHEME_RSA_PKCS1_SHA512	♦	♦	♦	♦	♦	♦	♦	♦	♦
TLS_SIGNATURE_SCHEME_RSA_PKCS1_SHA224	♦	♦	♦	♦	♦	♦	♦		
TLS_SIGNATURE_SCHEME_ECDSA_SHA256	♦	♦	♦	♦	♦	♦	♦	♦	♦
TLS_SIGNATURE_SCHEME_ECDSA_SHA384	♦	♦	♦	♦	♦	♦	♦	♦	♦
TLS_SIGNATURE_SCHEME_ECDSA_SHA512	♦	♦	♦	♦	♦	♦	♦	♦	♦
TLS_SIGNATURE_SCHEME_ECDSA_SHA224	♦	♦	♦	♦	♦	♦	♦		
TLS_SIGNATURE_SCHEME_ECDSA_SECP256R1_SHA256	♦	♦	♦	♦	♦	♦	♦	♦	♦

	Sicherheitsrichtlinie								
Signaturschemata	SSLv3	TLSv1	TLSv1_6	TLSv1.016	TLSv1.018	TLSv1.019	TLSv1.021	TLSv1.025	TLSv1.3_2025
TLS_SIGNATURE_SCHEME_ECDSA_SECP384R1_SHA384	♦	♦	♦	♦	♦	♦	♦	♦	♦
TLS_SIGNATURE_SCHEME_RSA_PKCS1_SHA1	♦	♦	♦	♦					
TLS_SIGNATURE_SCHEME_ECDSA_SHA1	♦	♦	♦	♦					

Unterstützte Protokolle und Chiffren zwischen CloudFront und dem Ursprung

Wenn Sie HTTPS zwischen CloudFront und Ihrem Ursprung benötigen, können Sie entscheiden, welches SSL/TLS Protokoll die sichere Verbindung zulassen soll, und CloudFront können mithilfe einer der in der folgenden Tabelle aufgeführten ECDSA- oder RSA-Verschlüsselungen eine Verbindung zum Ursprung herstellen. Ihr Ursprung muss mindestens eine dieser Chiffren unterstützen, um eine HTTPS-Verbindung CloudFront zu Ihrem Ursprung herzustellen.

OpenSSL und s2n verwenden für Chiffren andere Namen als die TLS-Standards verwenden (RFC 2246, RFC 4346, RFC 5246, und RFC 8446). Die folgende Tabelle beinhaltet den OpenSSL-Namen und s2n Namen den RFC-Namen für jedes Verschlüsselungsverfahren.

CloudFront Unterstützt für Chiffren mit Schlüsselaustauschalgorithmen für elliptische Kurven die folgenden elliptischen Kurven:

- prime256v1
- secp384r1

- X25519

OpenSSL- und s2n-Chiffrenname	RFC-Verschlüsselungsname
Unterstützte ECDSA-Verschlüsselungen	
ECDHE-ECDSA-AES256-GCM-SHA384	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
ECDHE-ECDSA-AES256-SHA384	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384
ECDHE-ECDSA-AES256-SHA	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA
ECDHE-ECDSA-AES128-GCM-SHA256	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
ECDHE-ECDSA-AES128-SHA256	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
ECDHE-ECDSA-AES128-SHA	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA
Unterstützte RSA-Verschlüsselungen	
ECDHE-RSA-AES256-GCM-SHA384	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
ECDHE-RSA-AES256-SHA384	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384
ECDHE-RSA-AES256-SHA	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA
ECDHE-RSA-AES128-GCM-SHA256	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256

OpenSSL- und s2n-Chiffrenname	RFC-Verschlüsselungsname
ECDHE-RSA-AES128-SHA256	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
ECDHE-RSA-AES128-SHA	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA
AES256-SHA	TLS_RSA_WITH_AES_256_CBC_SHA
AES128-SHA	TLS_RSA_WITH_AES_128_CBC_SHA
DES-CBC3-SHA	TLS_RSA_WITH_3DES_EDE_CBC_SHA
RC4-MD5	TLS_RSA_WITH_RC4_128_MD5

Unterstützte Signaturschemas zwischen und dem Ursprung CloudFront

CloudFront unterstützt die folgenden Signaturschemata für Verbindungen zwischen CloudFront und dem Ursprung.

- TLS_SIGNATURE_SCHEME_RSA_PKCS1_SHA256
- TLS_SIGNATURE_SCHEME_RSA_PKCS1_SHA384
- TLS_SIGNATURE_SCHEME_RSA_PKCS1_SHA512
- TLS_SIGNATURE_SCHEME_RSA_PKCS1_SHA224
- TLS_SIGNATURE_SCHEME_ECDSA_SHA256
- TLS_SIGNATURE_SCHEME_ECDSA_SHA384
- TLS_SIGNATURE_SCHEME_ECDSA_SHA512
- TLS_SIGNATURE_SCHEME_ECDSA_SHA224
- TLS_SIGNATURE_SCHEME_RSA_PKCS1_SHA1
- TLS_SIGNATURE_SCHEME_ECDSA_SHA1

Verwenden von alternativen Domainnamen und HTTPS

Wenn Sie Ihren eigenen Domain-Namen in den URLs für Ihre Dateien verwenden möchten (z. B. `https://www.example.com/image.jpg`) und wünschen, dass Ihre Viewer HTTPS

verwenden, müssen Sie die zusätzlichen Schritte in den folgenden Themen durchführen. (Wenn Sie beispielsweise den CloudFront Standard-Distributionsdomänennamen in Ihren URLs verwenden`https://d111111abcdef8.cloudfront.net/image.jpg`, folgen Sie stattdessen den Anweisungen im folgenden Thema:Erfordern Sie HTTPS für die Kommunikation zwischen Zuschauern und CloudFront.)

> **Important**
>
> Wenn Sie Ihrer Distribution ein Zertifikat hinzufügen, wird das Zertifikat CloudFront sofort an alle Edge-Standorte weitergegeben. Sobald neue Edge-Standorte verfügbar sind, wird das Zertifikat von CloudFront auch an diese Standorte verteilt. Sie können die Edge-Standorte, an die die CloudFront Zertifikate weitergegeben werden, nicht einschränken.

Themen

- Wählen Sie aus, wie CloudFront HTTPS-Anfragen verarbeitet werden
- Anforderungen für die Verwendung von SSL/TLS Zertifikaten mit CloudFront
- Kontingente für die Verwendung von SSL/TLS Zertifikaten mit CloudFront (HTTPS nur zwischen Zuschauern und CloudFront nur)
- Konfigurieren alternativer Domainnamen und HTTPS
- Ermitteln Sie die Größe des öffentlichen Schlüssels in einem SSL/TLS RSA-Zertifikat
- Erhöhen Sie die Kontingente für Zertifikate SSL/TLS
- SSL/TLS Zertifikate rotieren
- Kehren Sie von einem benutzerdefinierten SSL/TLS Zertifikat zum Standardzertifikat zurück CloudFront
- Wechseln Sie von einem benutzerdefinierten SSL/TLS Zertifikat mit dedizierten IP-Adressen zu SNI

Wählen Sie aus, wie CloudFront HTTPS-Anfragen verarbeitet werden

Wenn Sie möchten, dass Ihre Zuschauer HTTPS verwenden und alternative Domainnamen für Ihre Dateien verwenden, wählen Sie eine der folgenden Optionen für die Art und Weise, wie CloudFront HTTPS-Anfragen verarbeitet werden:

- Verwenden der Servernamensanzeige (Server Name Indication, SNI) – Empfohlen

- Verwenden einer dedizierten IP-Adresse an jedem Edge-Standort

In diesem Abschnitt wird erläutert, wie beide Optionen funktionieren.

Verwenden von SNI für die Beantwortung von HTTPS-Anforderungen (funktioniert für die meisten Clients)

Die Servernamensanzeige (SNI) ist eine Erweiterung des TLS-Protokolls, die in Browsern und Clients unterstützt wird, die nach 2010 veröffentlicht wurden. Wenn Sie so konfigurieren CloudFront , dass HTTPS-Anfragen mithilfe von SNI verarbeitet CloudFront werden, verknüpfen Sie Ihren alternativen Domainnamen mit einer IP-Adresse für jeden Edge-Standort. Sobald ein Viewer Inhalte von Ihnen durch Senden einer HTTPS-Anforderung abruft, leitet DNS die Anforderung an die IP-Adresse des korrekten Edge-Standorts weiter. Die IP-Adresse für Ihren Domainnamen wird während der SSL/TLS Handshake-Aushandlung festgelegt; die IP-Adresse ist nicht für Ihre Verteilung bestimmt.

Die SSL/TLS Verhandlung findet früh im Prozess des Aufbaus einer HTTPS-Verbindung statt. Wenn nicht sofort festgestellt werden CloudFront kann, für welche Domain die Anfrage gilt, wird die Verbindung unterbrochen. Sobald ein Viewer, der SNI unterstützt, eine HTTPS-Anforderung zwecks Abrufs Ihrer Inhalte sendet, geschieht Folgendes:

1. Der Viewer erhält automatisch den Domainnamen aus der Anforderungs-URL und fügt ihn der SNI-Erweiterung der TLS-ClientHello-Nachricht hinzu.
2. Wenn der TLS-Client Hallo CloudFront empfängt, verwendet er den Domainnamen in der SNI-Erweiterung, um die passende CloudFront Distribution zu finden, und sendet das zugehörige TLS-Zertifikat zurück.
3. Der Zuschauer und die SSL/TLS Verhandlung CloudFront durchführen.
4. CloudFront gibt den angeforderten Inhalt an den Betrachter zurück.

Eine aktuelle Liste der Browser, die SNI unterstützen, finden Sie im Wikipedia-Eintrag Server Name Indication.

Wenn Sie SNI nutzen möchten, jedoch einige Ihrer Benutzer Browser verwenden, die SNI nicht unterstützen, haben Sie mehrere Möglichkeiten:

- Konfigurieren Sie CloudFront es so, dass HTTPS-Anfragen mithilfe von dedizierten IP-Adressen anstelle von SNI bedient werden. Weitere Informationen finden Sie unter Verwenden dedizierter IP-Adressen für die Beantwortung von HTTPS-Anforderungen (funktioniert für alle Clients).

- Verwenden Sie das CloudFront SSL/TLS Zertifikat anstelle eines benutzerdefinierten Zertifikats. Dies erfordert, dass Sie den CloudFront Domainnamen für Ihre Distribution in den URLs für Ihre Dateien verwenden, `https://d111111abcdef8.cloudfront.net/logo.png` z. B.

  Wenn Sie das CloudFront Standardzertifikat verwenden, müssen Viewer das SSL-Protokoll TLSv1 oder höher unterstützen. CloudFront unterstützt SSLv3 mit dem Standardzertifikat nicht. CloudFront

  Sie müssen auch das verwendete SSL/TLS Zertifikat von einem benutzerdefinierten Zertifikat auf das CloudFront Standardzertifikat ändern: CloudFront

  - Wenn Sie Ihre Distribution noch nicht zur Distribution Ihrer Inhalte genutzt haben, können Sie einfach die Konfiguration ändern. Weitere Informationen finden Sie unter Eine Verteilung aktualisieren.
  - Wenn Sie Ihre Distribution verwendet haben, um Ihre Inhalte zu verteilen, müssen Sie eine neue CloudFront Distribution erstellen und die URLs für Ihre Dateien ändern, um die Zeit, in der Ihre Inhalte nicht verfügbar sind, zu reduzieren oder zu vermeiden. Weitere Informationen finden Sie unter Kehren Sie von einem benutzerdefinierten SSL/TLS Zertifikat zum Standardzertifikat zurück CloudFront.
- Wenn Sie Einfluss darauf haben, welche Browser Ihre Benutzer verwenden, können Sie veranlassen, dass sie ihre Browser auf eine Version aktualisieren, die SNI unterstützt.
- Verwenden Sie HTTP anstelle von HTTPS.

Verwenden dedizierter IP-Adressen für die Beantwortung von HTTPS-Anforderungen (funktioniert für alle Clients)

Die Servernamensanzeige (SNI) ist eine Möglichkeit, eine Anforderung mit einer Domäne zu verknüpfen. Eine weitere Möglichkeit ist die Verwendung einer dedizierten IP-Adresse. Wenn Sie Benutzer haben, die nicht zu einem Browser oder Client wechseln können, der nach 2010 veröffentlicht wurde, können Sie eine dedizierte IP-Adresse für die Bedienung von HTTPS-Anforderungen verwenden. Eine aktuelle Liste der Browser, die SNI unterstützen, finden Sie im Wikipedia-Eintrag Server Name Indication.

> **Important**
>
> Wenn Sie CloudFront die Verarbeitung von HTTPS-Anfragen über dedizierte IP-Adressen konfigurieren, fällt eine zusätzliche monatliche Gebühr an. Die Gebühr beginnt, wenn Sie Ihr SSL/TLS Zertifikat einer Verteilung zuordnen und die Verteilung aktivieren. Weitere

Informationen zur CloudFront Preisgestaltung finden Sie unter CloudFront Amazon-Preise. Zusätzliche, in diesem Zusammenhang interessante Details finden Sie unter Using the Same Certificate for Multiple CloudFront Distributions.

Wenn Sie CloudFront die Bearbeitung von HTTPS-Anfragen mithilfe dedizierter IP-Adressen konfigurieren, verknüpfen CloudFront Sie Ihr Zertifikat mit einer dedizierten IP-Adresse an jedem CloudFront Edge-Standort. Sobald ein Viewer eine HTTPS-Anforderung zwecks Abrufs Ihrer Inhalte sendet, geschieht Folgendes:

1. DNS leitet die Anforderung an die IP-Adresse Ihrer Distribution am jeweils zuständigen Edge-Standort weiter.
2. Wenn eine Client-Anfrage die SNI-Erweiterung in der `ClientHello` Nachricht bereitstellt, CloudFront sucht nach einer Distribution, die diesem SNI zugeordnet ist.
   - Wenn es eine Übereinstimmung gibt, CloudFront beantwortet die Anfrage mit dem SSL/TLS Zertifikat.
   - Wenn es keine Übereinstimmung gibt, CloudFront verwendet stattdessen die IP-Adresse, um Ihre Distribution zu identifizieren und zu bestimmen, welches SSL/TLS Zertifikat an den Betrachter zurückgegeben werden soll.
3. Der Betrachter und die SSL/TLS Verhandlung CloudFront führen mithilfe Ihres SSL/TLS Zertifikats durch.
4. CloudFront gibt den angeforderten Inhalt an den Betrachter zurück.

Diese Methode funktioniert für jede HTTPS-Anforderung, unabhängig von dem Browser oder Viewer anderer Art, den der Benutzer verwendet.

Note

Dedizierte IPs sind keine statischen IPs und können sich im Laufe der Zeit ändern. Die IP-Adresse, die für den Edge-Standort zurückgegeben wird, wird dynamisch aus den IP-Adressbereichen der CloudFront Edge-Serverliste zugewiesen.
Die IP-Adressbereiche für CloudFront Edge-Server können sich ändern. Um über Änderungen der IP-Adresse informiert zu werden, abonnieren Sie AWS Public IP Address Changes mithilfe von Amazon SNS.

Fordern Sie die Erlaubnis an, drei oder mehr dedizierte SSL/TLS IP-Zertifikate zu verwenden

Wenn Sie die Erlaubnis benötigen, drei oder mehr SSL/TLS dedizierte IP-Zertifikate dauerhaft mit ihnen zu verknüpfen CloudFront, gehen Sie wie folgt vor. Weitere Informationen zu HTTPS-Anforderungen finden Sie unter Wählen Sie aus, wie CloudFront HTTPS-Anfragen verarbeitet werden.

> **Note**
>
> Dieses Verfahren ist für die Verwendung von drei oder mehr dedizierten IP-Zertifikaten in Ihren CloudFront Distributionen vorgesehen. Der Standardwert lautet 2. Beachten Sie, dass Sie nicht mehr als ein SSL-Zertifikat an eine Distribution binden können.
> Sie können einer CloudFront Distribution jeweils nur ein einziges SSL/TLS Zertifikat zuordnen. Diese Zahl bezieht sich auf die Gesamtzahl der dedizierten IP-SSL-Zertifikate, die Sie für alle Ihre CloudFront Distributionen verwenden können.

Um die Genehmigung zur Verwendung von drei oder mehr Zertifikaten mit einer CloudFront Distribution anzufordern

1. Besuchen Sie das Support Center und erstellen Sie einen Fall.
2. Geben Sie die Anzahl der Zertifikate an, die Sie benötigen, und beschreiben Sie die Umstände in Ihrer Beantragung. Ihr Konto wird so schnell wie möglich aktualisiert.
3. Fahren Sie mit dem nächsten Schritt fort.

Anforderungen für die Verwendung von SSL/TLS Zertifikaten mit CloudFront

Die Anforderungen für SSL/TLS Zertifikate werden in diesem Thema beschrieben. Sie gelten, sofern nicht anders vermerkt, für die beiden folgenden Elemente:

- Zertifikate für die Verwendung von HTTPS zwischen Zuschauern und CloudFront
- Zertifikate für die Verwendung von HTTPS zwischen CloudFront und Ihrem Ursprung

Themen

- Zertifikataussteller
- AWS-Region for AWS Certificate Manager
- Zertifikatformat
- Zwischenzertifikate
- Schlüsseltyp
- Privater Schlüssel
- Berechtigungen
- Länge des Zertifikatschlüssels
- Unterstützte Typen von Zertifikaten
- Ablaufdatum des Zertifikats und Zertifikaterneuerung
- Domainnamen in der CloudFront Distribution und im Zertifikat
- Minimale SSL/TLS Protokollversion
- Unterstützte HTTP-Versionen

Zertifikataussteller

Wir empfehlen, ein öffentliches Zertifikat zu verwenden, das von AWS Certificate Manager (ACM) ausgestellt wurde. Weitere Informationen zu ACM finden Sie im AWS Certificate Manager - Benutzerhandbuch. Um ein ACM-Zertifikat für eine CloudFront Distribution zu verwenden, stellen Sie sicher, dass Sie das Zertifikat in der Region USA Ost (Nord-Virginia) anfordern (oder importieren) (`us-east-1`).

CloudFront unterstützt dieselben Zertifizierungsstellen (CAs) wie Mozilla. Wenn Sie ACM also nicht verwenden, verwenden Sie ein Zertifikat, das von einer Zertifizierungsstelle ausgestellt wurde, die in der Mozilla Included CA Certificate List aufgeführt ist.

TLS-Zertifikate, die von der Quelle verwendet werden, die Sie für Ihre CloudFront Distribution angegeben haben, müssen ebenfalls von der Zertifizierungsstelle ausgestellt werden, die auf der Liste der in Mozilla enthaltenen CA-Zertifikate steht.

Weitere Informationen zum Abrufen und Installieren eines Zertifikats finden Sie in der Dokumentation zu Ihrer HTTP-Server-Software und der Dokumentation, die von der Zertifizierungsstelle bereitgestellt wird.

AWS-Region for AWS Certificate Manager

Um ein Zertifikat in AWS Certificate Manager (ACM) zu verwenden, das HTTPS zwischen Zuschauern und erfordertCloudFront, stellen Sie sicher, dass Sie das Zertifikat in der Region USA Ost (Nord-Virginia) () anfordern (oder importieren`us-east-1`).

Wenn Sie HTTPS zwischen CloudFront und Ihrem Absender benötigen und Sie einen Load Balancer in Elastic Load Balancing als Absender verwenden, können Sie das Zertifikat in jedem beliebigen Land anfordern oder importieren. AWS-Region

Zertifikatformat

Das Zertifikat muss im X.509 PEM-Format vorliegen. Dies ist das Standardformat bei der Verwendung von AWS Certificate Manager.

Zwischenzertifikate

Wenn Sie eine Drittanbieter-Zertifizierungsstelle (CA) verwenden, müssen alle Zwischenzertifikate in der Zertifikatkette aufgelistet sein, die sich in der Datei `.pem` befinden, beginnend mit einem Zwischenzertifikat für die Zertifizierungsstelle, die das Zertifikat für Ihre Domäne signiert hat. In der Regel finden Sie eine Datei auf der Website der Zertifizierungsstelle, in der die Zwischen- und Stammzertifikate in der richtigen Reihenfolge aufgelistet sind.

> **Important**
>
> Fügen Sie Folgendes nicht hinzu: das Stammzertifikat, Zwischenzertifikate, die sich nicht im Vertrauenspfad befinden, und das Public-Key-Zertifikat Ihrer CA.

Ein Beispiel:

```
-----BEGIN CERTIFICATE-----
Intermediate certificate 2
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
Intermediate certificate 1
-----END CERTIFICATE-----
```

Schlüsseltyp

CloudFront unterstützt öffentlich-private RSA- und ECDSA-Schlüsselpaare.

CloudFront unterstützt HTTPS-Verbindungen sowohl zu Viewern als auch zu Ursprüngen mithilfe von RSA- und ECDSA-Zertifikaten. Mit AWS Certificate Manager (ACM) können Sie RSA- oder ECDSA-Zertifikate anfordern und importieren und sie dann Ihrer Distribution zuordnen. CloudFront

Eine Liste der RSA- und ECDSA-Chiffren, die von dieser Methode unterstützt werden, die Sie in HTTPS-Verbindungen aushandeln können CloudFront , finden Sie unter und. the section called “Unterstützte Protokolle und Chiffren zwischen Viewern und CloudFront” the section called “Unterstützte Protokolle und Chiffren zwischen CloudFront und dem Ursprung”

Privater Schlüssel

Bei Verwendung eines Zertifikats von einer Drittanbieter-Zertifizierungsstelle sind folgende Punkte zu beachten:

- Der private Schlüssel muss dem öffentlichen Schlüssel des Zertifikats entsprechen.
- Der private Schlüssel muss im PEM-Format vorliegen.
- Der private Schlüssel kann nicht mit einem Passwort verschlüsselt werden.

Wenn AWS Certificate Manager (ACM) das Zertifikat bereitgestellt hat, gibt ACM den privaten Schlüssel nicht frei. Der private Schlüssel wird in ACM gespeichert und kann von AWS Diensten verwendet werden, die in ACM integriert sind.

Berechtigungen

Sie müssen die Erlaubnis haben, das Zertifikat zu verwenden und zu importieren. SSL/TLS Wenn Sie AWS Certificate Manager (ACM) verwenden, empfehlen wir, dass Sie AWS Identity and Access Management Berechtigungen verwenden, um den Zugriff auf die Zertifikate einzuschränken. Weitere Informationen finden Sie unter Identity and Access Management im AWS Certificate Manager - Benutzerhandbuch.

Länge des Zertifikatschlüssels

Die unterstützte Zertifikatsschlüsselgröße hängt vom Typ des Schlüssels und des Zertifikats ab. CloudFront

Für RSA-Zertifikate:

CloudFront unterstützt 1024-Bit-, 2048-Bit-, 3072-Bit- und 4096-Bit-RSA-Schlüssel. Die maximale Schlüssellänge für ein RSA-Zertifikat, das Sie verwenden, beträgt 4096 Bit. CloudFront

Beachten Sie, dass ACM RSA-Zertifikate mit bis zu 2048-Bit-Schlüsseln ausstellt. Um ein 3072-Bit- oder 4096-Bit-RSA-Zertifikat zu verwenden, müssen Sie das Zertifikat extern beziehen und es in ACM importieren. Danach können Sie es verwenden. CloudFront

Informationen zum Ermitteln der Länge des RSA-Schlüssels finden Sie unter Ermitteln Sie die Größe des öffentlichen Schlüssels in einem SSL/TLS RSA-Zertifikat.

Für ECDSA-Zertifikate:

CloudFront unterstützt 256-Bit- und 384-Bit-Schlüssel. Um ein ECDSA-Zertifikat in ACM zu verwenden, das HTTPS zwischen Viewern und erfordert, verwenden Sie die elliptische Kurve prime256v1 oder CloudFront secp384r1.

Unterstützte Typen von Zertifikaten

CloudFront unterstützt alle Arten von Zertifikaten, die von einer vertrauenswürdigen Zertifizierungsstelle ausgestellt wurden.

Ablaufdatum des Zertifikats und Zertifikaterneuerung

Wenn Sie Zertifikate verwenden, die Sie von einer Drittanbieter-Zertifizierungsstelle (CA) erhalten, müssen Sie die Ablaufdaten der Zertifikate überwachen und die Zertifikate, die Sie in AWS Certificate Manager (ACM) importieren oder in den AWS Identity and Access Management Zertifikatsspeicher hochladen, erneuern, bevor sie ablaufen.

Important

Probleme mit dem Ablauf von Zertifikaten vermeiden Sie, indem Sie Ihr Zertifikat mindestens 24 Stunden vor dem `NotAfter`-Wert Ihres aktuellen Zertifikats verlängern oder erneut importieren. Wenn Ihr Zertifikat innerhalb von 24 Stunden abläuft, fordern Sie ein neues Zertifikat von ACM an oder importieren Sie ein neues Zertifikat in ACM. Ordnen Sie als Nächstes das neue Zertifikat der CloudFront Distribution zu.
CloudFront verwendet möglicherweise weiterhin das vorherige Zertifikat, während Ihre Zertifikatsverlängerung oder der erneute Import ausgeführt wird. Dies ist ein asynchroner Vorgang, der bis zu 24 Stunden dauern kann, bis Ihre Änderungen CloudFront angezeigt werden.

Wenn Sie von ACM bereitgestellte Zertifikate verwenden, werden die Erneuerungen dieser Zertifikate von ACM verwaltet. Weitere Informationen finden Sie unter Verwaltete Erneuerung im AWS Certificate Manager -Benutzerhandbuch.

Domainnamen in der CloudFront Distribution und im Zertifikat

Wenn Sie eine benutzerdefinierte Herkunft verwenden, enthält das SSL/TLS Zertifikat für Ihre Herkunft einen Domainnamen im Feld Allgemeiner Name und möglicherweise mehrere weitere im Feld Alternative Betreffnamen. (CloudFront unterstützt Platzhalterzeichen in Zertifikatsdomänennamen.)

Einer der Domänennamen im Zertifikat muss mit dem Domänennamen übereinstimmen, den Sie in „Origin Domain Name (Ursprungsdomänenname)“ angeben. Wenn kein Domainname übereinstimmt, wird der HTTP-Statuscode `502 (Bad Gateway)` an den Viewer CloudFront zurückgegeben.

> **Important**
>
> Wenn Sie einer Distribution einen alternativen Domainnamen hinzufügen, wird CloudFront überprüft, ob der alternative Domainname durch das von Ihnen angehängte Zertifikat abgedeckt ist. Das Zertifikat muss den alternativen Domänennamen im Feld „Subject Alternate Name (SAN)“ des Zertifikats abdecken. Dies bedeutet, dass das SAN-Feld eine exakte Übereinstimmung mit dem alternativen Domänennamen oder einen Platzhalter auf der gleichen Ebene des alternativen Domänennamens enthalten muss, den Sie hinzufügen. Weitere Informationen finden Sie unter Voraussetzungen für die Verwendung von alternativen Domänennamen.

Minimale SSL/TLS Protokollversion

Wenn Sie dedizierte IP-Adressen verwenden, legen Sie die SSL/TLS Mindestprotokollversion für die Verbindung zwischen Zuschauern fest und CloudFront wählen Sie eine Sicherheitsrichtlinie aus.

Weitere Informationen finden Sie Sicherheitsrichtlinie (SSL/TLS Mindestversion) im Thema Referenz für alle Distributionseinstellungen.

Unterstützte HTTP-Versionen

Wenn Sie ein Zertifikat mehreren CloudFront Distributionen zuordnen, müssen alle mit dem Zertifikat verknüpften Distributionen dieselbe Option für Unterstützte HTTP-Versionen verwenden. Sie geben diese Option an, wenn Sie eine CloudFront Distribution erstellen oder aktualisieren.

Kontingente für die Verwendung von SSL/TLS Zertifikaten mit CloudFront (HTTPS nur zwischen Zuschauern und CloudFront nur)

Beachten Sie die folgenden Kontingente für die Verwendung von SSL/TLS Zertifikaten mitCloudFront. Diese Kontingente gelten nur für die SSL/TLS Zertifikate, die Sie mithilfe von AWS Certificate Manager (ACM) bereitstellen, die Sie in ACM importieren oder für die HTTPS-Kommunikation zwischen Zuschauern und in den IAM-Zertifikatsspeicher hochladen. CloudFront

Weitere Informationen finden Sie unter Erhöhen Sie die Kontingente für Zertifikate SSL/TLS.

Maximale Anzahl von Zertifikaten pro Verteilung CloudFront

Sie können jeder CloudFront Verteilung maximal ein SSL/TLS Zertifikat zuordnen.

Maximale Anzahl an Zertifikaten, die Sie in ACM importieren oder in den IAM-Zertifikatspeicher hochladen können

Wenn Sie Ihre SSL/TLS Zertifikate von einer Drittanbieter-Zertifizierungsstelle erhalten haben, müssen Sie die Zertifikate an einem der folgenden Orte speichern:

- AWS Certificate Manager – Informationen zum aktuellen Kontingent für die Anzahl der ACM-Zertifikate finden Sie unter Kontingente im AWS Certificate Manager -Benutzerhandbuch. Das angegebene Kontingent umfasst sowohl die Zertifikate, die Sie mithilfe von ACM bereitstellen, als auch die Zertifikate, die Sie in ACM importieren.
- IAM-Zertifikatsspeicher — Das aktuelle Kontingent (früher bekannt als Limit) für die Anzahl der Zertifikate, die Sie für ein AWS Konto in den IAM-Zertifikatsspeicher hochladen können, finden Sie unter IAM- und STS-Grenzwerte im IAM-Benutzerhandbuch. Sie können über die Service-Quotas-Konsole eine Kontingenterhöhung beantragen.

Maximale Anzahl von Zertifikaten pro AWS Konto (nur dedizierte IP-Adressen)

Wenn Sie HTTPS-Anforderungen unter Verwendung dedizierter IP-Adressen bedienen möchten, beachten Sie bitte folgende Hinweise:

- CloudFront Gibt Ihnen standardmäßig die Erlaubnis, zwei Zertifikate mit Ihrem AWS Konto zu verwenden, eines für den täglichen Gebrauch und eines für den Fall, dass Sie Zertifikate für mehrere Distributionen rotieren müssen.
- Wenn Sie mehr als zwei benutzerdefinierte SSL/TLS Zertifikate für Ihr AWS Konto benötigen, können Sie in der Service Quota-Konsole ein höheres Kontingent beantragen.

Verwenden Sie dasselbe Zertifikat für CloudFront Distributionen, die unter Verwendung verschiedener AWS Konten erstellt wurden

Wenn Sie eine Drittanbieter-Zertifizierungsstelle verwenden und dasselbe Zertifikat für mehrere CloudFront Distributionen verwenden möchten, die mit unterschiedlichen AWS Konten erstellt wurden, müssen Sie das Zertifikat für jedes Konto einmal in ACM importieren oder es einmal in den IAM-Zertifikatsspeicher hochladen. AWS

Wenn Sie von ACM bereitgestellte Zertifikate verwenden, können Sie nicht so konfigurieren, dass Zertifikate verwendet werdenCloudFront , die von einem anderen Konto erstellt wurden. AWS

Verwenden Sie dasselbe Zertifikat für CloudFront und für andere Dienste AWS

Wenn Sie ein Zertifikat von einer vertrauenswürdigen Zertifizierungsstelle wie Comodo oder Symantec gekauft haben, können Sie dasselbe Zertifikat für CloudFront und für andere AWS Dienste verwenden. DigiCert Wenn Sie das Zertifikat in ACM importieren, müssen Sie es nur einmal importieren, um es für mehrere AWS -Services verwenden zu können.

Wenn Sie von ACM bereitgestellte Zertifikate verwenden, werden die Zertifikate in ACM gespeichert.

Verwenden Sie dasselbe Zertifikat für mehrere Distributionen CloudFront

Sie können ein und dasselbe Zertifikat für beliebige oder alle CloudFront-Verteilungen nutzen, die Sie zur Bedienung von HTTPS-Anforderungen verwenden. Beachten Sie Folgendes:

- Sie können mit ein und demselben Zertifikat Anforderungen sowohl unter Verwendung dedizierter IP-Adressen als auch unter Verwendung von SNI bedienen.
- Sie können jeweils nur ein Zertifikat mit jeder Distribution verknüpfen.
- Jede Distribution muss einen oder mehrere alternative Domänennamen beinhalten, die auch im Feld Common Name (Allgemeiner Name) bzw. im Feld Subject Alternative Names (Alternative Subjektnamen) im Zertifikat angezeigt werden.
- Wenn Sie HTTPS-Anfragen mit dedizierten IP-Adressen bearbeiten und alle Ihre Distributionen mit demselben AWS Konto erstellt haben, können Sie Ihre Kosten erheblich senken, indem Sie dasselbe Zertifikat für alle Distributionen verwenden. CloudFront Gebühren für jedes Zertifikat, nicht für jede Verteilung.

  Angenommen, Sie erstellen drei Verteilungen mit demselben AWS Konto und verwenden dasselbe Zertifikat für alle drei Verteilungen. In diesem Fall würde Ihnen nur eine Gebühr für die Nutzung dedizierter IP-Adressen in Rechnung gestellt.

Wenn Sie HTTPS-Anfragen jedoch mit dedizierten IP-Adressen bearbeiten und dasselbe Zertifikat verwenden, um CloudFront Verteilungen in verschiedenen AWS Konten zu erstellen, wird jedem Konto die Gebühr für die Verwendung dedizierter IP-Adressen berechnet. Wenn Sie beispielsweise drei Distributionen mit drei verschiedenen AWS Konten erstellen und für alle drei Verteilungen dasselbe Zertifikat verwenden, wird jedem Konto die volle Gebühr für die Verwendung dedizierter IP-Adressen berechnet.

Konfigurieren alternativer Domainnamen und HTTPS

Um alternative Domainnamen in den URLs für Ihre Dateien zu verwenden und HTTPS zwischen Viewern und zu verwenden CloudFront, führen Sie die entsprechenden Verfahren aus.

Themen

- Holen Sie sich ein SSL/TLS Zertifikat
- Importieren Sie ein Zertifikat SSL/TLS
- Aktualisieren Sie Ihre CloudFront Distribution

Holen Sie sich ein SSL/TLS Zertifikat

Besorgen Sie sich ein SSL/TLS Zertifikat, falls Sie noch keines haben. Weitere Informationen finden Sie in der entsprechenden Dokumentation:

- Informationen zur Verwendung eines von AWS Certificate Manager (ACM) bereitgestellten Zertifikats finden Sie im AWS Certificate Manager Benutzerhandbuch. Fahren Sie anschließend fort mit der unter Aktualisieren Sie Ihre CloudFront Distribution beschriebenen Anleitung.

  > **Note**
  >
  > Es wird empfohlen, ACM für die Bereitstellung, Verwaltung und Bereitstellung von SSL/TLS Zertifikaten für AWS verwaltete Ressourcen zu verwenden. Sie müssen ein ACM-Zertifikat in der Region USA Ost (Nord-Virginia) anfordern.

- Wenn Sie ein Zertifikat von einer Zertifizierungsstelle (CA, Certificate Authority) eines Drittanbieters beziehen möchten, finden Sie entsprechende Informationen zur Vorgehensweise in der Dokumentation der Zertifizierungsstelle. Wenn Sie das Zertifikat haben, fahren Sie mit dem nächsten Schritt fort.

Importieren Sie ein Zertifikat SSL/TLS

Wenn Sie Ihr Zertifikat von einer Drittanbieter-Zertifizierungsstelle bezogen haben, importieren Sie das Zertifikat in ACM oder laden Sie es in den IAM-Zertifikatspeicher hoch:

ACM (empfohlen)

Mit ACM können Sie Zertifikate von Drittanbietern sowohl von der ACM-Konsole als auch programmgesteuert importieren. Informationen zum Importieren eines Zertifikats in ACM finden Sie unter Importieren von Zertifikaten in AWS Certificate Manager im AWS Certificate Manager - Benutzerhandbuch. Sie müssen das Zertifikat in der Region USA Ost (Nord-Virginia) importieren.

IAM-Zertifikatspeicher

(Nicht empfohlen) Verwenden Sie den folgenden AWS CLI Befehl, um Ihr Drittanbieter-Zertifikat in den IAM-Zertifikatsspeicher hochzuladen.

```
aws iam upload-server-certificate \
        --server-certificate-name CertificateName \
        --certificate-body file://public_key_certificate_file \
        --private-key file://privatekey.pem \
        --certificate-chain file://certificate_chain_file \
        --path /cloudfront/path/
```

Beachten Sie Folgendes:

- AWS Konto — Sie müssen das Zertifikat mit demselben AWS Konto, das Sie für die Erstellung Ihrer CloudFront Distribution verwendet haben, in den IAM-Zertifikatsspeicher hochladen.
- --path-Parameter – Wenn Sie Ihr Zertifikat in IAM hochladen, muss der Wert des Parameters `--path` (Zertifikatspfad) mit `/cloudfront/` beginnen, wie z. B. `/cloudfront/production/` oder `/cloudfront/test/`. Zudem muss der Pfad mit einem „/“ (Schrägstrich) enden.
- Vorhandene Zertifikate – Die Werte, die Sie für die Parameter `--server-certificate-name` und `--path` angeben, müssen sich von den entsprechenden Werten bereits vorhandener Zertifikate unterscheiden.
- Verwenden der CloudFront Konsole — Der Wert, den Sie beispielsweise für den `--server-certificate-name` Parameter in der angeben AWS CLI`myServerCertificate`, wird in der Liste der SSL-Zertifikate in der CloudFront Konsole angezeigt.
- Verwenden der CloudFront API — Notieren Sie sich die alphanumerische Zeichenfolge, die AWS CLI zurückgegeben wird, `AS1A2M3P4L5E67SIIXR3J` z. B. Dies ist der Wert, den Sie im

Element `IAMCertificateId` angeben müssen. Den ebenfalls von der CLI zurückgegebenen IAM-ARN können Sie ignorieren.

Weitere Informationen zu dem AWS CLI finden Sie im AWS Command Line Interface Benutzerhandbuch und in der AWS CLI Befehlsreferenz.

Aktualisieren Sie Ihre CloudFront Distribution

Verfahren Sie wie folgt, um Einstellungen für Ihre Distribution zu aktualisieren:

Um Ihren CloudFront Vertrieb für alternative Domainnamen zu konfigurieren

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home.
2. Wählen Sie die ID für die Distribution aus, die Sie aktualisieren möchten.
3. Wählen Sie auf der Registerkarte General die Option Edit aus.
4. Aktualisieren Sie die folgenden Werte:

   Alternativer Domainname (CNAME)

   Wählen Sie Objekt hinzufügen aus, um die entsprechenden alternativen Domainnamen hinzuzufügen. Trennen Sie einzelne Domain-Namen mit Kommas oder geben Sie jeden Domain-Namen in einer neuen Zeile ein.

   Benutzerdefiniertes SSL-Zertifikat

   Wählen Sie ein Zertifikat aus der Dropdown-Liste aus.

   Bis zu 100 Zertifikate sind hier aufgelistet. Bei mehr als 100 Zertifikaten geben Sie ggf. einen Zertifikat-ARN in das Feld ein, um das Zertifikat, das Sie hinzufügen möchten, auszuwählen.

   Wenn Sie ein Zertifikat in den IAM-Zertifikatspeicher hochgeladen haben, das Zertifikat in der Liste jedoch nicht angezeigt wird und Sie es durch Eingabe des Namens in dem Feld nicht auswählen können, überprüfen Sie anhand der unter Importieren Sie ein Zertifikat SSL/TLS beschriebenen Vorgehensweise, ob Sie das Zertifikat korrekt hochgeladen haben.

   > **Important**
   >
   > Nachdem Sie Ihr SSL/TLS Zertifikat Ihrer CloudFront Distribution zugeordnet haben, löschen Sie das Zertifikat erst aus ACM oder dem IAM-Zertifikatsspeicher, wenn

Sie das Zertifikat aus allen Distributionen entfernt haben und alle Distributionen bereitgestellt sind.

5. Wählen Sie Änderungen speichern aus.
6. Konfigurieren Sie so CloudFront , dass HTTPS zwischen Viewern und folgenden Geräten erforderlich ist: CloudFront

   a. Wählen Sie auf der Registerkarte Behaviors das zu aktualisierende Cache-Verhalten und anschließend Edit aus.

   b. Geben Sie einen der folgenden Werte für Viewer Protocol Policy an:

   Redirect HTTP to HTTPS

   Betrachter können beide Protokolle verwenden. HTTP-Anforderungen jedoch werden automatisch an HTTPS-Anforderungen umgeleitet. CloudFront gibt den HTTP-Statuscode `301 (Moved Permanently)` zusammen mit der neuen HTTPS-URL zurück. Der Viewer sendet die Anfrage dann erneut CloudFront über die HTTPS-URL.

   > **Important**
   >
   > CloudFront leitet`DELETE`,, `OPTIONS` `PATCH``POST`, oder `PUT` Anfragen nicht von HTTP zu HTTPS um. Wenn Sie ein Cacheverhalten so konfigurieren, dass es zu HTTPS umleitet, CloudFront reagiert auf HTTP `DELETE``OPTIONS`,`PATCH`,`POST`, oder `PUT` Anfragen für dieses Cacheverhalten mit einem HTTP-Statuscode`403 (Forbidden)`.

   Wenn ein Zuschauer eine HTTP-Anfrage stellt, die zu einer HTTPS-Anfrage umgeleitet wird, CloudFront fallen Gebühren für beide Anfragen an. Die Gebühren für die HTTP-Anforderung beziehen sich nur auf die Anforderung und die Header, die von CloudFront an den Viewer zurückgegeben werden. Bezüglich der HTTPS-Anforderung fallen Gebühren für die Anforderung sowie für die vom Ursprung zurückgegebenen Header und die Datei an, die vom Ursprung zurückgegeben wird.

   HTTPS Only

   Viewer können auf Ihre Inhalte nur zugreifen, wenn sie HTTPS verwenden. Wenn ein Zuschauer eine HTTP-Anfrage anstelle einer HTTPS-Anfrage sendet, CloudFront gibt er den HTTP-Statuscode zurück `403 (Forbidden)` und gibt die Datei nicht zurück.

c. Wählen Sie Yes, Edit aus.

d. Wiederholen Sie Schritt a bis c für jedes weitere Cache-Verhalten, das die Verwendung von HTTPS für die Kommunikation zwischen Viewern und CloudFront erzwingen soll.

7. Vergewissern Sie sich, dass folgende Punkte erfüllt sind, bevor Sie die aktualisierte Konfiguration in einer Produktionsumgebung verwenden:

   - Das Pfadmuster eines jeden Cache-Verhaltens wird nur auf die Anforderungen angewendet, für die die Viewer HTTPS verwenden sollen.
   - Die Cache-Verhaltensweisen sind in der Reihenfolge aufgeführt, in der sie von CloudFront ausgewertet werden sollen. Weitere Informationen finden Sie unter Pfadmuster.
   - Die Cache-Verhaltensweisen leiten Anforderungen an die richtigen Ursprünge weiter.

Ermitteln Sie die Größe des öffentlichen Schlüssels in einem SSL/TLS RSA-Zertifikat

Wenn Sie CloudFront alternative Domainnamen und HTTPS verwenden, beträgt die maximale Größe des öffentlichen Schlüssels in einem SSL/TLS RSA-Zertifikat 4096 Bit. (Dies bezieht sich auf die Schlüsselgröße, nicht auf die Anzahl der Zeichen in dem öffentlichen Schlüssel.) Wenn Sie AWS Certificate Manager für Ihre Zertifikate verwenden, unterstützt ACM zwar größere RSA-Schlüssel, Sie können die größeren Schlüssel jedoch nicht mit verwenden. CloudFront

Sie können die Länge des öffentlichen RSA-Schlüssels durch Ausführen des folgenden OpenSSL-Befehls ermitteln:

```
openssl x509 -in path and filename of SSL/TLS certificate -text -noout
```

Wobei gilt:

- `-in` gibt den Pfad und den Dateinamen Ihres SSL/TLS RSA-Zertifikats an.
- `-text` veranlasst, dass OpenSSL die Länge des öffentlichen RSA-Schlüssels in Bit anzeigt.
- `-noout` verhindert, dass OpenSSL den öffentlichen Schlüssel anzeigt.

Beispielausgabe:

```
Public-Key: (2048 bit)
```

Erhöhen Sie die Kontingente für Zertifikate SSL/TLS

Es gibt Kontingente für die Anzahl der SSL/TLS Zertifikate, die Sie in AWS Certificate Manager (ACM) importieren oder auf AWS Identity and Access Management (IAM) hochladen können. Es gibt auch ein Kontingent für die Anzahl der SSL/TLS Zertifikate, die Sie mit an verwenden können, AWS-Konto wenn Sie die Konfiguration so konfigurieren CloudFront , dass HTTPS-Anfragen mithilfe dedizierter IP-Adressen bedient werden. Sie haben jedoch die Möglichkeit, höhere Kontingente anzufordern.

Themen

- Erhöhen des Kontingents für in ACM importierte Zertifikate
- Erhöhen des Kontingents für in IAM hochgeladene Zertifikate
- Erhöhen des Kontingents für Zertifikate, die mit dedizierten IP-Adressen verwendet werden

Erhöhen des Kontingents für in ACM importierte Zertifikate

Informationen zum Kontingent bezüglich der Anzahl von Zertifikaten, die Sie in ACM importieren können, finden Sie unter Kontingente im AWS Certificate Manager -Benutzerhandbuch.

Verwenden Sie zum Anfordern einer Kontingenterhöhung die Service-Quotas-Konsole. Weitere Informationen finden Sie unter Beantragen einer Kontingenterhöhung im Service-Quotas-Benutzerhandbuch.

Erhöhen des Kontingents für in IAM hochgeladene Zertifikate

Informationen zum Kontingent (früher als Limit bezeichnet) für die Anzahl der Zertifikate, die Sie auf IAM hochladen können, finden Sie unter IAM- und STS-Limits im AM Benutzerhandbuch.

Verwenden Sie zum Anfordern einer Kontingenterhöhung die Service-Quotas-Konsole. Weitere Informationen finden Sie unter Beantragen einer Kontingenterhöhung im Service-Quotas-Benutzerhandbuch.

Erhöhen des Kontingents für Zertifikate, die mit dedizierten IP-Adressen verwendet werden

Das Kontingent für die Anzahl der SSL-Zertifikate, die Sie für jedes Zertifikat verwenden können, AWS-Konto wenn Sie HTTPS-Anfragen mit dedizierten IP-Adressen bedienen, finden Sie unterKontingente für SSL-Zertifikate.

Verwenden Sie zum Anfordern einer Kontingenterhöhung die Service-Quotas-Konsole. Weitere Informationen finden Sie unter Beantragen einer Kontingenterhöhung im Service-Quotas-Benutzerhandbuch.

SSL/TLS Zertifikate rotieren

Wenn Ihre SSL/TLS Zertifikate kurz vor dem Ablauf stehen, müssen Sie sie rotieren, um die Sicherheit Ihrer Verteilung zu gewährleisten und Serviceunterbrechungen für Ihre Zuschauer zu vermeiden. Sie können sie auf folgende Arten rotieren:

- Bei SSL/TLS Zertifikaten, die von AWS Certificate Manager (ACM) bereitgestellt werden, müssen Sie sie nicht rotieren. ACM übernimmt für Sie automatisch die Verwaltung der Zertifikatserneuerungen. Weitere Informationen finden Sie unter Verwaltete Zertifikaterneuerung im Benutzerhandbuch für AWS Certificate Manager .
- Wenn Sie Zertifikate von einer Drittanbieter-Zertifizierungsstelle beziehen und in ACM importieren (empfohlen) bzw. in den IAM-Zertifikatspeicher hochladen, müssen Sie gelegentlich ein Zertifikat durch ein anderes ersetzen.

> **Important**
>
> - ACM verwaltet keine Erneuerungen für Zertifikate, die Sie von Drittanbieter-Zertifizierungsstellen erwerben und in ACM importieren.
> - Wenn Sie CloudFront die Bearbeitung von HTTPS-Anfragen mithilfe dedizierter IP-Adressen konfiguriert haben, fällt möglicherweise eine zusätzliche, anteilige Gebühr für die Verwendung eines oder mehrerer zusätzlicher Zertifikate an, während Sie die Zertifikate rotieren. Für diesen Fall wird empfohlen, dass Sie Ihre Distributionen aktualisieren, um die zusätzliche Kosten zu minimieren.

Zertifikate rotieren SSL/TLS

Gehen Sie wie unten beschrieben vor, um Zertifikate zu rotieren. Viewer können weiterhin auf Ihre Inhalte zugreifen, während Sie Zertifikate rotieren, aber auch nachdem der Vorgang abgeschlossen ist.

Um SSL/TLS Zertifikate zu rotieren

1. Überprüfen Sie anhand des Abschnitts Erhöhen Sie die Kontingente für Zertifikate SSL/TLS, ob Sie eine Berechtigung zur Verwendung mehrerer SSL-Zertifikate benötigen. Wenn dies der Fall ist, beantragen Sie die Berechtigung und warten Sie, bis die Berechtigung erteilt ist, bevor Sie mit Schritt 2 fortfahren.
2. Importieren Sie das neue Zertifikat in ACM oder laden Sie es zu IAM hoch. Weitere Informationen finden Sie im CloudFront Amazon-Entwicklerhandbuch unter SSL/TLS Zertifikat importieren.
3. (Nur für IAM-Zertifikate) Aktualisieren Sie Ihre Distributionen, indem Sie nacheinander jede von ihnen für die Verwendung des neuen Zertifikats konfigurieren. Weitere Informationen finden Sie unter Eine Verteilung aktualisieren.
4. (Optional) Löschen Sie das vorherige Zertifikat aus ACM oder IAM.

> **Important**
>
> Löschen Sie ein SSL/TLS Zertifikat erst, wenn Sie es aus allen Distributionen entfernen und bis sich der Status der Distributionen, auf die Sie aktualisiert haben, geändert hat. `Deployed`

Kehren Sie von einem benutzerdefinierten SSL/TLS Zertifikat zum Standardzertifikat zurück CloudFront

Wenn Sie CloudFront die Verwendung von HTTPS zwischen Viewern und CloudFront konfiguriert haben und Sie die Verwendung eines benutzerdefinierten SSL/TLS Zertifikats konfiguriert CloudFront haben, können Sie Ihre Konfiguration so ändern, dass das CloudFront SSL/TLS Standardzertifikat verwendet wird. Die Vorgehensweise hängt davon ab, ob Sie Ihre Distribution bereits zur Distribution Ihrer Inhalte verwendet haben:

- Wenn Sie Ihre Distribution noch nicht zur Distribution Ihrer Inhalte genutzt haben, können Sie einfach die Konfiguration ändern. Weitere Informationen finden Sie unter Eine Verteilung aktualisieren.
- Wenn Sie Ihre Verteilung bereits zum Verteilen Ihrer Inhalte genutzt haben, müssen Sie eine neue CloudFront-Verteilung erstellen und die URLs für Ihre Dateien ändern, um Zeiten der

Nichtverfügbarkeit Ihrer Inhalte zu verkürzen oder ganz zu eliminieren. In diesem Fall verfahren Sie wie nachfolgend beschrieben.

Stellen Sie das Standardzertifikat wieder her CloudFront

Das folgende Verfahren zeigt Ihnen, wie Sie von einem benutzerdefinierten SSL/TLS Zertifikat zum CloudFront Standardzertifikat zurückkehren.

So kehren Sie zum Standardzertifikat zurück CloudFront

1. Erstellen Sie eine neue CloudFront Distribution mit der gewünschten Konfiguration. Wählen Sie für SSL Certificate die Option Default CloudFront Certificate (*.cloudfront.net) aus.

   Weitere Informationen finden Sie unter Eine Distribution erstellen.

2. Aktualisieren Sie für Dateien, die Sie verwenden CloudFront, die URLs in Ihrer Anwendung so, dass sie den Domainnamen verwenden, der der neuen Distribution CloudFront zugewiesen wurde. Sie können beispielsweise `https://www.example.com/images/logo.png` in `https://d111111abcdef8.cloudfront.net/images/logo.png` ändern.
3. Löschen Sie entweder die Distribution, die mit einem benutzerdefinierten SSL/TLS Zertifikat verknüpft ist, oder aktualisieren Sie die Distribution, um den Wert von SSL Certificate auf Default CloudFront Certificate (*.cloudfront.net) zu ändern. Weitere Informationen finden Sie unter Eine Verteilung aktualisieren.

   > **Important**
   >
   > Bis Sie diesen Schritt abgeschlossen haben, werden Ihnen AWS weiterhin Gebühren für die Verwendung eines benutzerdefinierten Zertifikats berechnet. SSL/TLS

4. (Optional) Löschen Sie Ihr benutzerdefiniertes SSL/TLS Zertifikat.

   a. Führen Sie den AWS CLI Befehl aus`list-server-certificates`, um die Zertifikat-ID des Zertifikats abzurufen, das Sie löschen möchten. Weitere Informationen finden Sie unter list-server-certificates in der AWS CLI -Befehlsreferenz.

   b. Führen Sie den AWS CLI Befehl aus`delete-server-certificate`, um das Zertifikat zu löschen. Weitere Informationen finden Sie unter delete-server-certificate in der AWS CLI -Befehlsreferenz.

Wechseln Sie von einem benutzerdefinierten SSL/TLS Zertifikat mit dedizierten IP-Adressen zu SNI

Wenn Sie CloudFront die Verwendung eines benutzerdefinierten SSL/TLS Zertifikats mit dedizierten IP-Adressen konfiguriert haben, können Sie stattdessen zur Verwendung eines benutzerdefinierten SSL/TLS Zertifikats mit SNI wechseln und die Gebühren für dedizierte IP-Adressen vermeiden.

> **Important**
>
> Diese Aktualisierung Ihrer CloudFront Konfiguration hat keine Auswirkungen auf Zuschauer, die SNI unterstützen. Zuschauer können vor und nach der Änderung sowie während der Übertragung der Änderung an CloudFront Edge-Standorten auf Ihre Inhalte zugreifen. Viewer, die SNI nicht unterstützen, können nach der Änderung nicht auf Ihre Inhalte zugreifen. Weitere Informationen finden Sie unter Wählen Sie aus, wie CloudFront HTTPS-Anfragen verarbeitet werden.

Wechsel von einem benutzerdefinierten Zertifikat zu SNI

Das folgende Verfahren zeigt Ihnen, wie Sie von einem benutzerdefinierten SSL/TLS Zertifikat mit dedizierten IP-Adressen zu SNI wechseln.

Um von einem benutzerdefinierten SSL/TLS Zertifikat mit dedizierten IP-Adressen zu SNI zu wechseln

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home.
2. Wählen Sie die ID der Distribution aus, die Sie anzeigen oder aktualisieren möchten.
3. Wählen Sie Distribution Settings aus.
4. Wählen Sie auf der Registerkarte General die Option Edit aus.
5. Deaktivieren Sie unter Benutzerdefinierte SSL-Zertifizierung – optional die Option Unterstützung für Legacy-Clients.
6. Wählen Sie Yes, Edit aus.

Gegenseitige TLS-Authentifizierung mit CloudFront (Viewer mTLS)

Mutual TLS Authentication (Mutual Transport Layer Security Authentication — mTLS) ist ein Sicherheitsprotokoll, das die standardmäßige TLS-Authentifizierung um eine bidirektionale zertifikatsbasierte Authentifizierung erweitert, bei der sowohl Client als auch Server ihre Identität nachweisen müssen, bevor eine sichere Verbindung hergestellt werden kann. Mithilfe von Mutual TLS können Sie sicherstellen, dass nur Clients, die vertrauenswürdige TLS-Zertifikate vorlegen, Zugriff auf Ihre Distributionen erhalten. CloudFront

Funktionsweise

Bei einem standardmäßigen TLS-Handshake legt nur der Server dem Client ein Zertifikat vor, um seine Identität nachzuweisen. Bei Mutual TLS wird der Authentifizierungsprozess bidirektional. Wenn ein Client versucht, eine Verbindung zu Ihrer CloudFront Distribution herzustellen, CloudFront fordert er während des TLS-Handshakes ein Client-Zertifikat an. Der Client muss ein gültiges X.509 Zertifikat vorlegen, das anhand Ihres konfigurierten Vertrauensspeichers CloudFront validiert wird, bevor die sichere Verbindung hergestellt wird.

CloudFront führt diese Zertifikatsvalidierung an AWS Edge-Standorten durch, wodurch die Komplexität der Authentifizierung von Ihren Ursprungsservern abgenommen wird und gleichzeitig die CloudFront globalen Leistungsvorteile erhalten bleiben. Sie können mTLS in drei Modi konfigurieren:

- Erforderlicher Modus (Standard) — CloudFront validiert das Client-Zertifikat anhand eines Vertrauensspeichers. Wenn die Überprüfung fehlschlägt oder kein Zertifikat vorgelegt wird, wird die CloudFront Verbindung verweigert. Verwenden Sie den Modus „Erforderlich“, wenn sich jeder Client mit einem gültigen Zertifikat authentifizieren muss.
- Optionaler Modus — CloudFront validiert das Client-Zertifikat, falls eines vorgelegt wird, erlaubt aber Verbindungen ohne Zertifikat. Zertifikatsmetadaten sind in den Verbindungsfunktionen und HTTP-Headern verfügbar, sodass Ihr Absender Autorisierungsentscheidungen treffen kann. Verwenden Sie den optionalen Modus, wenn Sie sowohl authentifizierte als auch nicht authentifizierte Clients unterstützen.
- Passthrough-Modus — validiert das Client-Zertifikat CloudFront nicht anhand eines Vertrauensspeichers. CloudFront überprüft nur, ob der Client den entsprechenden privaten Schlüssel besitzt. Es leitet das Zertifikat als HTTP-Header an Ihren Ursprung weiter, damit Ihr Ursprung die Validierung durchführen kann. Es ist kein Trust Store erforderlich und es findet kein Caching statt. Verwenden Sie den Passthrough-Modus, wenn Sie an Ihrem Ursprung bereits mTLS-Implementierungen haben.

Weitere Informationen zu diesen Validierungsmodi finden Sie unter. Zusätzliche Validierungsmodi

Anwendungsfälle

Gegenseitige TLS-Authentifizierung mit CloudFront Lösungen für mehrere kritische Sicherheitsszenarien, in denen herkömmliche Authentifizierungsmethoden nicht ausreichen:

- Geräteauthentifizierung mit Inhalts-Caching — Sie können Spielekonsolen, IoT-Geräte oder Unternehmenshardware authentifizieren, bevor Sie Zugriff auf Firmware-Updates, Spiele-Downloads oder interne Ressourcen gewähren. Jedes Gerät enthält ein einzigartiges Zertifikat, das seine Echtheit nachweist und gleichzeitig von Caching-Funktionen profitiert CloudFront .
- API-to-API Authentifizierung — Sie können die Kommunikation von Maschine zu Maschine zwischen vertrauenswürdigen Geschäftspartnern, Zahlungssystemen oder Mikrodiensten sichern. Certificate-based Die Authentifizierung macht gemeinsame Geheimnisse oder API-Schlüssel überflüssig und bietet gleichzeitig eine starke Identitätsprüfung für den automatisierten Datenaustausch.

Themen

- Vertrauensspeicher und Zertifikatsverwaltung
- Aktivieren Sie Mutual TLS für Distributionen CloudFront
- Ordnen Sie eine CloudFront Verbindungsfunktion zu
- Konfiguration zusätzlicher Einstellungen
- Zeigt die mTLS-Header für Cache-Richtlinien an und leitet sie an den Ursprung weiter
- Widerruf von Zertifikaten
- Hilfsmethoden für gegenseitiges TLS
- Zusätzliche Validierungsmodi
- Beobachtbarkeit mithilfe von Verbindungsprotokollen

Vertrauensspeicher und Zertifikatsverwaltung

Das Erstellen und Konfigurieren eines Trust Stores ist eine zwingende Voraussetzung für die Implementierung der gegenseitigen TLS-Authentifizierung mit CloudFront. Trust Stores enthalten die Zertifikate der Zertifizierungsstelle (CA), die zur Validierung von Client-Zertifikaten während des Authentifizierungsprozesses CloudFront verwendet werden.

Was ist ein Trust Store?

Ein Trust Store ist ein Repository mit CA-Zertifikaten, das zur Validierung von Client-Zertifikaten bei der gegenseitigen TLS-Authentifizierung CloudFront verwendet wird. Trust Stores enthalten die Stamm- und Zwischenzertifikate der Zertifizierungsstelle, die die Vertrauenskette für die Authentifizierung von Client-Zertifikaten bilden.

Wenn Sie Mutual TLS mit implementieren CloudFront, definiert der Trust Store, welchen Zertifizierungsstellen Sie vertrauen, dass sie gültige Client-Zertifikate ausstellen. CloudFront validiert jedes Client-Zertifikat während des TLS-Handshakes anhand Ihres Vertrauensspeichers. Nur Clients, die Zertifikate vorlegen, die mit einer der Zertifizierungsstellen in Ihrem Trust Store verknüpft sind, werden erfolgreich authentifiziert.

Trust Stores in CloudFront sind Ressourcen auf Kontoebene, die Sie mehreren Distributionen zuordnen können. Auf diese Weise können Sie konsistente Richtlinien zur Zertifikatsvalidierung in Ihrer gesamten CloudFront Bereitstellung aufrechterhalten und gleichzeitig die CA-Zertifikatsverwaltung vereinfachen.

Unterstützung durch Zertifizierungsstellen

CloudFront unterstützt Zertifikate, die sowohl von AWS privaten Zertifizierungsstellen als auch von privaten Zertifizierungsstellen von Drittanbietern ausgestellt wurden. Dank dieser Flexibilität können Sie Ihre bestehende Zertifikatsinfrastruktur verwenden oder AWS verwaltete Zertifikatsdienste nutzen, die Ihren organisatorischen Anforderungen entsprechen.

- AWS Private Zertifizierungsstelle: Sie können Zertifikate verwenden, die von AWS Private CA ausgestellt wurden. Private CA stellt einen verwalteten Dienst für private Zertifizierungsstellen bereit. Diese Integration vereinfacht die Verwaltung des Zertifikatslebenszyklus und bietet eine nahtlose Integration mit anderen AWS Diensten.
- Third-party private Zertifizierungsstellen: Sie können auch Zertifikate aus Ihrer bestehenden privaten Zertifizierungsstellen-Infrastruktur verwenden, einschließlich Unternehmenszertifizierungsstellen oder anderer Drittanbieter von Zertifikaten. Auf diese Weise können Sie Ihre aktuellen Prozesse zur Zertifikatsverwaltung beibehalten und gleichzeitig CloudFront mTLS-Funktionen hinzufügen.

Anforderungen und Spezifikationen für Zertifikate

Trust Stores haben spezielle Anforderungen an die CA-Zertifikate, die sie enthalten:

Anforderungen an das Format von CA-Zertifikaten

- Format: PEM-Format (Privacy Enhanced Mail)
- Inhaltsgrenzen: Zertifikate müssen innerhalb der Grenzen -----BEGIN CERTIFICATE----- und -----END CERTIFICATE----- liegen
- Kommentare: Dem muss ein #-Zeichen vorangestellt werden und es dürfen keine -Zeichen enthalten
- Zeilenumbrüche: Zwischen Zertifikaten sind keine Leerzeilen zulässig

Unterstützte Zertifikatsspezifikationen

- Art des Zertifikats: X.509v3
- Typen von öffentlichen Schlüsseln:
  - RSA 2048, RSA 3072, RSA 4096
  - ECDSA: secp256r1, secp384r1, secp521r1
- Signaturalgorithmen:
  - SHA256, SHA384, SHA512 mit RSA
  - SHA256, SHA384, SHA512 mit EC
  - SHA256, SHA384, SHA512 mit MGF1 RSASSA-PSS

Beispiel für ein Zertifikatspaketformat

Mehrere Zertifikate (PEM-encoded):

```
# Root CA Certificate
-----BEGIN CERTIFICATE-----
MIIDXTCCAkWgAwIBAgIJAKoK/OvD/XqiMA0GCSqGSIb3DQEBCwUAMEUxCzAJBgNV
BAYTAkFVMRMwEQYDVQQIDApTb21lLVN0YXRlMSEwHwYDVQQKDBhJbnRlcm5ldCBX
aWRnaXRzIFB0eSBMdGQwHhcNMTcwNzEyMTU0NzQ4WhcNMjcwNzEwMTU0NzQ4WjBF
MQswCQYDVQQGEwJBVTETMBEGA1UECAwKU29tZS1TdGF0ZTEhMB8GA1UECgwYSW50
ZXJuZXQgV2lkZ2l0cyBQdHkgTHRkMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAuuExKvY1xzHFylsHiuowqpmzs7rEcuuylOuEszpFp+BtXh0ZuEtts9LP
-----END CERTIFICATE-----
# Intermediate CA Certificate
-----BEGIN CERTIFICATE-----
MIIDXTCCAkWgAwIBAgIJAKoK/OvD/XqjMA0GCSqGSIb3DQEBCwUAMEUxCzAJBgNV
BAYTAkFVMRMwEQYDVQQIDApTb21lLVN0YXRlMSEwHwYDVQQKDBhJbnRlcm5ldCBX
```

```
aWRnaXRzIFB0eSBMdGQwHhcNMTcwNzEyMTU0NzQ4WhcNMjcwNzEwMTU0NzQ4WjBF
MQswCQYDVQQGEwJBVTETMBEGA1UECAwKU29tZS1TdGF0ZTEhMB8GA1UECgwYSW50
ZXJuZXQgV2lkZ2l0cyBQdHkgTHRkMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAuuExKvY1xzHFylsHiuowqpmzs7rEcuuylOuEszpFp+BtXh0ZuEtts9LP
-----END CERTIFICATE-----
```

Erstellen Sie einen Trust Store

Bevor Sie einen Trust Store erstellen, müssen Sie Ihr CA-Zertifikatspaket im PEM-Format in einen Amazon S3 S3-Bucket hochladen. Das Zertifikatspaket sollte alle vertrauenswürdigen Stamm- und Zwischenzertifikate der Zertifizierungsstelle enthalten, die zur Validierung Ihrer Client-Zertifikate erforderlich sind.

Das CA-Zertifikatspaket wird bei der Erstellung eines Vertrauensspeichers nur einmal aus S3 gelesen. Wenn future Änderungen am CA-Zertifikatspaket vorgenommen werden, muss der Trust Store manuell aktualisiert werden. Zwischen dem Trust Store und dem S3-CA-Zertifikatspaket wird keine Synchronisierung aufrechterhalten.

Voraussetzungen

- Ein Zertifikatspaket von Ihrer Zertifizierungsstelle (CA), das in einen Amazon S3 S3-Bucket hochgeladen wurde
- Die erforderlichen Berechtigungen zum Erstellen von CloudFront Ressourcen

Um einen Trust Store (Konsole) zu erstellen

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home.
2. Wählen Sie im Navigationsbereich Trust Stores aus.
3. Wählen Sie Trust Store erstellen aus.
4. Geben Sie unter Name des Vertrauensspeichers einen Namen für Ihren Vertrauensspeicher ein.
5. Geben Sie für das Zertifizierungsstellenpaket (CA) den Amazon S3-Pfad zu Ihrem PEM-format CA-Zertifikatspaket ein.
6. Wählen Sie Trust Store erstellen aus.

Um einen Trust Store zu erstellen (AWS CLI)

```
aws cloudfront create-trust-store \
```

```
  --name MyTrustStore \
  --ca-certificates-bundle-source '{"CaCertificatesBundleS3Location":{"Bucket":"my-
bucket","Key":"ca-bundle.pem","Region":"bucket-region"}}' \
  --tags Items=[{Key=Environment,Value=Production}]
```

Verknüpfen Sie den Trust Store mit Distributionen

Nachdem Sie einen Trust Store erstellt haben, müssen Sie ihn einer CloudFront Distribution zuordnen, um die gegenseitige TLS-Authentifizierung zu aktivieren.

Voraussetzungen

- Eine bestehende CloudFront Distribution mit aktivierter HTTPS-only Viewer-Protokollrichtlinie und deaktivierter HTTP3-Unterstützung.

Um einen Trust Store (Konsole) zuzuordnen

Es gibt zwei Möglichkeiten, einen Trust Store innerhalb der CloudFront Konsole zuzuordnen: über die Seite mit den Trust Store-Details oder über die Seite mit den Verteilungseinstellungen.

Zuordnen eines Trust Stores über die Seite mit den Trust Store-Details:

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home.
2. Wählen Sie im Navigationsbereich Trust Stores aus.
3. Wählen Sie den Namen des Trust Store aus, den Sie verknüpfen möchten.
4. Wählen Sie Mit Vertrieb verknüpfen aus.
5. Konfigurieren Sie die verfügbaren Viewer-MTLS-Optionen:
   - Validierungsmodus für Client-Zertifikate: Wählen Sie zwischen dem Modus „Erforderlich“ und „Optional“. Im Modus „Erforderlich“ müssen alle Clients Zertifikate vorlegen. Im optionalen Modus werden Clients, die Zertifikate vorlegen, validiert, während Clients, die keine Zertifikate vorlegen, Zugriff erhalten.
   - Vertrauensspeicher-CA-Namen bekannt geben: Wählen Sie aus, ob die CA-Namen in Ihrem Trust Store den Clients während des TLS-Handshakes bekannt gegeben werden sollen.
   - Ablaufdatum des Zertifikats ignorieren: Wählen Sie aus, ob Verbindungen mit abgelaufenen Zertifikaten zugelassen werden sollen (andere Validierungskriterien gelten weiterhin).

- Verbindungsfunktion: Eine optionale Verbindungsfunktion kann allow/deny Verbindungen auf der Grundlage anderer benutzerdefinierter Kriterien zugeordnet werden.

6. Wählen Sie eine oder mehrere Distributionen aus, die dem Trust Store zugeordnet werden sollen. Nur Distributionen mit deaktiviertem HTTP3 und HTTPS-only Cache-Verhalten können Viewer-MTLs unterstützen.
7. Wählen Sie Associate aus.

Zuordnen eines Trust Stores über die Seite mit den Verteilungseinstellungen:

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home.
2. Wählen Sie die Distribution aus, die Sie zuordnen möchten
3. Wählen Sie auf der Registerkarte Allgemein im Container Einstellungen in der oberen rechten Ecke die Option Bearbeiten
4. Scrollen Sie auf der Seite nach unten, schalten Sie im Connectivity-Container den Viewer-mTLS-Schalter ein
5. Konfigurieren Sie die verfügbaren Viewer-MTLS-Optionen:

   - Validierungsmodus für Client-Zertifikate: Wählen Sie zwischen dem Modus „Erforderlich“ und „Optional“. Im Modus „Erforderlich“ müssen alle Clients Zertifikate vorlegen. Im optionalen Modus werden Clients, die Zertifikate vorlegen, validiert, während Clients, die keine Zertifikate vorlegen, Zugriff erhalten.
   - Vertrauensspeicher-CA-Namen bekannt geben: Wählen Sie aus, ob die CA-Namen in Ihrem Trust Store den Clients während des TLS-Handshakes bekannt gegeben werden sollen.
   - Ablaufdatum des Zertifikats ignorieren: Wählen Sie aus, ob Verbindungen mit abgelaufenen Zertifikaten zugelassen werden sollen (andere Validierungskriterien gelten weiterhin).
   - Verbindungsfunktion: Eine optionale Verbindungsfunktion kann allow/deny Verbindungen auf der Grundlage anderer benutzerdefinierter Kriterien zugeordnet werden.

6. Wählen Sie in der unteren rechten Ecke die Option Änderungen speichern aus.

Um einen Trust Store zuzuordnen (AWS CLI)

Trust Stores können mithilfe der DistributionConfig.ViewerMtlsConfig Eigenschaft Distributionen zugeordnet werden. Das bedeutet, dass Sie zuerst die Distributionskonfiguration abrufen und diese dann ViewerMtlsConfig in einer nachfolgenden UpdateDistribution Anfrage angeben müssen.

```
// First fetch the distribution
aws cloudfront get-distribution {DISTRIBUTION_ID}

// Update the distribution config, for example:
Distribution config, file://distConf.json:
{
  ...other fields,
  ViewerMtlsConfig: {
    Mode: 'required',
    TrustStoreConfig: {
        AdvertiseTrustStoreCaNames: false,
        IgnoreCertificateExpiry: true,
        TrustStoreId: {TRUST_STORE_ID}
    }
  }
}

aws cloudfront update-distribution \
   --id {DISTRIBUTION_ID} \
   --if-match {ETAG} \
   --distribution-config file://distConf.json
```

Vertrauensspeicher verwalten

Details zum Trust Store anzeigen

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home.
2. Wählen Sie im Navigationsbereich Trust Stores aus.
3. Wählen Sie den Namen des Trust Store aus, um dessen Detailseite anzuzeigen.

Die Detailseite zeigt:

- Name und ID des Vertrauensspeichers
- Anzahl der CA-Zertifikate
- Erstellungsdatum und Datum der letzten Änderung
- Assoziierte Distributionen
- Tags (Markierungen)

Ändern Sie einen Trust Store

Um das CA-Zertifikatspaket zu ersetzen:

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home.
2. Wählen Sie im Navigationsbereich Trust Stores aus.
3. Wählen Sie den Namen des Trust Store aus.
4. Wählen Sie „Aktionen“ und dann „Bearbeiten“.
5. Geben Sie für das Zertifizierungsstellenpaket (CA) den Amazon S3 S3-Speicherort der aktualisierten CA-Bundle-PEM-Datei ein.
6. Wählen Sie Trust Store aktualisieren.

Löschen Sie einen Trust Store

Voraussetzungen: Sie müssen zuerst den Trust Store von allen CloudFront Distributionen trennen.

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CloudFront Konsole unter. https://console.aws.amazon.com/cloudfront/v4/home
2. Wählen Sie im Navigationsbereich Trust Stores aus.
3. Wählen Sie den Namen des Trust Store aus.
4. Wählen Sie Trust Store löschen.
5. Wählen Sie zur Bestätigung Delete.

Nächste Schritte

Nachdem Sie Ihren Trust Store erstellt und mit einer CloudFront Distribution verknüpft haben, können Sie damit fortfahren, die gegenseitige TLS-Authentifizierung auf Ihrer Distribution zu aktivieren und zusätzliche Einstellungen wie die Weiterleitung von Zertifikatsheadern an Ihre Ursprünge zu konfigurieren. Ausführliche Anweisungen zur Aktivierung von mTLS auf Ihren Distributionen finden Sie unter. Aktivieren Sie Mutual TLS für Distributionen CloudFront

Aktivieren Sie Mutual TLS für Distributionen CloudFront

Voraussetzungen und Anforderungen

CloudFront Im Modus der gegenseitigen TLS-Überprüfung müssen alle Clients während des TLS-Handshakes gültige Zertifikate vorlegen. Verbindungen ohne gültige Zertifikate werden abgewiesen. Bevor Sie Mutual TLS auf einer CloudFront Distribution aktivieren, stellen Sie sicher, dass Sie über Folgendes verfügen:

- Sie haben einen Trust Store mit Ihren Zertifizierungsstellenzertifikaten erstellt
- Hat den Trust Store mit Ihrer CloudFront Distribution verknüpft
- Es wurde sichergestellt, dass für alle Verhaltensweisen des Verteilungs-Caches eine HTTPS-only Viewer-Protokollrichtlinie verwendet wird
- Es wurde sichergestellt, dass Ihre Distribution verwendet HTTP/2 (in der Standardeinstellung wird Viewer-MTLS nicht unterstützt) HTTP/3

> **Note**
>
> Für die gegenseitige TLS-Authentifizierung sind HTTPS-Verbindungen zwischen Zuschauern und CloudFront erforderlich. Sie können mTLS nicht für eine Distribution aktivieren, deren Cache-Verhalten HTTP-Verbindungen unterstützt.

Aktivieren Sie Mutual TLS (Konsole)

Für neue Distributionen

Viewer-MTLS können beim Erstellen einer neuen Distribution in der CloudFront Konsole nicht konfiguriert werden. Erstellen Sie zuerst die Distribution mit beliebigen Mitteln (Konsole, CLI, API) und bearbeiten Sie dann die Verteilungseinstellungen, um Viewer-MTLS gemäß den nachstehenden Anweisungen für bestehende Distributionen zu aktivieren.

Für bestehende Distributionen

1. Melden Sie sich bei an AWS-Managementkonsole und öffnen Sie die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home.
2. Wählen Sie in der Verteilerliste die Distribution aus, die Sie ändern möchten.

3. Stellen Sie sicher, dass die Viewer-Protokollrichtlinie für alle Cache-Verhaltensweisen auf „HTTP zu HTTPS umleiten“ oder „Nur HTTPS“ eingestellt ist. (Sie können die Registerkarte Cache-Verhalten auswählen, um alle Cache-Verhaltensweisen mit HTTP-Protokollrichtlinien anzuzeigen und zu aktualisieren.)
4. Wählen Sie die Registerkarte Allgemein.
5. Wählen Sie im Abschnitt Settings (Einstellungen) die Option Edit (Bearbeiten) aus.
6. Suchen Sie im Bereich Konnektivität nach Viewer Mutual Authentication (mTLS).
7. Stellen Sie „Gegenseitige Authentifizierung aktivieren“ auf „Ein“.
8. Wählen Sie für den Validierungsmodus für Client-Zertifikate die Option Erforderlich (alle Clients müssen Zertifikate vorlegen) oder Optional (Clients können optional Zertifikate vorlegen) aus.
9. Wählen Sie für Trust Store Ihren zuvor erstellten Trust Store aus.
10. (Optional) Aktivieren Sie die Option Vertrauensspeicher-Zertifizierungsstellennamen ankündigen, wenn Sie während des TLS-Handshakes CA-Namen an Clients senden möchten CloudFront .
11. (Optional) Aktivieren Sie die Option Ablaufdatum des Zertifikats ignorieren, wenn Sie Verbindungen mit abgelaufenen Zertifikaten zulassen möchten.
12. Wählen Sie Änderungen speichern aus.

Gegenseitiges TLS aktivieren (AWS CLI)

Für neue Distributionen

Das folgende Beispiel zeigt, wie eine Distributions-Konfigurationsdatei (distribution-config.json) erstellt wird, die mTLS-Einstellungen enthält:

```
{
  "CallerReference": "cli-example-1",
  "Origins": {
    "Quantity": 1,
    "Items": [
      {
        "Id": "my-origin",
        "DomainName": "example.com",
        "CustomOriginConfig": {
          "HTTPPort": 80,
          "HTTPSPort": 443,
          "OriginProtocolPolicy": "https-only"
        }
```

```
      }
    ]
  },
  "DefaultCacheBehavior": {
    "TargetOriginId": "my-origin",
    "ViewerProtocolPolicy": "https-only",
    "MinTTL": 0,
    "ForwardedValues": {
      "QueryString": false,
      "Cookies": {
        "Forward": "none"
      }
    }
  },
  "ViewerCertificate": {
    "CloudFrontDefaultCertificate": true
  },
  "ViewerMtlsConfig": {
    "Mode": "required",
    "TrustStoreConfig": {
        "TrustStoreId": {TRUST_STORE_ID},
        "AdvertiseTrustStoreCaNames": true,
        "IgnoreCertificateExpiry": true
    }
  },
  "Enabled": true
}
```

Erstellen Sie die Distribution mit aktiviertem mTLS mithilfe des folgenden Beispielbefehls:

```
aws cloudfront create-distribution --distribution-config file://distribution-
config.json
```

Für bestehende Distributionen

Rufen Sie die aktuelle Distributionskonfiguration mit dem folgenden Beispielbefehl ab:

```
aws cloudfront get-distribution-config --id E1A2B3C4D5E6F7 --output json > dist-
config.json
```

Bearbeiten Sie die Datei, um mTLS-Einstellungen hinzuzufügen. Fügen Sie Ihrer Distributionskonfiguration den folgenden Beispielabschnitt hinzu:

```
"ViewerMtlsConfig": {
    "Mode": "required",
    "TrustStoreConfig": {
        "TrustStoreId": {TRUST_STORE_ID},
        "AdvertiseTrustStoreCaNames": true,
        "IgnoreCertificateExpiry": true
    }
}
```

Entfernen Sie das ETag-Feld aus der Datei, speichern Sie seinen Wert jedoch separat.

Aktualisieren Sie die Distribution mit der neuen Konfiguration mithilfe des folgenden Beispielbefehls:

```
aws cloudfront update-distribution \
    --id E1A2B3C4D5E6F7 \
    --if-match YOUR-ETAG-VALUE \
    --distribution-config file://dist-config.json
```

Viewer-Protokollrichtlinien

Bei Verwendung von Mutual TLS müssen alle Verhaltensweisen des Verteilungscaches mit einer HTTPS-only Viewer-Protokollrichtlinie konfiguriert werden:

- HTTP zu HTTPS umleiten — Leitet HTTP-Anfragen an HTTPS weiter, bevor die Zertifikatsvalidierung durchgeführt wird.
- Nur HTTPS — Nimmt nur HTTPS-Anfragen an und führt eine Zertifikatsvalidierung durch.

> **Note**
>
> Die HTTP- und HTTPS-Viewer-Protokollrichtlinie wird bei gegenseitigem TLS nicht unterstützt, da HTTP-Verbindungen keine Zertifikatsvalidierung durchführen können.

Nächste Schritte

Nachdem Sie Viewer TLS auf Ihrer CloudFront Distribution aktiviert haben, können Sie Verbindungsfunktionen zuordnen, um eine benutzerdefinierte Logik zur Zertifikatsvalidierung zu implementieren. Mit Verbindungsfunktionen können Sie die integrierten mTLS-

Authentifizierungsfunktionen um benutzerdefinierte Validierungsregeln, Zertifikatssperrprüfungen und Protokollierung erweitern. Einzelheiten zum Erstellen und Zuordnen von Verbindungsfunktionen finden Sie unter. Ordnen Sie eine CloudFront Verbindungsfunktion zu

Ordnen Sie eine CloudFront Verbindungsfunktion zu

CloudFront Mit Verbindungsfunktionen können Sie bei TLS-Handshakes eine benutzerdefinierte Logik zur Zertifikatsvalidierung implementieren und so die integrierten mTLS-Authentifizierungsfunktionen erweitern.

Was sind Verbindungsfunktionen?

Verbindungsfunktionen sind JavaScript Funktionen, die während des TLS-Handshakes ausgeführt werden, nachdem die Client-Zertifikate validiert wurden. Das validierte Client-Zertifikat wird an die Verbindungsfunktion übergeben. An diesem Punkt kann die Verbindungsfunktion eine zusätzliche Entscheidung darüber treffen, ob Zugriff gewährt werden soll oder nicht. Ausführliche Informationen zu Verbindungsfunktionen finden Sie unterMit CloudFront Verbindungsfunktionen anpassen.

So funktionieren Verbindungsfunktionen mit mTLS

Wenn ein Client versucht, eine mTLS-Verbindung zu Ihrer CloudFront Distribution herzustellen, erfolgt die folgende Reihenfolge:

1. Der Client initiiert den TLS-Handshake mit dem Edge-Standort CloudFront .
2. CloudFront fordert ein Client-Zertifikat an und empfängt es.
3. CloudFront führt die Standardvalidierung von Zertifikaten anhand des Trust Store durch.
4. Wenn das Zertifikat die Standardvalidierung bestanden hat, CloudFront ruft es Ihre Verbindungsfunktion auf. Wenn in Ihrem aktiviert IgnoreCertificateExpiryist ViewerMtlsConfig, werden Ihre abgelaufenen, aber ansonsten gültigen Zertifikate ebenfalls an die Verbindungsfunktion übergeben. Wenn die Client-Zertifikate ungültig sind, werden die Verbindungsfunktionen nicht aufgerufen.
5. Ihre Verbindungsfunktion empfängt analysierte Zertifikatsinformationen und Verbindungsdetails.
6. Ihre Funktion trifft eine allow/deny Entscheidung auf der Grundlage einer benutzerdefinierten Logik.
7. CloudFront vervollständigt oder beendet die TLS-Verbindung auf der Grundlage Ihrer Entscheidung.

Verbindungsfunktionen werden sowohl für den Überprüfungsmodus als auch für den optionalen Modus (wenn Clients Zertifikate vorlegen) aufgerufen.

Erstellen Sie eine Verbindungsfunktion

Sie können Verbindungsfunktionen mit der CloudFront Konsole oder der AWS CLI erstellen.

Um eine Verbindungsfunktion (Konsole) zu erstellen

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home.
2. Wählen Sie im Navigationsbereich Funktionen aus.
3. Wählen Sie die Registerkarte Verbindungsfunktionen und dann Verbindungsfunktion erstellen aus.
4. Geben Sie einen Funktionsnamen ein, der in Ihrem AWS Konto einzigartig ist.
5. Klicken Sie auf Weiter.
6. Schreiben Sie im Funktionseditor Ihren JavaScript Code für die Zertifikatsvalidierung. Der Funktionshandler muss entweder allow oder deny aufrufen.
7. Optional: Der Verbindungsfunktion kann ein KeyValue Speicher zugeordnet werden, um die Sperrkontrolle zu implementieren.
8. Wählen Sie Änderungen speichern aus.

Um eine Verbindungsfunktion zu erstellen (AWS CLI)

Das folgende Beispiel zeigt, wie eine Verbindungsfunktion erstellt wird:

Schreiben Sie Ihren Funktionscode in eine separate Datei, zum Beispiel code.js:

```
function connectionHandler(connection) {
  connection.allow();
}
```

```
aws cloudfront create-connection-function \
  --name "certificate-validator" \
  --connection-function-config '{
      "Comment": "Client certificate validation function",
      "Runtime": "cloudfront-js-2.0"
  }' \
```

```
  --connection-function-code fileb://code.js
```

Struktur des Verbindungsfunktionscodes

Verbindungsfunktionen implementieren die ConnectionHandler-Funktion, die ein Verbindungsobjekt empfängt, das Zertifikat und Verbindungsinformationen enthält. Ihre Funktion muss entweder `connection.allow()` oder verwenden`connection.deny()`, um eine Entscheidung über die Verbindung zu treffen.

Beispiel für eine grundlegende Verbindungsfunktion

Das folgende Beispiel zeigt eine einfache Verbindungsfunktion, die das Betrefffeld von Client-Zertifikaten überprüft:

```
function connectionHandler(connection) {
    // Only process if a certificate was presented
    if (!connection.clientCertificate) {
        console.log("No certificate presented");
        connection.deny();
    }

    // Check the subject field for specific organization
    const subject = connection.clientCertificate.certificates.leaf.subject;
    if (!subject.includes("O=ExampleCorp")) {
        console.log("Certificate not from authorized organization");
       connection.deny();
    } else {
        // All checks passed
        console.log("Certificate validation passed");
        connection.allow();
    }
}
```

Die vollständige Spezifikation der Eigenschaften von Client-Zertifikaten, die für das Verbindungsobjekt verfügbar sind, finden Sie hier:

```
{
  "connectionId": "Fdb-Eb7L9gVn2cFakz7wWyBJIDAD4-oNO6g8r3vXDV132BtnIVtqDA==", // Unique
 identifier for this TLS connection
  "clientIp": "203.0.113.42", // IP address of the connecting client (IPv4 or IPv6)
  "clientCertificate": {
    "certificates": {
```

```
    "leaf": {
      "serialNumber": "4a:3f:5c:92:d1:e8:7b:6c", // Unique serial number assigned by
the issuing CA (hexadecimal)
      "issuer": "CN=Example Corp Intermediate CA,O=Example Corp,C=US", //
Distinguished Name (DN) of the certificate authority that issued this certificate.
      "subject": "CN=client.example.com,O=Example Corp,C=US", // Distinguished Name
(DN) of the certificate holder.
      "validity": {
        "notBefore": "2024-01-15T00:00:00Z", // Certificate validity start date (ISO
8601 format)
        "notAfter": "2025-01-14T23:59:59Z"   // Certificate expiration date (ISO 8601
format)
      },
      "sha256Fingerprint": "a1b2c3d4e5f6...abc123def456", // SHA-256 hash of the
certificate (64 hex characters)
      "ocspEndpoint": "http://ocsp.example.com", // OCSP responder URL from the
certificate's AIA extension.
      "san": "DNS:client.example.com,DNS:api.example.com", // Subject Alternative
Names.
      "spki":
"30820122300d06092a864886f70d01010105000382010f003082010a0282010100...", // Subject
Public Key Info (DER-encoded, hex).
      "sanHex":
"3032820f636c69656e742e6578616d706c652e636f6d820f6170692e6578616d706c652e636f6d..." //
Subject Alternative Names (DER-encoded, hex).
    },
    "intermediates": [
      {
        "serialNumber": "00:d4:04:a5:fd:cb:20:37:54", // Serial number of the
intermediate certificate.
        "issuer": "CN=Example Corp Root CA,O=Example Corp,C=US", // Issuer of the
intermediate certificate.
        "subject": "CN=Example Corp Intermediate CA,O=Example Corp,C=US", // Subject
of the intermediate certificate.
        "validity": {
          "notBefore": "2020-01-01T00:00:00Z",
          "notAfter": "2030-01-01T23:59:59Z"
        },
        "sha256Fingerprint": "b2c3d4e5f6a1...def456abc123",
        "ocspEndpoint": "http://ocsp.example.com/intermediate",
        "san": "DNS:ca.example.com",
        "spki":
"30820122300d06092a864886f70d01010105000382010f003082010a0282010100b389dd513b...",
        "sanHex": "3015820d63612e6578616d706c652e636f6d..."
```

```
        }
      ]
    }
  }
}
```

Ordnen Sie eine Verbindungsfunktion zu

Nachdem Sie Ihre Verbindungsfunktion erstellt haben, müssen Sie sie in der LIVE-Phase veröffentlichen und sie Ihrer Distribution zuordnen.

Um eine Verbindungsfunktion (Konsole) zu veröffentlichen und zuzuordnen

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home.
2. Wählen Sie im Navigationsbereich Funktionen aus
3. Wählen Sie die Registerkarte Verbindungsfunktionen und wählen Sie Ihre Verbindungsfunktion aus.
4. Wählen Sie Veröffentlichen, um es in die LIVE-Phase zu verschieben.
5. Wählen Sie in der Tabelle mit den zugehörigen Verteilungen unter dem Veröffentlichungsbereich die Option Assoziation hinzufügen aus.
6. Wählen Sie die Distribution mit aktiviertem Viewer-MTLS aus, die Sie verknüpfen möchten.

Alternativ können veröffentlichte Verbindungsfunktionen auch über die Seite mit den Distributionsdetails verknüpft werden.

1. Navigieren Sie zur Startseite der Konsole, auf der all Ihre Distributionen aufgeführt sind.
2. Wählen Sie die Distribution aus, die Sie zuordnen möchten.
3. Wählen Sie die Registerkarte Allgemein.
4. Wählen Sie im Abschnitt Settings (Einstellungen) die Option Edit (Bearbeiten) aus.
5. Suchen Sie im Bereich Konnektivität nach Viewer Mutual Authentication (mTLS).
6. Wählen Sie unter Verbindungsfunktion Ihre Funktion aus.
7. Wählen Sie Änderungen speichern aus.

Um eine Verbindungsfunktion zuzuordnen (AWS CLI)

Das folgende Beispiel zeigt, wie eine Verbindungsfunktion mit einer Distribution verknüpft wird:

```
// DistributionConfig:
{
   ...other settings,
    "ConnectionFunctionAssociation": {
        "Id": "cf_30c2CV2elHwCoInb3LtcaUJkZeD"
    }
}
```

Anwendungsfälle für Verbindungsfunktionen

Verbindungsfunktionen ermöglichen mehrere erweiterte mTLS-Anwendungsfälle:

- Validierung von Zertifikatsattributen — Überprüfen Sie bestimmte Felder in Client-Zertifikaten, z. B. Anforderungen an Organisationseinheiten oder alternative Namensmuster für Antragsteller.
- Überprüfung der Sperrung von Zertifikaten — Implementieren Sie eine benutzerdefinierte Überprüfung der Sperrung von Zertifikaten KeyValueStore , indem Sie die Seriennummern von gesperrten Zertifikaten speichern.
- IP-based Zertifikatsrichtlinien — Wenden Sie je nach Client-IP-Adressen oder geografischen Einschränkungen unterschiedliche Zertifikatsrichtlinien an.
- Multi-tenant Validierung — Implementieren Sie mandantenspezifische Validierungsregeln, bei denen je nach Hostnamen oder Zertifikatattributen unterschiedliche Zertifikatsanforderungen gelten.

> **Note**
>
> Verbindungsfunktionen werden einmal pro Client-Verbindung während des TLS-Handshakes ausgeführt.
> Verbindungsfunktionen können nur Verbindungen zulassen oder verweigern, HTTP requests/responses nicht ändern.
> Nur Funktionen der LIVE-Phase (veröffentlicht) können Distributionen zugeordnet werden.
> Jede Distribution kann höchstens eine Verbindungsfunktion haben.

Nächste Schritte

Nachdem Sie Ihrer CloudFront Distribution eine Verbindungsfunktion zugeordnet haben, können Sie optionale Einstellungen konfigurieren, um das Verhalten Ihrer mTLS-Implementierung anzupassen. Ausführliche Anweisungen zur Konfiguration zusätzlicher Einstellungen, wie z. B. eines optionalen Validierungsmodus für Client-Zertifikate, finden Sie unter. Konfiguration zusätzlicher Einstellungen

Konfiguration zusätzlicher Einstellungen

Nachdem Sie die grundlegende gegenseitige TLS-Authentifizierung aktiviert haben, können Sie zusätzliche Einstellungen konfigurieren, um das Authentifizierungsverhalten an bestimmte Anwendungsfälle und Anforderungen anzupassen.

Werbung durch die Zertifizierungsstelle

Das AdvertiseTrustStoreCaNames Feld steuert, ob die Liste der vertrauenswürdigen Zertifizierungsstellennamen während des TLS-Handshakes an die Clients CloudFront gesendet wird, sodass die Clients das entsprechende Zertifikat auswählen können.

So konfigurieren Sie CA-Werbung (Konsole)

1. Navigieren Sie in Ihren Distributionseinstellungen zur Registerkarte Allgemein und wählen Sie Bearbeiten aus.
2. Scrollen Sie im Connectivity-Container zum Abschnitt Viewer Mutual Authentication (mTLS).
3. Aktivieren oder deaktivieren Sie das Kontrollkästchen Vertrauensspeicher-Zertifizierungsstellennamen ankündigen.
4. Wählen Sie Änderungen speichern aus.

Um CA-Werbung zu konfigurieren (AWS CLI)

Das folgende Beispiel zeigt, wie CA-Werbung aktiviert wird:

```
"ViewerMtlsConfig": {
   "Mode": "required", // or "optional"
   "TrustStoreConfig": {
      "AdvertiseTrustStoreCaNames": true,
      ...other settings
   }
}
```

Behandlung des Ablaufs von Zertifikaten

Die IgnoreCertificateExpiry Eigenschaft bestimmt, wie CloudFront auf abgelaufene Client-Zertifikate reagiert wird. CloudFront Lehnt standardmäßig abgelaufene Clientzertifikate ab, Sie können es jedoch so konfigurieren, dass sie bei Bedarf akzeptiert werden. Dies ist in der Regel für Geräte mit abgelaufenen Zertifikaten aktiviert, die nicht ohne Weiteres aktualisiert werden können.

So konfigurieren Sie die Behandlung von Zertifikatsablaufzeiten (Konsole)

1. Navigieren Sie in Ihren Verteilungseinstellungen zur Registerkarte Allgemein und wählen Sie Bearbeiten aus.
2. Scrollen Sie im Connectivity-Container zum Bereich Viewer Mutual Authentication (mTLS).
3. Aktivieren oder deaktivieren Sie das Kontrollkästchen Ablaufdatum des Zertifikats ignorieren.
4. Wählen Sie Änderungen speichern aus.

Um die Behandlung des Ablaufs von Zertifikaten zu konfigurieren (AWS CLI)

Das folgende Beispiel zeigt, wie der Ablauf eines Zertifikats ignoriert wird:

```
"ViewerMtlsConfig": {
  "Mode": "required", // or "optional"
  "TrustStoreConfig": {
     "IgnoreCertificateExpiry": false,
     ...other settings
  }
}
```

> **Note**
>
> IgnoreCertificateExpirygilt nur für die Gültigkeitsdaten der Zertifikate. Alle anderen Prüfungen zur Validierung von Zertifikaten gelten weiterhin (Vertrauenskette, Signaturvalidierung).

Nächste Schritte

Nachdem Sie zusätzliche Einstellungen konfiguriert haben, können Sie die Header-Weiterleitung einrichten, um Zertifikatsinformationen an Ihre Ursprünge weiterzuleiten, den Zertifikatswiderruf mithilfe von Verbindungsfunktionen und KeyValueStore implementieren und Verbindungsprotokolle

für die Überwachung aktivieren. Einzelheiten zur Weiterleitung von Zertifikatsinformationen an Ursprünge finden Sie unter Header an Ursprünge weiterleiten.

Zeigt die mTLS-Header für Cache-Richtlinien an und leitet sie an den Ursprung weiter

Wenn Sie die gegenseitige TLS-Authentifizierung verwenden, CloudFront können Sie Informationen aus Client-Zertifikaten extrahieren und sie als HTTP-Header an Ihre Ursprünge weiterleiten. Auf diese Weise können Ihre Ursprungsserver auf Zertifikatsdetails zugreifen, ohne eine Logik zur Zertifikatsvalidierung implementieren zu müssen.

Die folgenden Header stehen für die Erstellung von Cache-Verhalten zur Verfügung:

Header-Name	Description	Beispielwert	Unterstützte Modi
CloudFront-Viewer-Cert-Serial-Number	Hexadezimale Darstellung der Seriennummer des Zertifikats	4a:3f:5c:92:d1:e8:7 b:6c	Erforderlich, optional, Passthrough
CloudFront-Viewer-Cert-Issuer	RFC2253-Zeichenkettendarstellung des definierten Namens (DN) des Emittenten	cn=RootCamTLS.com, ou=RootCA, o=MTLS, L=Seattle, ST=Washington, C=US	Erforderlich, optional, Passthrough
CloudFront-Viewer-Cert-Subject	RFC2253-Zeichenkettendarstellung des definierten Namens (DN) des Betreffenden	CN=Client_.com, OU=Client-3, O=MTLS, ST=Washington, C=USA	Erforderlich, optional, Passthrough
CloudFront-Viewer-Cert-Present	Entweder 1 (vorhanden) oder 0 (nicht vorhanden), was angibt, ob das Zertifikat vorhanden ist. Dieser Wert ist im Modus Erforderlich immer 1.	1	Erforderlich, optional, Passthrough

Header-Name	Description	Beispielwert	Unterstützte Modi
CloudFront-Viewer-Cert-Sha256	Der SHA256-Hash des Client-Zertifikats	01fbf94fe f55697534 20c349f49 adbfd80af 527537781 6e3ab1fb3 71b29cb586	Erforderlich, optional, Passthrough

Für ursprüngliche Anfragen werden zusätzlich zu den oben genannten Headern, die für Cache-Verhalten zur Verfügung gestellt wurden, zwei zusätzliche Header bereitgestellt. Aufgrund der möglichen Header-Größe ist der CloudFront-Viewer-Cert-Pem Header keinen Edge-Funktionen (Lambda @Edge oder CloudFront Functions) ausgesetzt und wird nur an den Ursprung weitergeleitet.

Header-Name	Description	Beispielwert	Unterstützte Modi
CloudFront-Viewer-Cert-Validity	ISO8601-Format des NotBefore- und NotAfter-Datums	CloudFront-Viewer-Cert-Validity: =2023-09-21; =2024-09-20 NotBefore T01:50:17Z NotAfter T01:50:17Z	Erforderlich, optional, Passthrough
CloudFront-Viewer-Cert-Pem	URL-encoded PEM-Format des Blattzertifikats	CloudFront-Viewer-Cert-Pem: -----BEGIN%20CERTIFICATE----- %0AMIIG<... reduziert... : NmrUlw -----ANFANG%20ZERTIFIKAT-----	Erforderlich, optional

Header-Name	Description	Beispielwert	Unterstützte Modi
		%0AMIIG %0A-----ENDE%20ZERTIFIKAT-----%0A	
Client-Cert	Leaf-Zertifikat	Client-Cert: :MIIG<...reduziert... :NmrUlw :MIIG:	Pass-Through
Client-Cert-Chain	Non-leaf Kette von Zertifikaten	Client-Cert-Chain: :MIIG<...reduziert... :NmrUlw :MIIG:	Pass-Through

Konfiguration der Header-Weiterleitung

Konsole

Fügt im Überprüfungsmodus CloudFront automatisch die Header CloudFront-Viewer-Cert -* zu allen Viewer-Anfragen hinzu. Um diese Header an Ihren Ursprung weiterzuleiten:

1. Wählen Sie auf der Hauptseite der Listenverteilungen Ihre Distribution mit aktivierten Viewer-MTLS aus und wechseln Sie zum Tab Verhalten
2. Wählen Sie das Cache-Verhalten aus und klicken Sie auf Bearbeiten
3. Wähle im Abschnitt Origin-Anforderungsrichtlinie die Option Richtlinie erstellen oder wähle eine bestehende Richtlinie aus
4. Vergewissere dich, dass die folgenden Header in der Richtlinie für Origin-Anfragen enthalten sind:

    - CloudFront-Viewer-Cert-Serial-Number
    - CloudFront-Viewer-Cert-Issuer
    - CloudFront-Viewer-Cert-Subject
    - CloudFront-Viewer-Cert-Present
    - Cloudfront-Viewer-Cert-Sha256
    - CloudFront-Viewer-Cert-Validity

- CloudFront-Viewer-Cert-Pem

5. Wählen Sie Erstellen (für neue Richtlinien) oder Änderungen speichern (für bestehende Richtlinien)
6. Wählen Sie die Richtlinie in Ihrem Cache-Verhalten aus und speichern Sie die Änderungen

Verwenden AWS CLI

Das folgende Beispiel zeigt, wie Sie eine Origin-Anforderungsrichtlinie erstellen, die mTLS-Header für den Überprüfungsmodus enthält:

```
aws cloudfront create-origin-request-policy \
  --origin-request-policy-config '{
    "Name": "MTLSHeadersPolicy",
    "HeadersConfig": {
      "HeaderBehavior": "whitelist",
      "Headers": {
        "Quantity": 5,
        "Items": [
          "CloudFront-Viewer-Cert-Serial-Number",
          "CloudFront-Viewer-Cert-Issuer",
          "CloudFront-Viewer-Cert-Subject",
          "CloudFront-Viewer-Cert-Validity",
          "CloudFront-Viewer-Cert-Pem"
        ]
      }
    },
    "CookiesConfig": {
      "CookieBehavior": "none"
    },
    "QueryStringsConfig": {
      "QueryStringBehavior": "none"
    }
  }'
```

Überlegungen zur Header-Verarbeitung

Beachten Sie bei der Arbeit mit Zertifikatsheadern die folgenden bewährten Methoden:

- Header-Validierung: Überprüfen Sie als zusätzliche Sicherheitsmaßnahme die Zertifikat-Header-Werte an Ihrem Ursprung

- Größenbeschränkungen für Header: Die Header des PEM-Zertifikats können groß sein. Stellen Sie sicher, dass Ihr Ursprungsserver damit umgehen kann
- Überlegungen zum Cache: Die Verwendung von Zertifikatsheadern in Ihrem Cache-Schlüssel erhöht die Cache-Fragmentierung
- Cross-origin Anfragen: Wenn Ihre Anwendung CORS verwendet, müssen Sie sie möglicherweise so konfigurieren, dass die Zertifikatsheader zulässig sind

Nächste Schritte

Nachdem Sie die Header-Weiterleitung konfiguriert haben, können Sie die Überprüfung des Zertifikatswiderrufs mithilfe der CloudFront Verbindungsfunktionen und implementieren. KeyValueStore Einzelheiten zur Implementierung von Sperrprüfungen finden Sie unterWiderruf von Zertifikaten.

Widerruf von Zertifikaten

Wenn Zertifikate aufgrund von Kompromittierungen, Richtlinienänderungen oder beendeten Beziehungen gesperrt werden müssen, benötigen Sie einen Mechanismus, mit dem diese Zertifikate während des mTLS-Handshakes zurückgewiesen werden können. CloudFront bietet zwei systemeigene Ansätze für den Widerruf von Zertifikaten, die Sie kombinieren können, um eine mehrstufige Kontrolle zu erreichen.

- OCSP (Online Certificate Status Protocol) — CloudFront fragt den OCSP-Responder Ihrer Zertifizierungsstelle in Echtzeit ab, um zu überprüfen, ob ein Client-Zertifikat gesperrt wurde. Aktivieren Sie OCSP in Ihrem Trust Store und wickeln die Validierung während des CloudFront TLS-Handshakes automatisch ab. Die OCSP-Ergebnisse sind auch in den Verbindungsfunktionen sichtbar, sodass Sie programmgesteuert auf den Sperrstatus zugreifen können, um benutzerdefinierte Entscheidungen treffen zu können.
- CloudFront Funktionen und KeyValueStore — Sie verwalten eine Liste mit den Seriennummern für gesperrte Zertifikate in einem. CloudFront KeyValueStore Eine Verbindungsfunktion fragt die KeyValueStore während des TLS-Handshakes ab und erlaubt oder verweigert die Verbindung. Auf diese Weise haben Sie die volle Kontrolle über Sperrdaten, den Zeitpunkt der Aktualisierung und benutzerdefinierte Logik wie Kulanzfristen oder IP-based Ausnahmen.

Vergleich: OCSP versus CloudFront Funktionen mit KeyValueStore

	OCSP	CloudFront Funktionen + KeyValueStore
Datenquelle	Der OCSP-Responder der Zertifizierungsstelle	Sie verwalten die Sperrliste
Mechanismus aktualisieren	Real-time Anfrage an CA	Sie übertragen Updates an KeyValueStore
Benutzerdefinierte Logik	Verfügbar mithilfe von Verbindungsfunktionen	In Ihren Funktionscode integriert
Externe Abhängigkeit	Erfordert die Verfügbarkeit des CA OCSP-Responders	Keine externe Abhängigkeit
Am besten geeignet für	Zertifizierungsstellen, die OCSP-Responder verwalten; Status in Echtzeit CA-authoritative	Self-managed Widerruf; benutzerdefinierte Richtlinien; Zertifizierungsstellen ohne OCSP-Unterstützung

Sie können beide Ansätze zusammen verwenden. Aktivieren Sie OCSP für die CA-authoritative Sperrüberprüfung und verwenden Sie dann eine Verbindungsfunktion, um dem OCSP-Ergebnis zusätzliche Logik hinzuzufügen. So können Sie beispielsweise während einer Übergangszeit widerrufene Zertifikate aus vertrauenswürdigen IP-Bereichen zulassen.

OCSP (Online Certificate Status Protocol)

OCSP ist ein Echtzeitprotokoll, das den Sperrstatus eines Zertifikats direkt bei der Zertifizierungsstelle (CA) überprüft. Während des Signiervorgangs des Zertifikats bettet die CA eine OCSP-Responder-URL in das Zertifikat ein. Wenn ein Client sein Zertifikat während des mTLS-Handshakes vorlegt, eine OCSP-Anfrage an die eingebettete Responder-URL CloudFront sendet und auf die Antwort reagiert: Gültige Zertifikate werden fortgesetzt, widerrufene Zertifikate werden beendet.

CloudFront validiert die gesamte Zertifikatskette — das Leaf-Zertifikat und bis zu drei Zwischenzertifikate — jedes anhand der jeweiligen OCSP-Responder-URLs. Zertifikate im Trust Store werden als Vertrauensanker behandelt und sind von der OCSP-Validierung ausgeschlossen. Dies gilt für jedes Zertifikat im Trust Store, unabhängig davon, ob es sich

um eine Stammzertifizierungsstelle oder eine Zwischenzertifizierungsstelle handelt. Wenn Sie eine Zwischenzertifizierungsstelle in Ihrem Vertrauensspeicher platzieren, wird sie zu einem Vertrauensanker, und die Kettenvalidierung endet mit diesem Zertifikat. Nur die Zertifikate, die in der Kette darunter stehen, sind OCSP-validated es.

Aktivieren Sie OCSP

Aktivieren Sie OCSP in Ihrem Trust Store. Wenn diese Option aktiviert ist, CloudFront wird automatisch die OCSP-Validierung für jedes Client-Zertifikat durchgeführt, das eine OCSP-Responder-URL in der Erweiterung Authority Information Access (AIA) enthält. Sobald OCSP aktiviert ist, muss jedes Zertifikat in der Client-Zertifikatskette unterhalb des Vertrauensankers eine OCSP-URL haben. Wenn ein Zertifikat unter dem Vertrauensanker keine OCSP-URL enthält, CloudFront wird die Verbindung nicht hergestellt. Zertifikate im Trust Store (Trust Anchors) sind von dieser Anforderung ausgenommen.

CloudFront speichert OCSP-Antworten am Netzwerkrand im Cache, um die Round-Trip-Zeit zu reduzieren und vor Ausfallzeiten des OCSP-Responders zu schützen. OCSP-Antworten werden etwa 30 Minuten lang zwischengespeichert, und es kann bis zu 30 Minuten dauern, bis der aktualisierte Sperrstatus übernommen wird.

OCSP führt zu Verbindungsfunktionen

Wenn eine Verbindungsfunktion auf derselben Distribution konfiguriert ist, wird sie nach Abschluss der OCSP-Validierung CloudFront aufgerufen. Das Verbindungsobjekt enthält den OCSP-Status für die Blatt- und Zwischenzertifikate:

```
{
  "clientCertificate": {
    "certificates": {
      "leaf": {
        "subject": "CN=client.example.com, O=Example Org",
        "issuer": "CN=Intermediate CA, O=Example Org",
        "serialNumber": "00:a7:30:9e:73:7b:3e:63:bd:b7:c0:7e:bf:d5:c9:86",
        "validity": {
          "notBefore": "2024-01-01T00:00:00Z",
          "notAfter": "2025-01-01T00:00:00Z"
        },
        "sha256Fingerprint":
 "AB:CD:EF:12:34:56:78:90:AB:CD:EF:12:34:56:78:90:AB:CD:EF:12:34:56:78:90:AB:CD:EF:12:34:56:78:
        "ocspEndpoint": "http://ocsp.example.org"
      },
```

```
      "intermediates": [
        {
          "subject": "CN=Intermediate CA, O=Example Org",
          "issuer": "CN=Root CA, O=Example Org",
          "serialNumber": "00:a7:30:9e:73:7b:3e:63:bd:b7:c0:7e:bf:d5:c9:86",
          "validity": {
            "notBefore": "2020-01-01T00:00:00Z",
            "notAfter": "2030-01-01T00:00:00Z"
          },
          "sha256Fingerprint":
 "12:34:56:78:90:AB:CD:EF:12:34:56:78:90:AB:CD:EF:12:34:56:78:90:AB:CD:EF:12:34:56:78:90:AB:CD:
          "ocspEndpoint": "http://ocsp.example.org"
        }
      ]
    },
    "revocationStatus": {
      "chainValidity": "Valid",           // "Valid" | "Invalid" | "Unknown"
      "certificates": {
        "leaf": {
          "method": "OCSP",               // "OCSP"
          "status": "Good",               // "Good" | "Revoked" | "Unknown" | "Error"
          "serialNumber": "00:a7:30:9e:73:7b:3e:63:bd:b7:c0:7e:bf:d5:c9:86"
        },
        "intermediates": [
          {
            "method": "OCSP",             // "OCSP"
            "status": "Error",            // "Good" | "Revoked" | "Unknown" | "Error"
            "errorType": "InternalError", // "InternalError" | "OCSP response
 verification failed: {Type}"
            "serialNumber": "00:a7:30:9e:73:7b:3e:63:bd:b7:c0:7e:bf:d5:c9:86"
          }
        ]
      }
    }
  },
  "clientIp":"127.0.0.1",
  "endpoint":"d123.cloudfront.net",
  "distributionId":"E1NXS4MQZH501R",
  "connectionId":"xdzQ6lJUDUt8b7OuqOD8lmzOC9HcMaXPmhH5ZdzLCZpKxqzfCPpR4A=="
}
```

Das `chainValidity` Feld kann `Valid`Invalid, oder Unknown sein. Einzelne `status` Zertifikatswerte könnenGood, `Revoked`Unknown, oder sein`Error`. Das `errorType` Feld enthält

`InternalError` oder`OCSP response verification failed: {Type}`, wenn der Status lautet`Error`.

Die Verbindungsfunktion kann das OCSP-Ergebnis außer Kraft setzen. Sie kann beispielsweise ein widerrufenes Zertifikat aus einem vertrauenswürdigen IP-Bereich zulassen oder ein Zertifikat verweigern, das OCSP aufgrund zusätzlicher Geschäftslogik als „gut“ meldet.

> **Note**
>
> Das `errorType` Feld ist nur vorhanden, wenn der Status lautet. `Error`

Beispiel: Benutzerdefinierte OCSP-Behandlung mit vertrauenswürdiger IP-Ausnahme

```
function connectionHandler(connection) {
    var revocationStatus = connection.clientCertificate.revocationStatus;
    var trustedIP = (connection.clientIp === "[IP_ADDRESS]");

    if (revocationStatus.chainValidity === "Invalid") {
        if (trustedIP) {
            connection.allow();
        } else {
            connection.deny();
        }
    } else if (revocationStatus.certificates.leaf.status === "Error") {
        console.log(revocationStatus.certificates.leaf.errorType);
        connection.deny();
    } else {
        connection.allow();
    }
}
```

OCSP-Fehlerverhalten

Wenn der Sperrstatus eines Zertifikats CloudFront nicht ermittelt werden kann — weil der OCSP-Responder nicht erreichbar ist, einen Fehler zurückgibt oder „unbekannt“ zurückgibt — wird die Verbindung standardmäßig CloudFront verweigert. Verwenden Sie eine Verbindungsfunktion, um ein Soft-Fail-Verhalten zu implementieren. Verbindungsfunktionen werden nicht ausgeführt, wenn OCSP aktiviert ist und die OCSP-URL in den Client-Zertifikaten fehlt. Das OCSP-Ergebnis ist im Verbindungsobjekt verfügbar, und Ihre Funktion kann Verbindungen zulassen, wenn der OCSP-Status unbestimmt ist:

```
async function connectionHandler(connection) {
    var revocationStatus = connection.clientCertificate.revocationStatus;

    if (revocationStatus.certificates.leaf.status === "Error" ||
        revocationStatus.certificates.leaf.status === "Unknown") {
        // OCSP responder unreachable — allow connection (soft-fail)
        connection.logCustomData(`OCSP_SOFT_FAIL:
${revocationStatus.certificates.leaf.errorType}`);
        return connection.allow();
    }

    if (revocationStatus.chainValidity === "Invalid") {
        return connection.deny();
    }

    return connection.allow();
}
```

Widerruf von Zertifikaten mit Funktionen und CloudFront KeyValueStore

Sie können CloudFront Verbindungsfunktionen mit verwenden KeyValueStore , um die Überprüfung des Zertifikatswiderrufs ohne externe Abhängigkeit zu implementieren. Sie verwalten eine Liste mit Seriennummern für gesperrte Zertifikate in a KeyValueStore, und Ihre Verbindungsfunktion überprüft jedes Client-Zertifikat während des TLS-Handshakes anhand dieser Liste.

Der Prozess zum Widerruf von Zertifikaten funktioniert wie folgt:

1. Speichern Sie die Seriennummern widerrufener Zertifikate in einem CloudFront KeyValueStore.
2. Wenn ein Client ein Zertifikat vorlegt, wird Ihre Verbindungsfunktion aufgerufen.
3. Die Funktion vergleicht die Seriennummer des Zertifikats mit der KeyValueStore.
4. Wenn die Seriennummer im Speicher gefunden wird, wird das Zertifikat gesperrt.
5. Ihre Funktion verweigert die Verbindung für gesperrte Zertifikate.

Dieser Ansatz ermöglicht eine Sperrüberprüfung nahezu in Echtzeit im gesamten CloudFront globalen Edge-Netzwerk.

Um diesen Ansatz zu implementieren, benötigen Sie:

- Eine mit Viewer-MTLS konfigurierte Distribution

- A, das KeyValueStore die Seriennummern des gesperrten Zertifikats enthält
- Eine Verbindungsfunktion, die abfragt, KeyValueStore um den Zertifikatsstatus zu überprüfen

Wenn ein Client eine Verbindung herstellt, CloudFront validiert er das Zertifikat anhand des Vertrauensspeichers und führt dann Ihre Verbindungsfunktion aus. Ihre Funktion vergleicht die Seriennummer des Zertifikats mit der KeyValueStore und erlaubt oder verweigert die Verbindung.

Schritt 1: Erstellen Sie eine KeyValueStore für widerrufene Zertifikate

Bereiten Sie die Seriennummern Ihrer widerrufenen Zertifikate im JSON-Format vor:

```
{
  "data": [
    { "key": "ABC123DEF456", "value": "" },
    { "key": "789XYZ012GHI", "value": "" }
  ]
}
```

Laden Sie diese JSON-Datei in einen S3-Bucket hoch und erstellen Sie dann: KeyValueStore

```
aws s3 cp revoked-serials.json s3://your-bucket-name/revoked-serials.json

aws cloudfront create-key-value-store \
  --name revoked-serials-kvs \
  --import-source '{
    "SourceType": "S3",
    "SourceARN": "arn:aws:s3:::your-bucket-name/revoked-serials.json"
  }'
```

Warten Sie KeyValueStore , bis die Bereitstellung abgeschlossen ist. Überprüfen Sie den Status mit:

```
aws cloudfront get-key-value-store --name "revoked-serials-kvs"
```

Schritt 2: Erstellen Sie die Sperrverbindungsfunktion

Erstellen Sie eine Verbindungsfunktion, die die Seriennummern von Zertifikaten anhand folgender Werte überprüft KeyValueStore:

```
aws cloudfront create-connection-function \
  --name "revocation-control" \
```

```
  --connection-function-config file://connection-function-config.json \
  --connection-function-code file://connection-function-code.txt
```

Die Konfigurationsdatei spezifiziert die KeyValueStore Zuordnung:

```
{
  "Runtime": "cloudfront-js-2.0",
  "Comment": "A function that implements revocation control via KVS",
  "KeyValueStoreAssociations": {
    "Quantity": 1,
    "Items": [
      {
        "KeyValueStoreArn": "arn:aws:cloudfront::account-id:key-value-store/kvs-id"
      }
    ]
  }
}
```

Beispiel für einen Verbindungsfunktionscode:

```
import cf from 'cloudfront';

async function connectionHandler(connection) {
    const kvsHandle = cf.kvs();

    // Get client serial number from client certificate
    const clientSerialNumber =
        connection.clientCertificate.certificates.leaf.serialNumber;

    // Check KVS to see if serial number exists as a key
    // Remove : from the clientSerialNumber if KVS entries dont have it
    const serialNumberExistsInKvs =
        await kvsHandle.exists(clientSerialNumber.replaceAll(":", ""));

    // Deny connection if serial number exists in KVS
    if (serialNumberExistsInKvs) {
        console.log("Connection denied – certificate revoked");
        connection.logCustomData("Connection denied – certificate revoked");
        return connection.deny();
    }

    // Allow connections that don't exist in KVS
    console.log("Connection allowed");
```

```
    return connection.allow();
}
```

Schritt 3: Testen Sie Ihre Sperrfunktion

Verwenden Sie die CloudFront Konsole, um Ihre Verbindungsfunktion mit Beispielzertifikaten zu testen. Navigieren Sie in der Konsole zur Verbindungsfunktion und verwenden Sie die Registerkarte Test.

- Fügen Sie ein Beispielzertifikat im PEM-Format in die Testoberfläche ein.
- Geben Sie optional eine Client-IP-Adresse für die IP-based Testlogik an.
- Wählen Sie Testfunktion, um die Ausführungsergebnisse zu sehen.
- Überprüfen Sie die Ausführungsprotokolle, um Ihre Funktionslogik zu überprüfen.

Testen Sie sowohl mit gültigen als auch mit gesperrten Zertifikaten, um sicherzustellen, dass Ihre Funktion beide Szenarien korrekt verarbeitet.

Schritt 4: Ordnen Sie die Funktion Ihrer Distribution zu

Nachdem Sie Ihre Verbindungsfunktion veröffentlicht haben, verknüpfen Sie sie mit Ihrer TLS-enabled M-Distribution, um die Überprüfung des Zertifikatswiderrufs zu aktivieren. Navigieren Sie zu Ihren Distributionseinstellungen, scrollen Sie zum Abschnitt „Viewer Mutual Authentication (mTLS)“, wählen Sie Ihre Verbindungsfunktion aus und speichern Sie die Änderungen.

Fortgeschrittene Widerrufsstrategien

Kombinieren Sie OCSP mit der Logik der Verbindungsfunktionen

Sie können OCSP für die CA-authoritative Sperrüberprüfung aktivieren und eine Verbindungsfunktion für benutzerdefinierte Richtlinien hinzufügen. Die Verbindungsfunktion empfängt das OCSP-Ergebnis und kann zusätzliche Logik anwenden:

- Übergangsfristen — Erlauben Sie während der Zertifikatsrotation für einen bestimmten Zeitraum widerrufene Zertifikate aus internen Netzwerken.
- Notfallzugriff — Erlaubt Verbindungen von bestimmten IP-Adressen aus, auch wenn OCSP den Status „Widerrufen“ meldet.
- Benutzerdefinierte Ablehnungslogik — Blockieren Sie Zertifikate, die OCSP auf der Grundlage Ihrer eigenen Sperrdaten in als „gut“ meldet. KeyValueStore

Example Kombinierter Widerruf durch OCSP und KVS

```
import cf from 'cloudfront';

async function connectionHandler(connection) {
    var kvsHandle = cf.kvs();
    var revocationStatus = connection.clientCertificate.revocationStatus;
    var serialNumber = connection.clientCertificate.certificates.leaf.serialNumber;

    // Check your own revocation list first (immediate revocation, no cache delay)
    var inKvs = await kvsHandle.exists(serialNumber.replaceAll(":", ""));
    if (inKvs) {
        connection.logCustomData("KVS_REVOKED:" + serialNumber);
        return connection.deny();
    }

    // Then check OCSP result
    if (revocationStatus.chainValidity === 'Valid' &&
 revocationStatus.certificates.leaf.status === "Revoked") {
        // OCSP says revoked — allow grace period from trusted IPs
        if (connection.clientIp.startsWith("10.0.")) {
            connection.logCustomData("GRACE_PERIOD:" + serialNumber + ":" +
 connection.clientIp);
            return connection.allow();
        }
        connection.logCustomData("OCSP_REVOKED:" + serialNumber);
        return connection.deny();
    }

    connection.allow();
}
```

Dieses Muster ermöglicht den sofortigen Widerruf mithilfe von KVS (ohne Cache-Verzögerung) sowie den CA-authoritative Widerruf mithilfe von OCSP, mit einer benutzerdefinierten Ausnahmebehandlung dazwischen.

Hilfsmethoden für gegenseitiges TLS

Verwenden Sie die `cf.mtls` Hilfsfunktionen in den Viewer-Anforderungsfunktionen, um die Header von Client-Zertifikaten umzubenennen, neu zu formatieren oder zu kombinieren, bevor Sie die Anfrage an Ihren Ursprung weiterleiten. Details hierzu finden Sie unter the section called “Hilfsmethoden für gegenseitiges TLS”.

Zusätzliche Validierungsmodi

Der Modus „Erforderlich“ ist die Standardeinstellung für CloudFront gegenseitiges TLS. Er CloudFront validiert jedes Client-Zertifikat und verweigert fehlgeschlagene Verbindungen. Zur Unterstützung zusätzlicher Anwendungsfälle — z. B. Anwendungen, die eine Mischung aus M TLS-authenticated und nicht authentifizierten Clients bedienen, oder Umgebungen, in denen der Ursprungsserver seine eigene mTLS-Validierung durchführt — CloudFront werden jedoch zwei zusätzliche Modi unterstützt.

Der optionale Modus ist für Anwendungen mit gemischten Kundengruppen konzipiert. Zum Beispiel ein Portal, das öffentliche Inhalte für nicht authentifizierte Browser bereitstellt und gleichzeitig API-Endpunkte auf Kunden beschränkt, die gültige Zertifikate vorlegen. Oder ein Migrationsszenario, in dem Sie Kunden schrittweise in mTLS einbinden und während der Umstellung sowohl authentifizierte als auch nicht authentifizierte Verbindungen zulassen müssen.

Der Passthrough-Modus ist für Kunden konzipiert, deren Ursprungsserver bereits eine mTLS-Validierung durchführen. Zum Beispiel Dienste, die von lokalen Reverse-Proxys, anderen CDNs oder Application Load Balancern migriert werden, die bereits die Zertifikatsverifizierung durchführen. Im Passthrough-Modus können diese Kunden ihre Anwendungen direkt vor ihren Anwendungen platzieren, ohne die CloudFront Validierungslogik am Edge erneut implementieren zu müssen.

Optionaler Modus

Validierung des Client-Zertifikats Optionaler Modus

CloudFront bietet einen alternativen optionalen Validierungsmodus für Clientzertifikate, der vorgelegte Client-Zertifikate validiert, aber den Zugriff auf Clients ermöglicht, die keine Zertifikate vorlegen.

Verhalten im optionalen Modus

- Gewährt Clients mit gültigen Zertifikaten eine Verbindung (ungültige Zertifikate werden verweigert).
- Ermöglicht die Verbindung zu Clients ohne Zertifikate.
- Ermöglicht gemischte Client-Authentifizierungsszenarien über eine einzige Distribution.

Der optionale Modus ist ideal für die schrittweise Migration zur mTLS-Authentifizierung. Er unterstützt Clients mit Zertifikaten und Clients ohne Zertifikate oder gewährleistet die Abwärtskompatibilität mit älteren Clients.

> **Note**
>
> Im optionalen Modus werden die Verbindungsfunktionen auch dann aufgerufen, wenn die Clients keine Zertifikate vorlegen. Auf diese Weise können Sie benutzerdefinierte Logik implementieren, z. B. das Protokollieren von Client-IP-Adressen oder das Anwenden verschiedener Richtlinien, je nachdem, ob Zertifikate vorgelegt werden.

So konfigurieren Sie den optionalen Modus (Konsole)

1. Navigieren Sie in Ihren Distributionseinstellungen zur Registerkarte Allgemein und wählen Sie Bearbeiten aus.
2. Scrollen Sie im Connectivity-Container zum Abschnitt Viewer Mutual Authentication (mTLS).
3. Wählen Sie für den Validierungsmodus für das Client-Zertifikat die Option Optional aus.
4. Speichern Sie die Änderungen.

Um den optionalen Modus zu konfigurieren (AWS CLI)

Das folgende Beispiel zeigt, wie der optionale Modus konfiguriert wird:

```
"ViewerMtlsConfig": {
    "Mode": "optional",
    ...other settings
}
```

Passen Sie die Zertifikats-Header an

Verwenden Sie die mTLS-Hilfsfunktionen in CloudFront Viewer-Request-Funktionen, um die Zertifikatsheader umzubenennen, neu zu formatieren oder zu kombinieren, bevor sie Ihren Ursprung erreichen. Dies ist nützlich, wenn Sie von anderen Diensten migrieren, die andere Header-Namen oder Verschlüsselungsformate für Zertifikate verwenden.

Passthrough-Modus

Den Passthrough-Modus können Sie CloudFront mit Ihren vorhandenen mTLS-Implementierungen an Ihrem Ursprung verwenden. CloudFront beendet die TLS-Verbindung und leitet das Client-Zertifikat als HTTP-Header an Ihren Ursprung weiter. Ihr Ursprung führt die gesamte

Zertifikatsvalidierung durch — einschließlich Kettenüberprüfung, Sperrüberprüfung und Durchsetzung benutzerdefinierter Richtlinien.

So funktioniert der Passthrough-Modus

1. Der Client stellt während des TLS-Handshakes eine Verbindung zum Client her CloudFront und präsentiert ein Client-Zertifikat.
2. CloudFront schließt den TLS-Handshake ab, ohne das Zertifikat anhand eines Trust Stores zu validieren.
3. CloudFront fügt das Client-Zertifikat und die Zertifikatskette der Anfrage als HTTP-Header hinzu.
4. Die Anfrage wird an Ihren Ursprung weitergeleitet. Es wird kein Inhalt zwischengespeichert.
5. Ihr Ursprung validiert das Zertifikat und verarbeitet die Anfrage.

Kunden können sich auch ohne Vorlage eines Zertifikats verbinden. Ihre Ursprungs- oder Verbindungsfunktion verarbeitet Szenarien mit leeren Zertifikaten.

> **Note**
>
> Im Passthrough-Modus werden die Verbindungsfunktionen auch dann aufgerufen, wenn die Clients keine Zertifikate vorlegen. Auf diese Weise können Sie benutzerdefinierte Logik implementieren, z. B. das Protokollieren von Client-IP-Adressen oder das Anwenden verschiedener Richtlinien, je nachdem, ob Zertifikate vorgelegt werden.

Konfigurationsanforderungen

- Kein Vertrauensspeicher — Die Verteilung darf keine Trust-Store-Verknüpfungen aufweisen.
- Caching deaktiviert — Für alle Cache-Verhaltensweisen muss die Richtlinie für verwaltete `CachingDisabled` Caches verwendet werden.
- Origin Shield verboten — Origin Shield kann nicht aktiviert werden.
- Lambda @Edge verboten — Lambda @Edge -Funktionszuordnungen sind nicht zulässig.
- Quellanforderungsrichtlinie erforderlich — Sie müssen die `Client-Cert-Chain` Header `Client-Cert` und in Ihrer Quell-Anforderungsrichtlinie zulassen, damit der Ursprung sie empfängt.
- Certificate-chain-depth— CloudFront ermöglicht eine maximale Tiefe von 4 für die Weiterleitung der Client-Zertifikatskette an die Ursprünge.

Aktivieren Sie den Passthrough-Modus

Konsole

1. Aktualisieren Sie alle Cache-Verhaltensweisen, um die Richtlinie für verwalteten `CachingDisabled` Cache zu verwenden.
2. Öffnen Sie die CloudFront Konsole und wählen Sie Ihre Distribution aus.
3. Wählen Sie den Tab Allgemein und dann unter Einstellungen die Option Bearbeiten aus.
4. Wählen Sie unter Gegenseitige Viewer-Authentifizierung (mTLS) die Option Passthrough aus.
5. Wählen Sie Änderungen speichern aus.

AWS CLI

Stellen Sie sicher, dass alle Cache-Verhaltensweisen auf die verwaltete `CachingDisabled` Richtlinie verweisen, und aktualisieren Sie dann die Verteilungskonfiguration:

```
{
  "ViewerMtlsConfig": {
    "Mode": "passthrough"
  }
}
```

Header wurden an den Ursprung weitergeleitet

CloudFront fügt der an Ihren Ursprung gesendeten Anfrage die folgenden Header hinzu:

- `Client-Cert`— Das vom Client vorgelegte Endentitätszertifikat (Leaf), Base64-kodiert.
- `Client-Cert-Chain`— Die Zertifikatskette (ohne das Blatt) als strukturierte Feldliste. Jedes Zertifikat ist Base64-kodiert. `Client-Cert-Chain`ist ein Header vom Typ List. Es kann mehrmals in einer Anfrage vorkommen. Durch die Verkettung aller Werte wird die ursprüngliche Kettenreihenfolge beibehalten. `Client-Cert-Chain`wird weggelassen, wenn der Client nur ein einziges Zertifikat vorlegt.

CloudFront löscht alle eingehenden `Client-Cert` oder `Client-Cert-Chain` Header aus der Client-Anfrage, bevor die eigentlichen Zertifikatsdaten hinzugefügt werden. Dies verhindert Header-Spoofing.

Passen Sie die Zertifikat-Header an

Verwenden Sie die mTLS-Hilfsfunktionen in CloudFront Viewer-Request-Funktionen, um die Zertifikatsheader umzubenennen, neu zu formatieren oder zu kombinieren, bevor sie Ihren Ursprung erreichen. Dies ist nützlich, wenn Sie von anderen Diensten migrieren, die andere Header-Namen oder Verschlüsselungsformate für Zertifikate verwenden.

Beobachtbarkeit mithilfe von Verbindungsprotokollen

CloudFront Verbindungsprotokolle bieten einen detaillierten Einblick in gegenseitige TLS-Authentifizierungsereignisse, sodass Sie die Zertifikatsvalidierung überwachen, Verbindungsversuche verfolgen und Authentifizierungsprobleme beheben können.

Was sind Verbindungsprotokolle?

Verbindungsprotokolle enthalten detaillierte Informationen über TLS-Handshakes und die Zertifikatsvalidierung für gemeinsame TLS-enabled Verteilungen. Im Gegensatz zu Standardzugriffsprotokollen, die Informationen zu HTTP-Anfragen aufzeichnen, konzentrieren sich Verbindungsprotokolle speziell auf die Phase des TLS-Verbindungsaufbaus, einschließlich:

- Verbindungsstatus (success/failure)
- Einzelheiten zum Client-Zertifikat
- TLS-Protokoll und Chiffrierinformationen
- Metriken zum Verbindungszeitpunkt
- Benutzerdefinierte Daten aus Verbindungsfunktionen

Diese Protokolle bieten einen umfassenden Einblick in zertifikatsbasierte Authentifizierungsereignisse und helfen Ihnen dabei, die Sicherheit zu überwachen, Probleme zu beheben und Compliance-Anforderungen zu erfüllen.

Aktivieren Sie Verbindungsprotokolle

Verbindungsprotokolle sind nur für Distributionen mit aktivierter gegenseitiger TLS-Authentifizierung verfügbar. Sie können Verbindungsprotokolle an mehrere Ziele senden, darunter CloudWatch Logs, Amazon Data Firehose und Amazon S3.

Voraussetzungen

Bevor Sie Verbindungsprotokolle aktivieren:

- Konfigurieren Sie Mutual TLS für Ihre CloudFront Distribution

- Aktivieren Sie Verbindungsprotokolle für Ihre CloudFront Distribution
- Stellen Sie sicher, dass Sie über die erforderlichen Berechtigungen für das von Ihnen gewählte Protokollierungsziel verfügen
- Für die kontoübergreifende Bereitstellung konfigurieren Sie die entsprechenden IAM-Richtlinien

Um Verbindungsprotokolle zu aktivieren (Konsole)

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home.
2. Wählen Sie aus der Verteilerliste Ihre TLS-enabled M-Distribution aus.
3. Wählen Sie die Registerkarte Logging (Protokollierung) aus.
4. Wählen Sie Hinzufügen aus.
5. Wählen Sie den Dienst aus, der Ihre Logs erhalten soll:
   - CloudWatch Protokolle
   - Firehose
   - Amazon S3
6. Wählen Sie unter Ziel die Ressource für den ausgewählten Dienst aus:
   - Geben Sie für CloudWatch Logs den Namen der Protokollgruppe ein
   - Wählen Sie für Firehose den Firehose-Lieferstream
   - Geben Sie für Amazon S3 den Bucket-Namen ein (optional mit einem Präfix)
7. (Optional) Zusätzliche Einstellungen konfigurieren
   - Feldauswahl: Wählen Sie bestimmte Protokollfelder aus, die aufgenommen werden sollen.
   - Ausgabeformat: Wählen Sie zwischen JSON, Plain, W3C, Raw oder Parquet (nur S3).
   - Feldtrennzeichen: Geben Sie an, wie Protokollfelder getrennt werden sollen.
8. Wählen Sie Save Changes (Änderungen speichern)

Um Verbindungsprotokolle zu aktivieren (AWS CLI)

Das folgende Beispiel zeigt, wie Verbindungsprotokolle mithilfe der CloudWatch API aktiviert werden:

```
# Step 1: Create a delivery source
aws logs put-delivery-source \
```

```
  --name "cf-mtls-connection-logs" \
  --resource-arn "arn:aws:cloudfront::123456789012:distribution/E1A2B3C4D5E6F7" \
  --log-type CONNECTION_LOGS

# Step 2: Create a delivery destination
aws logs put-delivery-destination \
  --name "s3-destination" \
  --delivery-destination-configuration \
  "destinationResourceArn=arn:aws:s3:::amzn-s3-demo-bucket1"

# Step 3: Create the delivery
aws logs create-delivery \
  --delivery-source-name "cf-mtls-connection-logs" \
  --delivery-destination-arn "arn:aws:logs:us-east-1:123456789012:delivery-
destination:s3-destination"
```

> **Note**
>
> Wenn Sie die CloudWatch API verwenden, müssen Sie die Region USA Ost (Nord-Virginia) (us-east-1) angeben, auch wenn Sie Logs in andere Regionen liefern.

Felder des Verbindungsprotokolls

Verbindungsprotokolle enthalten detaillierte Informationen zu jedem TLS-Verbindungsversuch:

Feld	Description	Beispiel
`eventTimestamp`	Zeitstempel nach ISO 8601, als die Verbindung hergestellt wurde oder fehlgeschlagen ist	1731620046814
`connectionId`	Eindeutiger Bezeichner für die TLS-Verbindung	`oLHiEKbQSn8lkvJfA3D4gFowK3_iZ0g4i5nMUjE1Akod8TuAzn5nzg==`
`connectionStatus`	Der Status des mTLS-Verbindungsversuchs.	`Success` oder `Failed`

Feld	Description	Beispiel
`clientIp`	IP-Adresse des Clients, der eine Verbindung herstellt	`2001:0db8:85a3:0000:0000:8a2e:0370:7334`
`clientPort`	Vom Client verwendeter Port	12137
`serverIp`	IP-Adresse des CloudFront Edge-Servers	`99.84.71.136`
`distributionId`	CloudFront Vertriebs-ID	`E2DX1SLDPK0123`
`distributionTenantId`	CloudFront ID des Vertriebs mandanten (falls zutreffend)	`dt_2te1Ura9X3R2iCGNjW123`
`tlsProtocol`	Verwendete TLS-Protokollversion	`TLSv1.3`
`tlsCipher`	Für die Verbindung verwendete TLS-Verschlüsselungssuite	`TLS_AES_128_GCM_SHA256`
`tlsHandshakeDuration`	Dauer des TLS-Handshakes in Millisekunden	153
`tlsSni`	Wert für die Angabe des Servernamens aus dem TLS-Handshake	`d111111abcdef8.cloudfront.net`
`clientLeafCertSerialNumber`	Seriennummer des Client-Zertifikats	`00:b1:43:ed:93:d2:d8:f3:9d`
`clientLeafCertSubject`	Betrefffeld des Kundenzertifikats	`C=US, ST=WA, L=Seattle, O=Amazon.com, OU=CloudFront, CN=client.test.mtls.net`

Feld	Description	Beispiel
`clientLeafCertIssuer`	Feld des Ausstellers des Kundenzertifikats	`C=US, ST=WA, L=Seattle, O=Amazon.com, OU=CloudFront, CN=test.mtls.net`
`clientLeafCertValidity`	Gültigkeitsdauer des Zertifikats des Kunden	`NotBefore=2025-06-05T23:28:21Z;NotAfter=2125-05-12T23:28:21Z`
`connectionLogCustomData`	Mithilfe von Verbindungsfunktionen hinzugefügte benutzerdefinierte Daten	`REVOKED:00:b1:43:ed:93:d2:d8:f3:9d`

Fehlercodes für die Verbindung

```
Failed:ClientCertMaxChainDepthExceeded
Failed:ClientCertMaxSizeExceeded
Failed:ClientCertUntrusted
Failed:ClientCertNotYetValid
Failed:ClientCertExpired
Failed:ClientCertTypeUnsupported
Failed:ClientCertInvalid
Failed:ClientCertIntentInvalid
Failed:ClientCertRejected
Failed:ClientCertMissing
Failed:ClientCertOcspProcessingError
Failed:ClientCertOcspDenied
Failed:TcpError
Failed:TcpTimeout
Failed:ConnectionFunctionError
Failed:ConnectionFunctionDenied
Failed:Internal
Failed:UnmappedConnectionError
```

Wenn Verbindungen fehlschlagen, werden bestimmte Ursachencodes CloudFront aufgezeichnet:

Code	Description
ClientCertMaxChain DepthExceeded	Die maximale Tiefe der Zertifikatskette wurde überschritten
ClientCertMaxSizeExceeded	Die maximale Zertifikatsgröße wurde überschritten
ClientCertUntrusted	Das Zertifikat ist nicht vertrauenswürdig
ClientCertNotYetValid	Das Zertifikat ist noch nicht gültig
ClientCertExpired	Das Zertifikat ist abgelaufen
ClientCertTypeUnsupported	Der Zertifikattyp wird nicht unterstützt
ClientCertInvalid	Das Zertifikat ist ungültig
ClientCertIntentInvalid	Die Absicht des Zertifikats ist ungültig
ClientCertRejected	Das Zertifikat wurde durch eine benutzerdefinierte Überprüfung abgelehnt
ClientCertMissing	Das Zertifikat fehlt
TcpError	Beim Versuch, eine Verbindung herzustellen, ist ein Fehler aufgetreten
TcpTimeout	Die Verbindung konnte innerhalb des Timeouts nicht hergestellt werden
ConnectionFunctionError	Während der Ausführung der Verbindungsfunktion wurde eine nicht abgefangene Ausnahme ausgelöst
Intern	Ein interner Dienstfehler ist aufgetreten
UnmappedConnectionError	Es ist ein Fehler aufgetreten, der in keine der anderen Kategorie n fällt

Origin Mutual TLS mit CloudFront

Die gegenseitige TLS-Authentifizierung (Mutual Transport Layer Security Authentication — mTLS) ist ein Sicherheitsprotokoll, das die Standard-TLS-Authentifizierung um eine bidirektionale zertifikatsbasierte Authentifizierung erweitert, bei der sowohl Client als auch Server ihre Identität nachweisen müssen, bevor eine sichere Verbindung hergestellt werden kann.

Viewer mTLS gegen Origin mTLS

Gegenseitige Authentifizierung (mTLs) kann zwischen Zuschauern und deiner CloudFront Distribution (Zuschauer-mTLs) and/or sowie zwischen deiner CloudFront Distribution und dem Absender (Ursprungs-MTLs) aktiviert werden. Diese Dokumentation bezieht sich auf die ursprüngliche mTLS-Konfiguration. Die mTLS-Konfiguration für den Viewer finden Sie unter:. Gegenseitige TLS-Authentifizierung mit CloudFront (Viewer mTLS)

Origin mTLS ermöglicht es CloudFront , sich mithilfe von Client-Zertifikaten gegenüber Ihren Ursprungsservern zu authentifizieren. Mit Origin-mTLS können Sie sicherstellen, dass nur Ihre autorisierten CloudFront Distributionen Verbindungen zu Ihren Anwendungsservern herstellen können, was zum Schutz vor unbefugten Zugriffsversuchen beiträgt.

> **Note**
>
> Bei ursprünglichen mTLS-Verbindungen CloudFront fungiert es als Client und übergibt sein Client-Zertifikat während des TLS-Handshakes an Ihren Ursprungsserver. CloudFront führt keine Überprüfung der Gültigkeit oder des Sperrstatus des Client-Zertifikats durch — dies liegt in der Verantwortung Ihres Ursprungsservers. Ihre ursprüngliche Infrastruktur muss so konfiguriert sein, dass sie das Client-Zertifikat anhand seines Trust Stores validiert, den Ablauf des Zertifikats überprüft und gemäß Ihren Sicherheitsanforderungen Sperrprüfungen (z. B. CRL- oder OCSP-Validierung) durchführt. CloudFront Die Rolle ist auf die Vorlage des Zertifikats beschränkt. Die gesamte Logik und Sicherheitsrichtlinien für das Zertifikat werden von Ihren Ursprungsservern durchgesetzt.

Funktionsweise

Bei einem Standard-TLS-Handshake zwischen CloudFront und einem Ursprungsserver legt nur der Ursprungsserver ein Zertifikat vor, gegenüber dem seine Identität nachgewiesen werden kann. CloudFront Bei originalen mTLS wird der Authentifizierungsprozess bidirektional. Wenn CloudFront

versucht wird, eine Verbindung zu Ihrem Ursprungsserver herzustellen, CloudFront wird während des TLS-Handshakes ein Client-Zertifikat vorgelegt. Ihr Ursprungsserver validiert dieses Zertifikat anhand seines Trust Stores, bevor die sichere Verbindung hergestellt wird.

Anwendungsfälle

Origin mTLS befasst sich mit mehreren kritischen Sicherheitsszenarien, in denen herkömmliche Authentifizierungsmethoden zu einem betrieblichen Aufwand führen:

- Hybrid- und Multi-Cloud-Sicherheit — Sie können Verbindungen zwischen CloudFront und Startpunkten sichern, die außerhalb AWS oder von öffentlichen Quellen gehostet werden. AWS Dadurch entfällt die Notwendigkeit, IP-Zulassungslisten oder benutzerdefinierte Header-Lösungen zu verwalten, sodass eine konsistente zertifikatsbasierte Authentifizierung in Rechenzentren vor Ort AWS und bei Drittanbietern gewährleistet ist. Medienunternehmen, Einzelhändler und Unternehmen, die verteilte Infrastrukturen betreiben, profitieren von standardisierten Sicherheitskontrollen für ihre gesamte Infrastruktur.
- B2B-API- und Backend-Sicherheit — Sie können Ihre Backend-APIs und Microservices vor direkten Zugriffsversuchen schützen und gleichzeitig die Leistungsvorteile beibehalten CloudFront . SaaS-Plattformen, Zahlungsabwicklungssysteme und Unternehmensanwendungen mit strengen Authentifizierungsanforderungen können überprüfen, ob API-Anfragen nur von autorisierten CloudFront Distributionen stammen, wodurch Man-in-the-Middle-Angriffe und unbefugte Zugriffsversuche verhindert werden.

Wichtig: Anforderungen an den Origin-Server

Origin mTLS erfordert, dass Ihre Ursprungsserver so konfiguriert sind, dass sie die gegenseitige TLS-Authentifizierung unterstützen. Ihre ursprüngliche Infrastruktur muss in der Lage sein:

- Anforderung und Validierung von Client-Zertifikaten bei TLS-Handshakes
- Pflege eines Vertrauensspeichers mit den Zertifikaten der Zertifizierungsstelle, die die Client-Zertifikate ausgestellt haben CloudFront
- Protokollierung und Überwachung gegenseitiger TLS-Verbindungsereignisse
- Verwaltung von Richtlinien zur Zertifikatsvalidierung und Behandlung von Authentifizierungsfehlern

CloudFront übernimmt die clientseitige Zertifikatspräsentation, aber Ihre Ursprungsserver sind für die Validierung dieser Zertifikate und die Verwaltung der gegenseitigen TLS-Verbindung verantwortlich.

Stellen Sie sicher, dass Ihre Ausgangsinfrastruktur ordnungsgemäß konfiguriert ist, bevor Sie die ursprünglichen mTLs aktivieren. CloudFront

Erste Schritte

Um Origin-mTLs mit zu implementieren CloudFront, müssen Sie das Client-Zertifikat in AWS Certificate Manager importieren, Ihren Ursprungsserver so konfigurieren, dass gegenseitiges TLS erforderlich ist, und Origin-mTLs in Ihrer Distribution aktivieren. CloudFront Die folgenden Abschnitte enthalten schrittweise Anleitungen für jede Konfigurationsaufgabe.

Themen

- Zertifikatsverwaltung mit AWS Zertifikatsmanager
- Aktivieren Sie Mutual Origin TLS für Distributionen CloudFront
- Verwenden von CloudFront Funktionen mit gegenseitigem Ursprungs-TLS

Zertifikatsverwaltung mit AWS Zertifikatsmanager

AWS Certificate Manager (ACM) speichert die Client-Zertifikate, die CloudFront Ihren Ursprungsservern bei der gegenseitigen TLS-Originalauthentifizierung vorgelegt werden.

Unterstützung durch Zertifizierungsstellen

CloudFront Origin-mTLS erfordert Client-Zertifikate mit Extended Key Usage (EKU) für die TLS-Client-Authentifizierung. Aufgrund dieser Anforderung müssen Sie Zertifikate von Ihrer Zertifizierungsstelle ausstellen und sie in AWS Certificate Manager importieren. Die Funktionen zur automatischen Bereitstellung und Verlängerung von ACM-Zertifikaten sind für originale mTLS-Client-Zertifikate nicht verfügbar. CloudFront Origin-mTLS unterstützt Client-Zertifikate aus zwei Quellen:

- AWS Private Zertifizierungsstelle: Sie können Zertifikate von AWS Private CA mithilfe von Zertifikatsvorlagen ausstellen, die die TLS-Client-Authentifizierung im Feld „Erweiterte Schlüsselverwendung“ enthalten (z. B. die EndEntityClientAuthCertificate Vorlage). Nachdem Sie das Zertifikat von AWS Private CA ausgestellt haben, müssen Sie es in ACM in der Region USA Ost (Nord-Virginia) (us-east-1) importieren. Dieser Ansatz bietet die Sicherheitsvorteile von AWS Private CA und gibt Ihnen gleichzeitig die Kontrolle über das Lebenszyklusmanagement von Zertifikaten.
- Third-party private Zertifizierungsstellen: Sie können auch Zertifikate aus Ihrer bestehenden privaten Zertifizierungsstellen-Infrastruktur ausstellen und sie in ACM importieren. Auf diese Weise können Sie Ihre aktuellen Prozesse zur Zertifikatsverwaltung beibehalten und gleichzeitig

die mTLS-Funktionen von CloudFront Origin nutzen. Zertifikate müssen die TLS-Client-Authentifizierung im Feld „Erweiterte Schlüsselverwendung“ enthalten und im PEM-Format mit dem Zertifikat, dem privaten Schlüssel und der Zertifikatskette vorliegen.

> **Important**
>
> Sowohl bei AWS privaten Zertifizierungsstellen als auch bei Zertifizierungsstellen von Drittanbietern sind Sie dafür verantwortlich, die Ablaufdaten der Zertifikate zu überwachen und erneuerte Zertifikate vor Ablauf in ACM zu importieren. Die automatische ACM-Verlängerungsfunktion gilt nicht für importierte Zertifikate, die für Original-MTLs verwendet werden.

Anforderungen und Spezifikationen des Zertifikats

Anforderungen an das Kundenzertifikat

- Format: PEM-Format (Privacy Enhanced Mail)
- Komponenten: Zertifikat, privater Schlüssel und Zertifikatskette
- Maximale Tiefe der Zertifikatskette: 3 (Blattzertifikat + Zwischenzertifikat + Stammzertifikat)
- Maximale Größe der Zertifikatskette: 64 KB
- Größe des Zertifikats: 96 KB darf nicht überschritten werden
- Maximale Größe des privaten Schlüssels: 5 KB (ACM-Beschränkung)
- Maximale Anzahl eindeutiger Ursprungs-MTLS-Zertifikat-ARNs, die pro API-Aufruf zum Erstellen oder Aktualisieren einer CloudFront Distribution hinzugefügt oder geändert werden können: 5
- Region: Zertifikate müssen in ACM in der Region USA Ost (Nord-Virginia) (us-east-1) gespeichert werden

Unterstützte Zertifikatsspezifikationen

- Typ des Zertifikats: X.509v3
- Algorithmen für öffentliche Schlüssel:
  - RSA: 2048-Bit
  - ECDSA: P-256
- Signatur-Algorithmen:

- SHA256, SHA384, SHA512 mit RSA
- SHA256, SHA384, SHA512 mit ECDSA
- SHA256, SHA384, SHA512 mit MGF1 RSASSA-PSS

- Erweiterte Schlüsselnutzung (erforderlich): Das Zertifikat erfordert die Erweiterung Extended Key Usage (EKU), die auf TLS Client Authentication gesetzt ist, um sicherzustellen, dass es für mTLS-Zwecke autorisiert ist

Anforderungen an das Serverzertifikat

Ihre Ursprungsserver müssen während des gegenseitigen TLS-Handshakes Zertifikate von öffentlich vertrauenswürdigen Zertifizierungsstellen vorlegen. Vollständige Informationen zu den Anforderungen an das Ursprungsserverzertifikat finden Sie unter Anforderungen für die Verwendung von SSL/TLS Zertifikaten mit CloudFront.

Fordern Sie ein Zertifikat an oder importieren Sie es

Bevor Sie Origin-mTLs aktivieren, muss in ACM ein Client-Zertifikat verfügbar sein.

Fordern Sie ein Zertifikat an und importieren Sie es von AWS Private CA

Voraussetzungen:

- Eine AWS private Zertifizierungsstelle, die in Ihrem Konto konfiguriert ist
- Genehmigung zur Ausstellung von Zertifikaten von AWS Private CA
- Erlaubnis zum Import von Zertifikaten in ACM
- Eine https://docs.aws.amazon.com/privateca/latest/userguide/UsingTemplates.html Zertifikatsvorlagen-ARN`Extended key usage:TLS web client authentication`, die für Ihren Anwendungsfall geeignet ist
- Installieren Sie OpenSSL, AWS CLI und jq (zum Parsen von JSON).

Um ein Zertifikat von PCA anzufordern und in ACM zu importieren (AWS CLI)

1. Legen Sie Ihren privaten CA-ARN zur einfacheren Wiederverwendung in einer Variablen fest.

```
PCA_ARN="arn:aws:acm-pca:region:account:certificate-authority/12345678..."
```

2. Verwenden Sie OpenSSL, um einen P-256 privaten ECDSA-Schlüssel (Prime256v1-Kurve) und eine Certificate Signing Request (CSR) zu generieren. Stellen Sie dabei sicher, dass das Flag -

nodes verwendet wird, damit der private Schlüssel unverschlüsselt bleibt, wie es für den ACM-Import erforderlich ist.

```
openssl req -new -newkey ec -pkeyopt ec_paramgen_curve:prime256v1 -nodes \
    -keyout private.key \
    -out request.csr \
    -subj "/CN=client.example.com"
```

3. Senden Sie den CSR an Ihre AWS private Zertifizierungsstelle, um ein Zertifikat auszustellen, das die ARN des neu ausgestellten Zertifikats zurückgibt.

```
CERT_ARN=$(aws acm-pca issue-certificate \
    --certificate-authority-arn "$PCA_ARN" \
    --csr fileb://request.csr \
    --signing-algorithm "SHA256WITHECDSA" \
    --validity Value=365,Type="DAYS" \
    --template-arn arn:aws:acm-pca:::template/EndEntityCertificate/V1 \
    --query 'CertificateArn' --output text)
```

4. Rufen Sie das Zertifikatspaket von AWS PCA mit dem Befehl get-certificate ab, der sowohl das Leaf-Zertifikat als auch die Kette zurückgibt, und verwenden Sie dann jq, um sie in verschiedene Dateien aufzuteilen.

```
# Retrieve the full certificate bundle in JSON format
aws acm-pca get-certificate \
    --certificate-authority-arn "$PCA_ARN" \
    --certificate-arn "$CERT_ARN" \
    --output json > full_cert.json

# Split into Leaf and Chain
jq -r '.Certificate' full_cert.json > leaf_cert.pem
jq -r '.CertificateChain' full_cert.json > cert_chain.pem
```

5. Importieren Sie den unverschlüsselten privaten Schlüssel, das Leaf-Zertifikat und die Zertifikatskette in AWS ACM. Verwenden Sie dabei das fileb: //-Protokoll, um Binärdateidaten in der CLI korrekt zu verarbeiten.

```
aws acm import-certificate \
    --certificate fileb://leaf_cert.pem \
    --private-key fileb://private.key \
    --certificate-chain fileb://cert_chain.pem \
    --region us-east-1 \
```

```
    --query 'CertificateArn' \
    --output text
```

Importieren Sie ein Zertifikat von einer Drittanbieter-Zertifizierungsstelle

Voraussetzungen:

- Ein Zertifikat, ein unverschlüsselter privater Schlüssel und eine Zertifikatskette von Ihrer Zertifizierungsstelle im PEM-Format
- Das Zertifikat muss die erweiterte Schlüsselnutzung für die TLS-Client-Authentifizierung enthalten
- Berechtigungen zum Importieren von Zertifikaten in ACM

Um ein Zertifikat in ACM zu importieren (AWS CLI)

```
aws acm import-certificate \
  --certificate fileb://certificate.pem \
  --private-key fileb://private-key.pem \
  --certificate-chain fileb://certificate-chain.pem \
  --region us-east-1 \
  --query 'CertificateArn' \
  --output text
```

Nächste Schritte

Nachdem Sie Ihr Client-Zertifikat in ACM erhalten oder importiert haben, können Sie Ihren Ursprungsserver so konfigurieren, dass eine gegenseitige TLS-Authentifizierung erforderlich ist, und Origin-mTLS in Ihrer CloudFront Distribution aktivieren. Anweisungen zur Aktivierung von Origin-mTLs finden Sie im CloudFront nächsten Abschnitt „Mutual Origin TLS für CloudFront Distributionen aktivieren“.

Aktivieren Sie Mutual Origin TLS für Distributionen CloudFront

Nachdem Sie ein Client-Zertifikat über AWS Certificate Manager erhalten und Ihren Ursprungsserver so konfiguriert haben, dass gegenseitiges TLS erforderlich ist, können Sie Origin-mTLS in Ihrer CloudFront Distribution aktivieren.

Voraussetzungen und Anforderungen

Bevor Sie Origin-mTLs für eine CloudFront Distribution aktivieren, stellen Sie sicher, dass Sie über Folgendes verfügen:

- Ein Client-Zertifikat, das im AWS Certificate Manager in der Region USA Ost (Nord-Virginia) (us-east-1) gespeichert ist
- Ursprungsserver, die so konfiguriert sind, dass sie eine gegenseitige TLS-Authentifizierung erfordern und Client-Zertifikate validieren
- Ursprungsserver, die Zertifikate von öffentlich vertrauenswürdigen Zertifizierungsstellen vorlegen
- Berechtigungen zum Ändern von CloudFront Distributionen
- Origin mTLS ist nur in den Tarifen Business, Premium oder Pay as you go verfügbar.

> **Note**
>
> Origin-mTLs können für benutzerdefinierte Ursprünge (einschließlich außerhalb gehosteter Ursprünge AWS) und für AWS Ursprünge konfiguriert werden, die gegenseitiges TLS unterstützen, wie Application Load Balancer und API Gateway.

> **Important**
>
> Die folgenden CloudFront Funktionen werden mit Origin-mTLs nicht unterstützt:
>
> - gRPC-Verkehr: Das gRPC-Protokoll wird für Ursprünge mit aktiviertem Origin-mTLS nicht unterstützt
> - WebSocket Verbindungen: Das WebSocket Protokoll wird für Ursprünge mit aktiviertem Origin-mTLS nicht unterstützt
> - VPC-Ursprünge: Ursprungs-MTLs können nicht mit VPC-Ursprüngen verwendet werden
> - Trigger für Origin-Request und Origin-Response mit Lambda @Edge: Lambda @Edge - Funktionen in den Positionen Origin Request und Origin Response werden mit Origin-mTLs nicht unterstützt
> - Eingebettete POPs: Origin-mTLS wird für eingebettete POPs nicht unterstützt

Aktivieren Sie die ursprünglichen mTLs

Per-origin Die Konfiguration ermöglicht es Ihnen, verschiedene Client-Zertifikate für verschiedene Ursprünge innerhalb derselben Distribution anzugeben. Dieser Ansatz bietet maximale Flexibilität, wenn für Ihre Ursprünge unterschiedliche Authentifizierungsanforderungen gelten.

Für neue Distributionen (Konsole)

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home.
2. Wählen Sie Distribution erstellen
3. Wähle einen Preisplan: Wähle entweder Business oder Premium oder Pay As You Go (Origin mTLS ist im kostenlosen Tarif nicht verfügbar)
4. Wähle im Abschnitt Origin-Einstellungen die Option Origin Type als Andere
5. Wähle im Abschnitt Origin-Einstellungen die Option Origin-Einstellungen anpassen
6. Konfiguriere deinen ersten Ursprung (Domainname, Protokoll usw.)
7. Suchen Sie in der Originalkonfiguration nach mTLS
8. Schalten Sie mTLS auf Ein
9. Wählen Sie für das Client-Zertifikat Ihr Zertifikat im Certificate Manager aus AWS
10. (Optional) Fügen Sie zusätzliche Ursprünge mit ihren eigenen mTLS-Ursprungskonfigurationen hinzu
11. Vervollständigen Sie die verbleibenden Verteilungseinstellungen und wählen Sie Distribution erstellen

Für bestehende Distributionen (Konsole)

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home.
2. Wählen Sie aus der Verteilerliste die Distribution aus, die Sie ändern möchten. (Hinweis: Vergewissern Sie sich, dass für Ihre Distribution ein Pro-, Premium- oder Pay-As-You-Go-Preisplan gilt. Wenn nicht, müssen Sie Ihren Preisplan aktualisieren, bevor Sie Origin-mTLs aktivieren.)
3. Wähle den Tab Origins
4. Wählen Sie den Ursprung aus, den Sie konfigurieren möchten, und wählen Sie Bearbeiten
5. Suchen Sie in den Ursprungseinstellungen nach mTLS

6. Schalten Sie mTLS auf Ein
7. Wählen Sie für das Client-Zertifikat Ihr Zertifikat im Certificate Manager aus AWS . (Hinweis: Es werden nur Client-Zertifikate aufgeführt, bei denen die EKU-Eigenschaft (Extended Key Usage) auf „TLS-Client-Authentifizierung“ gesetzt ist.)
8. Wählen Sie Save Changes (Änderungen speichern)
9. Wiederholen Sie diesen Vorgang bei Bedarf für weitere Ursprünge

Verwenden AWS CLI

Geben Sie für die Konfiguration pro Ursprung die ursprünglichen mTLS-Einstellungen in der Konfiguration jedes Ursprungs an:

```
{
  "Origins": {
    "Quantity": 2,
    "Items": [
      {
        "Id": "origin-1",
        "DomainName": "api.example.com",
        "CustomOriginConfig": {
          "HTTPSPort": 443,
          "OriginProtocolPolicy": "https-only"
        },
        "OriginMtlsConfig": {
          "ClientCertificateArn": "arn:aws:acm:us-east-1:123456789012:certificate/
cert-1"
        }
      },
      {
        "Id": "origin-2",
        "DomainName": "backend.example.com",
        "CustomOriginConfig": {
          "HTTPSPort": 443,
          "OriginProtocolPolicy": "https-only"
        },
        "OriginMtlsConfig": {
          "CertificateArn": "arn:aws:acm:us-east-1:123456789012:certificate/cert-2"
        }
      }
    ]
  }
```

```
}
```

> **Note**
>
> CloudFront stellt das Client-Zertifikat nicht zur Verfügung, wenn der Server es nicht anfordert, sodass die Verbindung normal hergestellt werden kann.

Nächste Schritte

Nachdem Sie die ursprünglichen mTLs in Ihrer CloudFront Distribution aktiviert haben, können Sie Authentifizierungsereignisse mithilfe von CloudFront Zugriffsprotokollen überwachen.

Verwenden von CloudFront Funktionen mit gegenseitigem Ursprungs-TLS

CloudFront Functions bietet einfache, serverlose Rechenleistung am Netzwerkrand, um die Inhaltsbereitstellung individuell anzupassen. Wenn Sie Origin Mutual TLS mit CloudFront Functions verwenden, gibt es bestimmte Verhaltensweisen und Einschränkungen, die Sie bei der Auswahl und Manipulation von Origin beachten sollten.

Unterstützte CloudFront Funktionen und Operationen

CloudFront Funktionen können auf folgende Weise mit m TLS-enabled Origins interagieren:

aktualisieren RequestOrigin ()

Die Funktion update RequestOrigin () unterstützt begrenzte Änderungen bei der Arbeit mit m TLS-enabled Origins:

- Zwischen ursprünglichen mTLS-Ursprüngen wechseln: Sie können die Anfrage so aktualisieren, dass sie an einen anderen Ursprung weitergeleitet wird, der Original-mTLs verwendet, vorausgesetzt, beide Ursprünge verwenden dasselbe Client-Zertifikat. Auf diese Weise können Sie eine benutzerdefinierte Routing-Logik implementieren und gleichzeitig die gegenseitige TLS-Authentifizierung beibehalten. Das Umschalten zwischen Ursprüngen, die unterschiedliche Zertifikate verwenden, wird über die `createRequestOriginGroup()` APIs `selectRequestOriginById()` und unterstützt.
- Deaktivieren von Ursprungs-MTLS: Sie können von einem TLS-enabled M-Ursprung zu einem Nicht-MTLS-Ursprung wechseln, indem Sie die Funktion einrichten`mTLSConfig: 'off'`.

Dies bietet die Flexibilität, die gegenseitige TLS-Authentifizierung auf der Grundlage der Anforderungsmerkmale bedingt zu deaktivieren.

Beispiel: Umschalten zwischen mTLS-Ursprüngen mit demselben Zertifikat

```
import cf from 'cloudfront';

function handler(event) {
    var request = event.request;

    // Route to different origin based on request path
    if (request.uri.startsWith('/api/v2')) {
        cf.updateRequestOrigin({
            "domainName": "api-v2.example.com",
            "mTLSConfig": "inherit",
            // If no value is provided for mTLSConfig, it defaults to inherit
            // Both origins must use the same certificate
        });
    }

    return request;
}
```

Beispiel: Bedingtes Deaktivieren der ursprünglichen mTLs

```
import cf from 'cloudfront';

function handler(event) {
    var request = event.request;

    // Disable mTLS for specific paths
    if (request.uri.startsWith('/public')) {
        cf.updateRequestOrigin({
            "domainName": "public-origin.example.com",
            "mTLSConfig": "off"
        });
    }

    return request;
}
```

wählen Sie () RequestOriginById

Die `selectRequestOriginById()` Funktion unterstützt die Auswahl von Ursprüngen, für die Mutual TLS (Origin) aktiviert ist. Sie können diese Funktion verwenden, um Anfragen dynamisch an meine in Ihrer Distribution konfigurierten TLS-enabled Ursprünge weiterzuleiten. Wenn Sie einen Ursprung mit gegenseitigem TLS (Ursprung) anhand der ID auswählen, wird das Client-Zertifikat CloudFront verwendet, das in den Verteilungseinstellungen für diesen Ursprung konfiguriert ist.

Beispiel: Auswahl eines für beide Seiten aktivierten Ursprungs mit TLS (Ursprung) anhand der ID

```
import cf from 'cloudfront';

function handler(event) {
    var request = event.request;

    // Select mTLS-enabled origin based on request characteristics
    if (request.uri.startsWith('/secure-api')) {
        cf.selectRequestOriginById("mtls-origin-1");
    }

    return request;
}
```

erstellen RequestOriginGroup ()

Die `createRequestOriginGroup()` Funktion unterstützt das Erstellen von Ursprungsgruppen, die gegenseitige TLS-aktivierte (Origin) -aktivierte Ursprünge enthalten. Sie können innerhalb von CloudFront Functions dynamisch Ursprungsgruppen mit m TLS-enabled Ursprüngen für Failover-Szenarien erstellen.

Beispiel: Erstellen einer Ursprungsgruppe mit gegenseitigem TLS-aktiviertem Ursprung (Origin)

```
import cf from 'cloudfront';

function handler(event) {
    // Create origin group with mTLS-enabled primary and failover origins
    cf.createRequestOriginGroup({
        "originIds": ["mtls-origin-primary", "mtls-origin-failover"],
        "failoverCriteria": {
            "statusCodes": [500, 502, 503, 504]
        }
    });
```

```
    return event.request;
}
```

Bereitstellen privater Inhalte mit signierten URLs und signierten Cookies

Viele Unternehmen, die Inhalte über das Internet verteilen, möchten den Zugriff auf Dokumente, geschäftliche Daten, Medien-Streams oder Inhalte für ausgewählte Benutzer (z. B. Benutzer, die eine Gebühr bezahlt haben) einschränken. Um diese privaten Inhalte mithilfe von folgenden Methoden sicher bereitzustellen CloudFront, können Sie Folgendes tun:

- Erfordern Sie, dass Ihre Benutzer mithilfe speziell CloudFront signierter URLs oder signierter Cookies auf Ihre privaten Inhalte zugreifen.
- Erfordern Sie, dass Ihre Benutzer mithilfe von CloudFront URLs auf Ihre Inhalte zugreifen, nicht über URLs, die direkt auf dem Ursprungsserver (z. B. Amazon S3 oder einen privaten HTTP-Server) auf Inhalte zugreifen. Das Anfordern von CloudFront URLs ist nicht erforderlich, wir empfehlen es jedoch, um zu verhindern, dass Benutzer die Einschränkungen umgehen, die Sie in signierten URLs oder signierten Cookies angeben.

Weitere Informationen finden Sie unter Beschränken des Zugriffs auf Dateien.

So können Sie private Inhalte bereitstellen

Gehen Sie wie folgt vor, CloudFront um die Bereitstellung privater Inhalte zu konfigurieren:

1. (Optional, aber empfohlen) Erfordern Sie, dass Ihre Benutzer nur über Folgendes auf Ihre Inhalte zugreifen können CloudFront. Die dafür verwendete Methode hängt davon ab, ob Sie Amazon S3 oder benutzerdefinierte Ursprünge verwenden:

   - Amazon S3 – Siehe the section called “Beschränken des Zugriffs auf einen Amazon-S3-Ursprung”.
   - Benutzerdefinierter Ursprung – Siehe Beschränken des Zugriffs auf Dateien auf benutzerdefinierten Ursprungsservern.

   Benutzerdefinierte Ursprünge umfassen Amazon EC2, als Website-Endpunkte konfigurierte Amazon S3 Buckets, Elastic Load Balancing und Ihre eigenen HTTP-Webserver.

2. Geben Sie die vertrauenswürdigen Schlüsselgruppen oder vertrauenswürdigen Aussteller an, die Sie zum Erstellen von signierten URLs oder signierten Cookies verwenden möchten. Es wird empfohlen, vertrauenswürdige Schlüsselgruppen zu verwenden. Weitere Informationen finden Sie unter Festlegen der Signaturgeber, die signierte URLs und signierte Cookies erstellen können.
3. Schreiben Sie Ihre Anwendung so, dass Anfragen von autorisierten Benutzern entweder mit signierten URLs oder mit `Set-Cookie`-Headern beantwortet werden, die signierte Cookies setzen. Befolgen Sie die Schritte auf einem der folgenden Themen:

   - Verwenden signierter URLs
   - Verwenden signierter Cookies

   Wenn Sie sich nicht sicher sind, welche Methode Sie verwenden sollen, lesen Sie nach unter Entscheidung zwischen signierten URLs und signierter Cookies.

Themen

- Beschränken des Zugriffs auf Dateien
- Festlegen der Signaturgeber, die signierte URLs und signierte Cookies erstellen können
- Entscheidung zwischen signierten URLs und signierter Cookies
- Verwenden signierter URLs
- Verwenden signierter Cookies
- Linux-Befehle und OpenSSL für die Base64-Codierung und die Verschlüsselung
- Code-Beispiele für das Erstellen einer Signatur für eine signierte URL

Beschränken des Zugriffs auf Dateien

Sie können den Benutzerzugriff auf Ihre privaten Inhalte auf zwei Arten steuern:

- Beschränken Sie den Zugriff auf Dateien in CloudFront Caches.
- Beschränken Sie den Zugriff auf Dateien auf Ihrem Ursprungsserver, indem Sie einen der folgenden Schritte ausführen:
  - Richten Sie eine Ursprungszugriffssteuerung (OAC) für Ihren Amazon-S3-Bucket ein.
  - Konfigurieren Sie benutzerdefinierte Header für einen privaten HTTP-Server (ein benutzerdefinierter Ursprung).

Beschränken Sie den Zugriff auf Dateien in Caches CloudFront

Sie können so konfigurieren CloudFront , dass Benutzer entweder über signierte URLs oder signierte Cookies auf Ihre Dateien zugreifen müssen. Anschließend entwickeln Sie Ihre Anwendung entweder so, dass signierte URLs erstellt und an authentifizierte Benutzer verteilt werden, oder so, dass `Set-Cookie`-Header gesendet werden, die signierte Cookies für authentifizierte Benutzer setzen. (Wenn Sie einigen Benutzern langfristigen Zugriff auf eine geringe Anzahl von Dateien gewähren möchten, können Sie auch manuell signierte URLs erstellen.)

Wenn Sie signierte URLs oder signierte Cookies erstellen, um den Zugriff auf Ihre Dateien zu kontrollieren, können Sie die folgenden Einschränkungen angeben:

- Ein Enddatum und eine Endzeit, nach welchen die URL nicht mehr gültig ist.
- (Optional) Das Datum und die Zeit, an welchen die URL gültig wird.
- (Optional) Die IP-Adresse oder der IP-Adressbereich der Computer, die für den Zugriff auf Ihre Inhalte verwendet werden können.

Ein Teil einer signierten URL oder eines signierten Cookies wird gehasht und mit dem privaten Schlüssel von einem Schlüsselpaar aus einem privaten und einem öffentlichen Schlüssel signiert. Wenn jemand eine signierte URL oder ein signiertes Cookie verwendet, um auf eine Datei zuzugreifen, werden die signierten und unsignierten Teile der URL oder des Cookies CloudFront verglichen. Wenn sie nicht übereinstimmen, wird die Datei CloudFront nicht bereitgestellt.

Sie müssen entweder private RSA-2048- oder ECDSA-256-Schlüssel für das Signieren von URLs oder Cookies verwenden.

Beschränken des Zugriffs auf Dateien in Amazon-S3-Buckets

Sie können den Inhalt in Ihrem Amazon S3-Bucket optional sichern, sodass Benutzer über die angegebene CloudFront Distribution darauf zugreifen können, aber nicht direkt über Amazon S3-URLs darauf zugreifen können. Dadurch wird verhindert, dass jemand die Amazon S3-URL umgeht CloudFront und verwendet, um Inhalte abzurufen, für die Sie den Zugriff einschränken möchten. Dieser Schritt ist für die Verwendung signierter URLs nicht notwendig, wird jedoch empfohlen.

Um zu verlangen, dass Benutzer über CloudFront URLs auf Ihre Inhalte zugreifen, gehen Sie wie folgt vor:

- Erteilen Sie einer CloudFront Origin-Zugriffskontrolle die Erlaubnis, die Dateien im S3-Bucket zu lesen.

- Erstellen Sie die Origin-Zugriffskontrolle und verknüpfen Sie sie mit Ihrer CloudFront Distribution.
- Entfernen Sie die Berechtigung, Amazon S3-URLs zum Lesen der Dateien zu verwenden, für alle anderen Benutzer.

Für weitere Informationen siehe the section called “Beschränken des Zugriffs auf einen Amazon-S3-Ursprung” oder the section called “Beschränken Sie den Zugriff auf einen Amazon S3 Multi-Region Access Point-Ursprung”.

Beschränken des Zugriffs auf Dateien auf benutzerdefinierten Ursprungsservern

Wenn Sie einen benutzerdefinierten Ursprungsserver verwenden, können Sie optional benutzerdefinierte Header einrichten, um den Zugriff einzuschränken. CloudFront Um Ihre Dateien von einem benutzerdefinierten Ursprung abzurufen, müssen die Dateien CloudFront über eine Standard-HTTP- (oder HTTPS-) Anfrage zugänglich sein. Durch die Verwendung benutzerdefinierter Header können Sie den Zugriff auf Ihre Inhalte jedoch weiter einschränken, sodass Benutzer nur über CloudFront und nicht direkt darauf zugreifen können. Dieser Schritt ist für die Verwendung signierter URLs nicht notwendig, wird jedoch empfohlen.

Um zu verlangen, dass Benutzer über diese Website auf Inhalte zugreifen CloudFront, ändern Sie die folgenden Einstellungen in Ihren CloudFront Distributionen:

Angepasste Ursprungs-Header

Konfigurieren Sie so CloudFront , dass benutzerdefinierte Header an Ihren Ursprung weitergeleitet werden. Siehe Konfiguriere CloudFront , um benutzerdefinierte Header zu Quellanfragen hinzuzufügen.

Viewer-Protokollrichtlinien

Konfigurieren Sie Ihre Verteilung so, dass Betrachter für den Zugriff auf Ihre CloudFront HTTPS verwenden müssen. Siehe Viewer-Protokollrichtlinien.

Ursprungsprotokollrichtlinien

Konfigurieren Sie Ihre Distribution so, dass CloudFront Sie dasselbe Protokoll wie die Zuschauer verwenden müssen, um Anfragen an den Ursprung weiterzuleiten. Siehe Protokoll (nur benutzerdefinierte Ursprünge).

Nachdem Sie diese Änderungen vorgenommen haben, aktualisieren Sie Ihre Anwendung auf Ihrem benutzerdefinierten Ursprung, sodass nur Anfragen akzeptiert werden, die die benutzerdefinierten Header enthalten, für deren Versand Sie konfiguriert CloudFront haben.

Die Kombination aus Viewer Protocol Policy und Origin Protocol Policy stellt sicher, dass die benutzerdefinierten Header während der Übertragung verschlüsselt werden. Wir empfehlen jedoch, dass Sie in regelmäßigen Abständen die folgenden Schritte ausführen, um die benutzerdefinierten Header so zu rotieren, dass sie zu Ihrem CloudFront Ursprung weitergeleitet werden:

1. Aktualisieren Sie Ihre CloudFront Distribution, um mit der Weiterleitung neuer Header an Ihren benutzerdefinierten Ursprung zu beginnen.
2. Aktualisieren Sie Ihre Anwendung so, dass der neue Header als Bestätigung dafür akzeptiert wird, von wem die Anfrage stammt CloudFront.
3. Wenn Anfragen den Header, den Sie ersetzen, nicht mehr enthalten, aktualisieren Sie Ihre Anwendung so, dass der alte Header nicht mehr als Bestätigung akzeptiert wird, von dem die Anfrage stammt CloudFront.

Festlegen der Signaturgeber, die signierte URLs und signierte Cookies erstellen können

Themen

- Wählen Sie zwischen vertrauenswürdigen Schlüsselgruppen (empfohlen) und AWS-Konten
- Erstellen von Schlüsselpaaren für Ihre Signaturgeber
- Neuformatierung des privaten Schlüssels (nur .NET und Java)
- Hinzufügen eines Signaturgebers zu einer Distribution
- Rotieren von Schlüsselpaaren

Um signierte URLs oder signierte Cookies zu erstellen, benötigen Sie einen Aussteller. Ein Unterzeichner ist entweder eine vertrauenswürdige Schlüsselgruppe, in der Sie erstellen CloudFront, oder ein AWS Konto, das ein CloudFront Schlüsselpaar enthält. Es wird empfohlen, vertrauenswürdige Schlüsselgruppen mit signierten URLs und signierten Cookies zu verwenden. Weitere Informationen finden Sie unter Wählen Sie zwischen vertrauenswürdigen Schlüsselgruppen (empfohlen) und AWS-Konten.

Der Aussteller erfüllt zwei Zwecke:

- Sobald Sie den Unterzeichner zu Ihrer Distribution hinzufügen, wird ab sofort verlangt CloudFront , dass Zuschauer signierte URLs oder signierte Cookies verwenden, um auf Ihre Dateien zuzugreifen.
- Beim Erstellen signierter URLs oder signierter Cookies verwenden Sie den privaten Schlüssel aus dem Schlüsselpaar des Ausstellers, um einen Teil der URL oder des Cookies zu signieren. Wenn jemand eine eingeschränkte Datei anfordert, CloudFront vergleicht er die Signatur in der URL oder im Cookie mit der unsignierten URL oder dem Cookie, um sicherzustellen, dass sie nicht manipuliert wurde. CloudFront überprüft auch, ob die URL oder das Cookie gültig ist, was beispielsweise bedeutet, dass das Ablaufdatum und die Uhrzeit nicht abgelaufen sind.

Wenn Sie einen Aussteller angeben, geben Sie indirekt auch die Dateien an, die signierte URLs oder signierte Cookies erfordern, indem Sie den Aussteller zu einem Cache-Verhalten hinzufügen. Wenn Ihre Distribution über nur ein Cache-Verhalten verfügt, müssen Viewer für den Zugriff auf in der Distribution befindliche Dateien die signierten URLs oder signierten Cookies verwenden. Wenn Sie mehrere Cache-Verhalten erstellen und vertrauenswürdige Aussteller zu einigen Cache-Verhalten hinzufügen und zu anderen nicht, können Sie es erforderlich machen, dass Viewer signierte URLs oder signierte Cookies für den Zugriff auf einige Dateien verwenden und für den Zugriff auf andere nicht.

Gehen Sie wie folgt vor, um die Unterzeichner (die privaten Schlüssel) anzugeben, die signierte URLs oder signierte Cookies erstellen dürfen, und um die Unterzeichner zu Ihrer CloudFront Distribution hinzuzufügen:

1. Entscheiden Sie, ob Sie eine vertrauenswürdige Schlüsselgruppe oder einen AWS-Konto als Unterzeichner verwenden möchten. Es wird empfohlen, eine vertrauenswürdige Schlüsselgruppe zu verwenden. Weitere Informationen finden Sie unter Wählen Sie zwischen vertrauenswürdigen Schlüsselgruppen (empfohlen) und AWS-Konten.
2. Erstellen Sie für den Aussteller, den Sie in Schritt 1 ausgewählt haben, Schlüsselpaar aus öffentlichem und privatem Schlüssel. Weitere Informationen finden Sie unter Erstellen von Schlüsselpaaren für Ihre Signaturgeber.
3. Wenn Sie .NET oder Java verwenden, um signierte URLs oder signierte Cookies zu erstellen, formatieren Sie den privaten Schlüssel neu. Weitere Informationen finden Sie unter Neuformatierung des privaten Schlüssels (nur .NET und Java).
4. Geben Sie in der Distribution, für die Sie signierte URLs oder signierte Cookies erstellen, den Aussteller an. Weitere Informationen finden Sie unter Hinzufügen eines Signaturgebers zu einer Distribution.

Wählen Sie zwischen vertrauenswürdigen Schlüsselgruppen (empfohlen) und AWS-Konten

Um signierte URLs oder signierte Cookies verwenden zu können, benötigen Sie einen Aussteller. Ein Unterzeichner ist entweder eine vertrauenswürdige Schlüsselgruppe, in der Sie erstellen CloudFront, oder eine, AWS-Konto die ein CloudFront Schlüsselpaar enthält. Es wird empfohlen, vertrauenswürdige Schlüsselgruppen zu verwenden, und zwar aus den folgenden Gründen:

- Bei CloudFront Schlüsselgruppen müssen Sie den Root-Benutzer des AWS Kontos nicht verwenden, um die öffentlichen Schlüssel für CloudFront signierte URLs und signierte Cookies zu verwalten. AWS Best Practices empfehlen, den Root-Benutzer nicht zu verwenden, wenn Sie ihn nicht verwenden müssen.
- Mit CloudFront Schlüsselgruppen können Sie öffentliche Schlüssel, Schlüsselgruppen und vertrauenswürdige Unterzeichner mithilfe der CloudFront API verwalten. Sie können die API verwenden, um die Schlüsselerstellung und die Schlüsselrotation zu automatisieren. Wenn Sie den AWS Root-Benutzer verwenden, müssen Sie den verwenden, AWS-Managementkonsole um CloudFront Schlüsselpaare zu verwalten, sodass Sie den Prozess nicht automatisieren können.
- Da Sie Schlüsselgruppen mit der CloudFront API verwalten können, können Sie auch AWS Identity and Access Management (IAM-) Berechtigungsrichtlinien verwenden, um einzuschränken, was verschiedene Benutzer tun dürfen. Beispielsweise können Sie Benutzern erlauben, öffentliche Schlüssel hochzuladen, aber nicht zu löschen. Oder Sie können Benutzern erlauben, öffentliche Schlüssel zu löschen, jedoch nur dann, wenn bestimmte Bedingungen erfüllt sind, z. B. Multi-Factor-Authentication, das Senden der Anforderung aus einem bestimmten Netzwerk oder das Senden der Anforderung innerhalb eines bestimmten Datums- und Uhrzeitbereichs.
- Mit CloudFront Schlüsselgruppen können Sie Ihrer CloudFront Distribution eine höhere Anzahl an öffentlichen Schlüsseln zuordnen, sodass Sie die öffentlichen Schlüssel flexibler verwenden und verwalten können. Standardmäßig können Sie bis zu vier Schlüsselgruppen einer einzelnen Distribution zuordnen, und Sie können bis zu fünf öffentliche Schlüssel in einer Schlüsselgruppe haben.

  Wenn Sie den Root-Benutzer des AWS Kontos zur Verwaltung von CloudFront Schlüsselpaaren verwenden, können Sie nur bis zu zwei aktive CloudFront Schlüsselpaare pro AWS Konto haben.

Erstellen von Schlüsselpaaren für Ihre Signaturgeber

Jeder Unterzeichner, den Sie zum Erstellen CloudFront signierter URLs oder signierter Cookies verwenden, muss über ein öffentlich-privates Schlüsselpaar verfügen. Der Unterzeichner verwendet

seinen privaten Schlüssel, um die URL oder die Cookies zu signieren, und CloudFront verwendet den öffentlichen Schlüssel, um die Signatur zu verifizieren.

Die Art und Weise, wie Sie ein Schlüsselpaar erstellen, hängt davon ab, ob Sie eine vertrauenswürdige Schlüsselgruppe als Unterzeichner (empfohlen) oder ein CloudFront Schlüsselpaar verwenden. Weitere Informationen finden Sie in den folgenden Abschnitten. Das Schlüsselpaar, das Sie erstellen, muss die folgenden Anforderungen erfüllen:

- Es muss sich entweder um ein SSH-2 RSA 2048- oder ECDSA 256-Schlüsselpaar handeln.
- Es muss im base64-kodierten PEM-Format vorliegen.

Um Ihre Anwendungen zu schützen, empfehlen wir Ihnen, Schlüsselpaare regelmäßig zu rotieren. Weitere Informationen finden Sie unter Rotieren von Schlüsselpaaren.

Erstellen eines Schlüsselpaars für eine vertrauenswürdige Schlüsselgruppe (empfohlen)

Führen Sie die folgenden Schritte aus, um ein Schlüsselpaar für eine vertrauenswürdige Schlüsselgruppe zu erstellen:

1. Erstellen Sie das öffentliche-private Schlüsselpaar.
2. Laden Sie den öffentlichen Schlüssel auf hoch. CloudFront
3. Fügen Sie den öffentlichen Schlüssel zu einer CloudFront Schlüsselgruppe hinzu.

Weitere Informationen finden Sie in den folgenden Verfahren.

Erstellen eines Schlüsselpaares

> **Note**
>
> In den folgenden Schritten wird OpenSSL als Beispiel für eine Methode zur Erstellung eines Schlüsselpaars verwendet. Es gibt viele andere Möglichkeiten, ein RSA- oder ECDSA-Schlüsselpaar zu erstellen.

1. Führen Sie einen der folgenden Beispielbefehle aus:

   - Der folgende Beispielbefehl verwendet OpenSSL, um ein RSA-Schlüsselpaar mit einer Länge von 2048 Bit zu generieren und in der Datei mit dem Namen `private_key.pem` zu speichern.

```
openssl genrsa -out private_key.pem 2048
```

- Der folgende Beispielbefehl verwendet OpenSSL, um ein ECDSA-Schlüsselpaar mit der `prime256v1`-Kurve zu generieren und in der Datei namens `private_key.pem` zu speichern.

```
openssl ecparam -name prime256v1 -genkey -noout -out privatekey.pem
```

2. Die erstellte Datei enthält den öffentlichen und den privaten Schlüssel. Der folgende Beispielbefehl extrahiert den öffentlichen Schlüssel aus der Datei mit dem Namen `private_key.pem`.

```
openssl rsa -pubout -in private_key.pem -out public_key.pem
```

Sie laden den öffentlichen Schlüssel (in der `public_key.pem`-Datei) später im folgenden Verfahren hoch.

Um den öffentlichen Schlüssel hochzuladen CloudFront

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home.
2. Wählen Sie im Navigationsmenü die Option Public keys (Öffentliche Schlüssel).
3. Wählen Sie Öffentlichen Schlüssel erstellen.
4. Führen Sie im Fenster Öffentlichen Schlüssel erstellen die folgenden Schritte aus:

   a. Geben Sie unter Key name (Schlüsselname) einen Namen ein, um den öffentlichen Schlüssel zu identifizieren.

   b. Fügen Sie unter Key value (Schlüsselwert) den öffentlichen Schlüssel ein. Wenn Sie die Schritte im vorangegangenen Verfahren ausgeführt haben, befindet sich der öffentliche Schlüssel in der Datei mit dem Namen `public_key.pem`. Um den Inhalt des öffentlichen Schlüssels zu kopieren und einzufügen, können Sie Folgendes tun:

      - Verwenden Sie den cat-Befehl in der macOS- oder Linux-Befehlszeile wie folgt:

```
cat public_key.pem
```

Kopieren Sie die Ausgabe dieses Befehls und fügen Sie sie dann in das Feld Key value (Schlüsselwert) ein.

- Öffnen Sie die `public_key.pem` Datei mit einem Klartext-Editor wie Notepad (unter Windows) oder TextEdit (unter macOS). Kopieren Sie den Inhalt der Datei und fügen Sie ihn dann in das Feld key value (Schlüsselwert) ein.

c. (Optional) Fügen Sie unter Comment (Kommentar) einen Kommentar hinzu, um den öffentlichen Schlüssel zu beschreiben.

Wenn Sie fertig sind, wählen Sie Add (Hinzufügen).

5. Notieren Sie die ID des öffentlichen Schlüssels. Sie verwenden diese später, wenn Sie signierte URLs oder signierte Cookies erstellen, als Wert des `Key-Pair-Id`-Feldes.

So fügen Sie den öffentlichen Schlüssel zu einer Schlüsselgruppe hinzu:

1. Öffnen Sie die CloudFront Konsole unter. https://console.aws.amazon.com/cloudfront/v4/home
2. Wählen Sie im Navigationsmenü die Option Key groups (Schlüsselgruppen).
3. Wählen Sie Add key group (Schlüsselgruppe hinzufügen).
4. Führen Sie auf der Seite Create key group (Schlüsselgruppe erstellen) die folgenden Schritte aus:

   a. Geben Sie unter Key group name (Schlüsselgruppenname) einen Namen ein, um die Schlüsselgruppe zu identifizieren.

   b. (Optional) Geben Sie unter Comment (Kommentar) einen Kommentar ein, um die Schlüsselgruppe zu beschreiben.

   c. Wählen Sie für Public keys (Öffentliche Schlüssel) den öffentlichen Schlüssel aus, der der Schlüsselgruppe hinzugefügt werden soll. Wählen Sie dann Add (Hinzufügen) aus. Wiederholen Sie diesen Schritt für jeden öffentlichen Schlüssel, den Sie der Schlüsselgruppe hinzufügen möchten.

5. Wählen Sie Create key group (Schlüsselgruppe erstellen) aus.
6. Notieren Sie den Namen der Schlüsselgruppe. Sie verwenden es später, um die Schlüsselgruppe einem Cache-Verhalten in einer CloudFront Distribution zuzuordnen. (In der

CloudFront API verwenden Sie die Schlüsselgruppen-ID, um die Schlüsselgruppe einem Cache-Verhalten zuzuordnen.)

Erstellen Sie ein CloudFront Schlüsselpaar (nicht empfohlen, erfordert das AWS-Konto Root-Benutzer)

> **Important**
>
> Es wird empfohlen, einen öffentlichen Schlüssel für eine vertrauenswürdige Schlüsselgruppe zu erstellen, anstatt diese Schritte auszuführen. Die empfohlene Methode zum Erstellen öffentlicher Schlüssel für signierte URLs und signierte Cookies finden Sie unter Erstellen eines Schlüsselpaars für eine vertrauenswürdige Schlüsselgruppe (empfohlen).

Sie können ein CloudFront Schlüsselpaar auf folgende Arten erstellen:

- Erstellen Sie ein Schlüsselpaar in AWS-Managementkonsole und laden Sie den privaten Schlüssel herunter. Weitere Informationen finden Sie im folgenden Verfahren.
- Erstellen Sie mithilfe einer Anwendung wie OpenSSL ein RSA-Schlüsselpaar und laden Sie den öffentlichen Schlüssel auf die AWS-Managementkonsole hoch. Weitere Informationen zum Erstellen eines RSA-Schlüsselpaars finden Sie unter Erstellen eines Schlüsselpaars für eine vertrauenswürdige Schlüsselgruppe (empfohlen).

Um CloudFront Schlüsselpaare zu erstellen in der AWS-Managementkonsole

1. Melden Sie sich AWS-Managementkonsole mit den Anmeldeinformationen des Root-Benutzers des AWS Kontos an.

   > **Important**
   >
   > IAM-Benutzer können keine CloudFront Schlüsselpaare erstellen. Sie müssen sich mit Stammbenutzer-Anmeldeinformationen anmelden, um Schlüsselpaare zu erstellen.

2. Wählen Sie Ihren Kontonamen und dann My Security Credentials (Meine Sicherheitsanmeldeinformationen) aus.
3. Wählen Sie CloudFront Schlüsselpaare aus.

4. Vergewissern Sie sich, dass Sie nicht mehr als ein aktives Schlüsselpaar haben. Sie können kein Schlüsselpaar erstellen, wenn Sie bereits zwei aktive Schlüsselpaare haben.
5. Wählen Sie Create a new key pair (Neues Schlüsselpaar erstellen) aus.

> **Note**
>
> Sie können sich auch dafür entscheiden, Ihr eigenes Schlüsselpaar zu erstellen und den öffentlichen Schlüssel hochzuladen. CloudFront Schlüsselpaare unterstützen 1024-, 2048- oder 4096-Bit-Schlüssel.

6. Wählen Sie im Dialogfeld Create Key Pair (Schlüsselpaar erstellen) die Option Download Private Key File (Private Schlüsseldatei herunterladen) und speichern Sie die Datei dann auf Ihrem Computer.

> **Important**
>
> Speichern Sie den privaten Schlüssel für Ihr CloudFront Schlüsselpaar an einem sicheren Ort und legen Sie die Berechtigungen für die Datei fest, sodass nur Sie sie lesen können. Wenn eine Person in den Besitz Ihres privaten Schlüssels gelangt, kann sie gültige signierte URLs und signierte Cookies generieren und Ihre Inhalte herunterladen. Sie können den privaten Schlüssel nicht erneut abrufen. Wenn Sie ihn also verlieren oder löschen, müssen Sie ein neues CloudFront Schlüsselpaar erstellen.

7. Notieren Sie die Schlüsselpaar-ID für Ihr Schlüsselpaar. (In der AWS-Managementkonsole wird dies als Zugriffsschlüssel-ID bezeichnet.) Sie verwenden sie beim Erstellen signierter URLs oder signierter Cookies.

Neuformatierung des privaten Schlüssels (nur .NET und Java)

Wenn Sie .NET oder Java verwenden, um signierte URLs oder signierte Cookies zu erstellen, können Sie den privaten Schlüssel aus Ihrem Schlüsselpaar nicht im PEM-Standardformat verwenden, um die Signatur zu erstellen. Führen Sie stattdessen die folgenden Schritte aus:

- .NET Framework – Konvertieren Sie den privaten Schlüssel in das vom .NET Framework verwendete XML-Format. Es sind mehrere Tools verfügbar.
- Java – Konvertieren Sie den privaten Schlüssel in das DER-Format. Eine Möglichkeit, dies zu tun, ist der folgende OpenSSL-Befehl. Im folgenden Befehl `private_key.pem` ist dies der Name

der Datei, die den PEM-formatted privaten Schlüssel enthält, und `private_key.der` der Name der Datei, die den DER-formatted privaten Schlüssel enthält, nachdem Sie den Befehl ausgeführt haben.

```
openssl pkcs8 -topk8 -nocrypt -in private_key.pem -inform PEM -out private_key.der -
outform DER
```

Um sicherzustellen, dass der Encoder ordnungsgemäß funktioniert, fügen Sie Ihrem Projekt die JAR für die Bouncy Castle-Java-Kryptografie-APIs hinzu und fügen Sie dann den Bouncy Castle-Anbieter hinzu.

Hinzufügen eines Signaturgebers zu einer Distribution

Ein Unterzeichner ist die vertrauenswürdige Schlüsselgruppe (empfohlen) oder das CloudFront Schlüsselpaar, das signierte URLs und signierte Cookies für eine Verteilung erstellen kann. Um signierte URLs oder signierte Cookies mit einer CloudFront Distribution zu verwenden, müssen Sie einen Unterzeichner angeben.

Aussteller sind mit Cache-Verhaltensweisen verknüpft. Damit können Sie signierte URLs oder signierte Cookies für einige Dateien verlangen und für andere in derselben Distribution nicht. Eine Distribution erfordert signierte URLs oder Cookies nur für Dateien, die mit den entsprechenden Cache-Verhaltensweisen verknüpft sind.

Ebenso kann ein Aussteller URLs oder Cookies nur für Dateien signieren, die mit den entsprechenden Cache-Verhaltensweisen verknüpft sind. Wenn Sie beispielsweise einen vertrauenswürdigen Aussteller für ein Cache-Verhalten und einen anderen vertrauenswürdigen Aussteller für ein anderes Cache-Verhalten verwenden, kann keiner dieser Aussteller signierte URLs oder Cookies für Dateien erstellen, die mit dem jeweils anderen Cache-Verhalten verknüpft sind.

> **Important**
>
> Bevor Sie einen Aussteller zu Ihrer Distribution hinzufügen, gehen Sie wie folgt vor:
>
> - Definieren Sie die Pfadmuster in den Cache-Verhaltensweisen und die Reihenfolge der Cache-Verhaltensweisen sorgfältig, damit Sie Benutzern keinen unbeabsichtigten Zugriff auf Ihre Inhalte gewähren oder verhindern, dass sie auf Inhalte zugreifen, die für alle verfügbar sein sollen.

Nehmen wir beispielsweise an, eine Anfrage stimmt mit dem Pfadmuster für zwei Cache-Verhalten überein. Das erste Cache-Verhalten erfordert keine signierten URLs oder signierten Cookies, das zweite Cache-Verhalten hingegen schon. Benutzer können auf die Dateien zugreifen, ohne signierte URLs oder signierte Cookies zu verwenden, da CloudFront das Cache-Verhalten verarbeitet, das mit der ersten Übereinstimmung verknüpft ist.

Weitere Informationen zu Pfadmustern finden Sie unter Pfadmuster.

- Überprüfen Sie bei einer Distribution, die Sie bereits zum Verteilen von Inhalten verwenden, ob Sie signierte URLs und signierte Cookies generieren können, bevor Sie einen Aussteller hinzufügen. Wenn Sie einen Unterzeichner hinzufügen, CloudFront lehnt Anfragen ab, die keine gültige signierte URL oder kein signiertes Cookie enthalten.

Sie können Ihrer Distribution Unterzeichner entweder über die CloudFront Konsole oder die API hinzufügen. CloudFront

Console

Die folgenden Schritte zeigen, wie Sie eine vertrauenswürdige Schlüsselgruppe als Aussteller hinzufügen. Sie können auch einen AWS-Konto als vertrauenswürdigen Unterzeichner hinzufügen, dies wird jedoch nicht empfohlen.

So fügen Sie einen Aussteller mit der Konsole zu einer Distribution hinzu:

1. Notieren Sie die Schlüsselgruppen-ID der Schlüsselgruppe, die Sie als vertrauenswürdigen Aussteller verwenden möchten. Weitere Informationen finden Sie unter Erstellen eines Schlüsselpaars für eine vertrauenswürdige Schlüsselgruppe (empfohlen).
2. Öffnen Sie die CloudFront Konsole unter. https://console.aws.amazon.com/cloudfront/v4/home
3. Wählen Sie die Distribution aus, deren Dateien Sie mit signierten URLs oder signierten Cookies schützen möchten.

> **Note**
>
> Wenn Sie einer neuen Distribution einen Aussteller hinzufügen möchten, geben Sie dieselben Einstellungen an, die in Schritt 6 beschrieben werden, wenn Sie die Distribution erstellen.

4. Wählen Sie die Registerkarte Behaviors aus.
5. Wählen Sie das Cache-Verhalten aus, dessen Pfadmuster mit den Dateien übereinstimmt, die Sie mit signierten URLs oder signierten Cookies schützen möchten, und wählen Sie dann Edit (Bearbeiten).
6. Führen Sie auf der Seite Edit Behavior (Verhalten bearbeiten) die folgenden Schritte aus:
   a. Klicken Sie bei Restrict Viewer Access (Use Signed URLs or Signed Cookies) (Viewer-Zugriff einschränken (Signierte URLs oder signierte Cookies verwenden)) auf Yes (Ja).
   b. Wählen Sie in Trusted Key Groups or Trusted Signer (Vertrauenswürdige Schlüsselgruppen oder vertrauenswürdiger Aussteller) die Option Trusted Key Groups (Vertrauenswürdige Schlüsselgruppen) aus.
   c. Wählen Sie unter Trusted Key Groups (Vertrauenswürdige Schlüsselgruppen) die hinzuzufügende Schlüsselgruppe aus, und wählen Sie dann Add (Hinzufügen). Wiederholen Sie diesen Vorgang, wenn Sie mehr als eine Schlüsselgruppe hinzufügen möchten.
7. Wählen Sie Yes, Edit (Ja, Bearbeiten), um das Cache-Verhalten zu aktualisieren.

API

Sie können die CloudFront API verwenden, um eine vertrauenswürdige Schlüsselgruppe als Unterzeichner hinzuzufügen. Sie können einen Aussteller zu einer vorhandenen oder zu einer neuen Distribution hinzufügen. Geben Sie in beiden Fällen die entsprechenden Werte im `TrustedKeyGroups`-Element an.

Sie können auch einen AWS-Konto als vertrauenswürdigen Unterzeichner hinzufügen, dies wird jedoch nicht empfohlen.

Weitere Informationen finden Sie in der CloudFront Amazon-API-Referenz zu den folgenden Themen:

- Eine bestehende Distribution aktualisieren — UpdateDistribution

- Eine neue Distribution erstellen — CreateDistribution

Rotieren von Schlüsselpaaren

Wir empfehlen, Ihre Schlüsselpaare für signierte URLs und signierte Cookies regelmäßig zu rotieren (zu ändern). Führen Sie die folgenden Schritte aus, um Schlüsselpaare zu rotieren, die Sie verwenden, um signierte URLs oder signierte Cookies zu erstellen, ohne die Gültigkeit von URLs oder Cookies aufzuheben, die noch nicht abgelaufen sind:

1. Erstellen Sie ein neues Schlüsselpaar, und fügen Sie den öffentlichen Schlüssel zu einer Schlüsselgruppe hinzu. Weitere Informationen finden Sie unter Erstellen eines Schlüsselpaars für eine vertrauenswürdige Schlüsselgruppe (empfohlen).
2. Wenn Sie im vorherigen Schritt eine neue Schlüsselgruppe erstellt haben, fügen Sie die Schlüsselgruppe der Distribution als Aussteller hinzu.

   > **Important**
   >
   > Entfernen Sie noch keine vorhandenen öffentlichen Schlüssel aus der Schlüsselgruppe oder Schlüsselgruppen aus der Distribution. Fügen Sie nur die neuen hinzu.

3. Aktualisieren Sie Ihre Anwendung so, dass Signaturen mithilfe der privaten Schlüssel von den neuen Schlüsselpaaren erstellt werden. Prüfen Sie, ob die signierten URLs oder Cookies, die mit den neuen privaten Schlüsseln signiert sind, funktionieren.
4. Warten Sie, bis das Ablaufdatum in URLs oder Cookies überschritten ist, die mittels des vorherigen privaten Schlüssels signiert wurden. Entfernen Sie dann den alten öffentlichen Schlüssel aus der Schlüsselgruppe. Wenn Sie in Schritt 2 eine neue Schlüsselgruppe erstellt haben, entfernen Sie die alte Schlüsselgruppe aus Ihrer Distribution.

Entscheidung zwischen signierten URLs und signierter Cookies

CloudFront signierte URLs und signierte Cookies bieten dieselbe grundlegende Funktionalität: Sie ermöglichen es Ihnen, zu kontrollieren, wer auf Ihre Inhalte zugreifen kann. Wenn Sie private Inhalte bereitstellen möchten CloudFront und versuchen zu entscheiden, ob Sie signierte URLs oder signierte Cookies verwenden möchten, sollten Sie Folgendes beachten.

Verwenden Sie signierte URLs in den folgenden Fällen:

- Sie möchten den Zugriff auf einzelne Dateien einschränken, z. B. einen Installations-Download für Ihre Anwendung.
- Ihre Benutzer verwenden einen Client (z. B. einen benutzerdefinierten HTTP-Client), der keine Cookies unterstützt.

Verwenden Sie signierte Cookies in den folgenden Fällen:

- Sie möchten Zugriff auf mehrere beschränkte Dateien bereitstellen, z. B. auf alle Dateien für ein Video im HLS-Format oder alle Dateien im Bereich des Abonnenten einer Website.
- Sie möchten Ihre aktuellen URLs nicht ändern.

Wenn Sie derzeit keine signierten URLs verwenden und Ihre (unsignierten) URLs einen der folgenden Abfragezeichenfolgeparameter enthalten, können Sie weder signierte URLs noch signierte Cookies verwenden:

- `Expires`
- `Policy`
- `Signature`
- `Key-Pair-Id`
- `Hash-Algorithm`

CloudFront geht davon aus, dass es sich bei URLs, die einen dieser Abfragezeichenfolgenparameter enthalten, um signierte URLs handelt, weshalb signierte Cookies nicht berücksichtigt werden.

Gleichzeitige Verwendung von signierten URLs und signierten Cookies

Signierte URLs haben Vorrang vor signierten Cookies. Wenn Sie sowohl signierte URLs als auch signierte Cookies verwenden, um den Zugriff auf dieselben Dateien zu kontrollieren, und ein Betrachter eine signierte URL verwendet, um eine Datei anzufordern, CloudFront bestimmt nur anhand der signierten URL, ob die Datei an den Betrachter zurückgegeben wird.

Verwenden signierter URLs

Eine signierte URL enthält zusätzliche Informationen, wie z. B. Ablaufdatum und -zeit, mit denen Sie den Zugriff auf Ihre Inhalte besser kontrollieren können. Diese zusätzlichen Informationen sind in einer Richtlinienanweisung enthalten, die entweder auf einer vordefinierten oder

einer benutzerdefinierten Richtlinie basieren. Die Unterschiede zwischen vordefinierten und benutzerdefinierten Richtlinien sind in den nächsten beiden Abschnitten beschrieben.

> **Note**
>
> Sie können einige signierte URLs mit vordefinierten Richtlinien und einige signierte URLs mit benutzerdefinierten Richtlinien für dieselbe Distribution erstellen.

Themen

- Entscheidung zwischen vordefinierten und benutzerdefinierten Richtlinien für signierte URLs
- Funktionsweise von signierten URLs
- Auswählen, wie lange signierte URLs gültig sind
- Wenn CloudFront überprüft das Ablaufdatum und die Uhrzeit einer signierten URL
- Beispiel-Code und Drittanbieter-Tools
- Erstellen einer signierten URL mit einer vordefinierten Richtlinie
- Erstellen einer signierten URL mit einer benutzerdefinierten Richtlinie

Entscheidung zwischen vordefinierten und benutzerdefinierten Richtlinien für signierte URLs

Wenn Sie eine signierte URL erstellen, schreiben Sie eine Richtlinienanweisung im JSON-Format, welche die Einschränkungen für die signierte URL festlegt, z. B. wie lange die URL gültig ist. Sie können entweder eine vordefinierte Richtlinie oder eine benutzerdefinierte Richtlinie verwenden. Im Folgenden finden Sie einen Vergleich zwischen vordefinierten und benutzerdefinierten Richtlinien:

Beschreibung	Vordefinierte Richtlinie	Benutzerdefinierte Richtlinie
Sie können die Richtlinienanweisung für mehrere Dateien wiederverwenden. Um die Richtlini enanweisung wiederzuverwenden, müssen Sie Platzhalterzeichen im `Resource`-Objekt verwenden . Weitere Informationen finden Sie unter Werte, die Sie in der Richtlinienanweisung für eine signierte	Nein	Ja

Beschreibung	Vordefinierte Richtlinie	Benutzerdefinierte Richtlinie
URL angeben, die eine benutzerdefinierte Richtlinie verwendet.)		
Sie können das Datum und die Zeit festlegen, ab denen Benutzer auf Ihre Inhalte zugreifen können.	Nein	Ja (optional)
Sie können das Datum und die Zeit festlegen, ab denen Benutzer nicht mehr auf Ihre Inhalte zugreifen können.	Ja	Ja
Sie können die IP-Adresse oder den Bereich von IP-Adressen der Benutzer festlegen, die auf Ihre Inhalte zugreifen können.	Nein	Ja (optional)
Die signierte URL enthält eine Base64-codierte Version der Richtlinie, was zu einer längeren URL führt.	Nein	Ja

Informationen zum Erstellen von signierten URLs mit einer vordefinierten Richtlinie finden Sie unter Erstellen einer signierten URL mit einer vordefinierten Richtlinie.

Informationen zum Erstellen von signierten URLs mit einer benutzerdefinierten Richtlinie finden Sie unter Erstellen einer signierten URL mit einer benutzerdefinierten Richtlinie.

Funktionsweise von signierten URLs

Hier finden Sie eine Übersicht darüber, wie Sie Amazon S3 für signierte URLs konfigurieren CloudFront und wie CloudFront reagiert wird, wenn ein Benutzer eine signierte URL verwendet, um eine Datei anzufordern.

1. Geben Sie in Ihrer CloudFront Distribution eine oder mehrere vertrauenswürdige Schlüsselgruppen an, die die öffentlichen Schlüssel enthalten, die zur Überprüfung der URL-Signatur verwendet werden CloudFront können. Sie verwenden die entsprechenden privaten Schlüssel, um die URLs zu signieren.

   CloudFront unterstützt signierte URLs mit RSA 2048- und ECDSA 256-Schlüsselsignaturen.

Weitere Informationen finden Sie unter Festlegen der Signaturgeber, die signierte URLs und signierte Cookies erstellen können.

2. Sie entwickeln Ihre Anwendung so, dass ermittelt wird, ob ein Benutzer über Zugriff auf Ihre Inhalte verfügen sollte, und dass signierte URLs für die Dateien oder Teile Ihrer Anwendung erstellt werden, für die Sie den Zugriff einschränken möchten. Weitere Informationen finden Sie unter den folgenden Themen:

   - Erstellen einer signierten URL mit einer vordefinierten Richtlinie
   - Erstellen einer signierten URL mit einer benutzerdefinierten Richtlinie

3. Ein Benutzer fordert eine Datei an, für die Sie signierte URLs verlangen.
4. Ihre Anwendung stellt sicher, dass der Benutzer zum Zugriff auf die Datei berechtigt ist: Er hat sich angemeldet, für den Zugriff auf die Inhalte bezahlt oder andere Anforderungen für den Zugriff erfüllt.
5. Ihre Anwendung erstellt eine signierte URL und gibt diese an den Benutzer zurück.
6. Über die signierte URL kann der Benutzer die Inhalte herunterladen oder streamen.

   Dieser Schritt erfolgt automatisch. Der Benutzer muss in der Regel keine zusätzlichen Schritte ausführen, um auf den Inhalt zuzugreifen. Wenn ein Benutzer beispielsweise in einem Web-Browser auf Ihre Inhalte zugreift, gibt Ihre Anwendung die signierte URL an den Browser zurück. Der Browser verwendet sofort die signierte URL, um auf die Datei im CloudFront Edge-Cache zuzugreifen, ohne dass der Benutzer eingreifen muss.
7. CloudFront verwendet den öffentlichen Schlüssel, um die Signatur zu validieren und zu bestätigen, dass die URL nicht manipuliert wurde. Wenn die Signatur ungültig ist, wird die Anfrage abgelehnt.

   Wenn die Signatur gültig ist, überprüft CloudFront die Richtlinien-Anweisung in der URL (oder erstellt eine, wenn Sie eine vordefinierte Richtlinie verwenden), um zu bestätigen, dass die Anfrage noch gültig ist. Wenn Sie beispielsweise ein Anfangs- und Enddatum sowie eine Uhrzeit für die URL angegeben haben, CloudFront bestätigt dies, dass der Benutzer versucht, während des Zeitraums, den Sie den Zugriff zulassen möchten, auf Ihre Inhalte zuzugreifen.

   Wenn die Anforderung die Anforderungen in der Richtlinienanweisung CloudFront erfüllt, werden die Standardoperationen ausgeführt: bestimmt, ob sich die Datei bereits im Edge-Cache befindet, leitet die Anforderung gegebenenfalls an den Ursprung weiter und gibt die Datei an den Benutzer zurück.

> **Note**
>
> Wenn eine unsignierte URL Abfragezeichenfolgenparameter enthält, stellen Sie sicher, dass Sie diese in den Teil der URL einschließen, den Sie signieren. Wenn Sie einer signierten URL nach der Erstellung eine Abfragezeichenfolge hinzufügen, gibt die URL einen HTTP 403-Status zurück.

> **Note**
>
> CloudFront erfordert keine signierte URL oder signierte Cookies für `OPTIONS` Anfragen. Dadurch können CORS-Preflight-Anfragen, die Browser ohne Anmeldeinformationen senden, bei Cacheverhalten, für die signierte URLs oder signierte Cookies erforderlich sind, erfolgreich sein. Leitet `OPTIONS` Anfragen ohne Überprüfung einer Signatur an Ihren Ursprung CloudFront weiter`OPTIONS`, sofern das Cache-Verhalten Folgendes zulässt. Stellen Sie sicher, dass Ihr Ursprung als Antwort auf `OPTIONS` Anfragen keine geschützten Inhalte zurückgibt, oder beschränken Sie die zulässigen Methoden auf `GET` und `HEAD` falls Ihre Anwendung CORS nicht verwendet.

Auswählen, wie lange signierte URLs gültig sind

Sie können private Inhalte mithilfe einer signierten URL verteilen, die nur für einen kurzen Zeitraum gültig ist – vielleicht nur für ein paar Minuten. Signierte URLs, die für solch einen kurzen Zeitraum gültig sind, eignen sich zum schnellen Verteilen von Inhalten für einen Benutzer für einen spezifischen Zweck. Ein Beispiel ist die Bereitstellung von Filmen und Musik-Downloads auf Abruf an Kunden. Wenn Ihre signierten URLs nur für einen kurzen Zeitraum gültig sind, möchten Sie diese wahrscheinlich mithilfe einer von Ihnen entwickelten Anwendung automatisch generieren. Wenn der Benutzer beginnt, eine Datei herunterzuladen oder eine Mediendatei abzuspielen, CloudFront vergleicht er die Ablaufzeit in der URL mit der aktuellen Uhrzeit, um festzustellen, ob die URL noch gültig ist.

Sie können private Inhalte auch mithilfe einer signierten URL verteilen, die für einen längeren Zeitraum gültig ist – vielleicht für viele Jahre. Signierte URLs, die für einen längeren Zeitraum gültig sind, eignen sich zum Verteilen von privaten Inhalten für bekannte Benutzer. Ein Beispiel ist die Bereitstellung von Unternehmensplänen für Investoren oder die Bereitstellung von

Schulungsmaterialien für Mitarbeiter. Sie können eine Anwendung entwickeln, um diese längerfristig signierten URLs für Sie zu generieren.

Wenn CloudFront überprüft das Ablaufdatum und die Uhrzeit einer signierten URL

CloudFront überprüft das Ablaufdatum und die Uhrzeit in einer signierten URL zum Zeitpunkt der HTTP-Anfrage. Wenn ein Client unmittelbar vor der Ablaufzeit mit dem Download einer großen Datei beginnt, sollte der Download abgeschlossen werden, auch wenn die Ablaufzeit während des Downloads überschritten wird. Wenn die TCP-Verbindung getrennt wird und der Client nach Überschreitung der Ablaufzeit versucht, den Download erneut zu starten, schlägt der Download fehl.

Wenn ein Client Bereichs-GETs verwendet, um eine Datei in kleineren Teilen abzurufen, schlagen alle GET-Anfragen nach Überschreitung der Ablaufzeit fehl. Weitere Informationen zu Bereichs-GETs finden Sie unter Wie CloudFront verarbeitet Teilanfragen für ein Objekt (Bereichs-GETs).

Beispiel-Code und Drittanbieter-Tools

Informationen zu Beispiel-Code, mit dem sich der gehashte und signierte Teil von signierten URLs erstellen lässt, finden Sie in den folgenden Themen:

- Erstellen einer URL-Signatur mit Perl
- Erstellen einer URL-Signatur mit PHP
- Erstellen einer URL-Signatur mithilfe von C# und dem .NET Framework
- Erstellen einer URL-Signatur mit Java

Erstellen einer signierten URL mit einer vordefinierten Richtlinie

Führen Sie die folgenden Schritte aus, um eine signierte URL mit einer vordefinierten Richtlinie zu erstellen.

So erstellen Sie eine signierte URL mit einer vordefinierten Richtlinie

1. Wenn Sie .NET oder Java verwenden, um signierte URLs zu erstellen, und den privaten Schlüssel für Ihr Schlüsselpaar noch nicht vom PEM-Standardformat in ein mit .NET oder Java kompatibles Format neu formatiert haben, holen Sie diesen Schritt jetzt nach. Weitere Informationen finden Sie unter Neuformatierung des privaten Schlüssels (nur .NET und Java).
2. Verketten Sie die folgenden Werte. Sie können das Format in diesem Beispiel für eine signierte URL verwenden.

```
https://d111111abcdef8.cloudfront.net/
image.jpg?color=red&size=medium&Expires=1767290400&Signature=nitfHRCrtziwO2HwPfWw~yYDhUF5Ew
j19DzZrvDh6hQ73lDx~-ar3UocvvRQVw6EkC~GdpGQyyOSKQim-
TxAnW7d8F5Kkai9HVx0FIu-5jcQb0UEmatEXAMPLE3ReXySpLSMj0yCd3ZAB4UcBCAqEijkytL6f3fVYNGQI6&Key-
Pair-Id=K2JCJMDEHXQW5F&Hash-Algorithm=SHA256
```

Entfernen Sie alle Leerzeichen (einschließlich Tabulatoren und Zeilenumbruchzeichen). Möglicherweise müssen Sie in der Zeichenfolge im Anwendungscode Escape-Zeichen einfügen. Alle Werte verfügen über `String` als Typ.

1. ***Base URL for the file***

   Die Basis-URL ist die CloudFront URL, die Sie für den Zugriff auf die Datei verwenden würden, wenn Sie keine signierten URLs verwenden würden, einschließlich Ihrer eigenen Abfragezeichenfolgenparameter, falls vorhanden. Im vorherigen Beispiel lautet die Basis-URL `https://d111111abcdef8.cloudfront.net/image.jpg`. Weitere Informationen zum Format der URLs für Distributionen finden Sie unter Passen Sie das URL-Format für Dateien in an CloudFront.

   - Die folgende CloudFront URL bezieht sich auf eine Bilddatei in einer Distribution (unter Verwendung des CloudFront Domainnamens). Beachten Sie, dass `image.jpg` ein `images`-Verzeichnis ist. Der Pfad zur Datei in der URL muss mit dem Pfad zur Datei auf Ihrem HTTP-Server oder in Ihrem Amazon S3 Bucket übereinstimmen.

     ```
     https://d111111abcdef8.cloudfront.net/images/image.jpg
     ```

   - Die folgende CloudFront URL enthält eine Abfragezeichenfolge:

     ```
     https://d111111abcdef8.cloudfront.net/images/image.jpg?size=large
     ```

   - Die folgenden CloudFront URLs gelten für Bilddateien in einer Distribution. Beide verwenden einen alternativen Domainnamen. Die zweite enthält eine Abfragezeichenfolge:

     ```
     https://www.example.com/images/image.jpg
     ```

     ```
     https://www.example.com/images/image.jpg?color=red
     ```

   - Die folgende CloudFront URL bezieht sich auf eine Bilddatei in einer Distribution, die einen alternativen Domainnamen und das HTTPS-Protokoll verwendet:

     ```
     https://www.example.com/images/image.jpg
     ```

2. **?**

   ? zeigt an, dass Abfrageparameter hinter der Basis-URL angegeben sind. Schließen Sie das ? auch dann ein, wenn Sie keine Abfrageparameter angeben.

   > **Note**
   >
   > Sie können die folgenden Abfrageparameter in beliebiger Reihenfolge angeben.

3. ***Your query string parameters, if any*&**

   (Optional) Sie können Ihre eigenen Abfragezeichenfolgenparameter eingeben. Fügen Sie dazu zwischen jedem ein Et-Zeichen (&) hinzu, wie in `color=red&size=medium`. Sie können die Abfragezeichenfolgenparameter in beliebiger Reihenfolge innerhalb der URL angeben.

   > **Important**
   >
   > Ihre Abfragezeichenfolgenparameter können nicht mit`Expires`, `SignatureKey-Pair-Id`, oder benannt werden`Hash-Algorithm`.

4. **Expires=*date and time in Unix time format (in seconds) and Coordinated Universal Time (UTC)***

   Das Datum und die Uhrzeit, wann die URL den Zugriff auf die Datei nicht mehr zulassen soll.

   Geben Sie das Ablaufdatum und die Ablaufzeit im Unix-Zeitformat (in Sekunden) und in koordinierter Weltzeit (UTC) an. Beispielsweise wird der 1. Januar 2026, 10:00 Uhr UTC in das Unix-Zeitformat `1767290400` umgewandelt, wie im Beispiel am Anfang dieses Themas dargestellt.

   Um die Epochenzeit zu verwenden, geben Sie eine 64-Bit-Ganzzahl für ein Datum an, das nicht nach `9223372036854775807` (Freitag, 11. April 2262 um 23:47:16.854 UTC) liegt.

   Weitere Informationen zu UTC finden Sie unter RFC 3339, Datum und Uhrzeit im Internet: Zeitstempel.

5. **&Signature=*hashed and signed version of the policy statement***

   Eine gehashte, signierte und Base64-codierte Version der JSON-Richtlinienanweisung. Weitere Informationen finden Sie unter Erstellen einer Signatur für eine signierte URL, die eine vordefinierte Richtlinie verwendet.

6. **&Key-Pair-Id=*public key ID for the CloudFront public key whose corresponding private key you're using to generate the signature***

   Die ID für einen CloudFront öffentlichen Schlüssel, zum Beispiel`K2JCJMDEHXQW5F`. Die öffentliche Schlüssel-ID gibt an CloudFront , welcher öffentliche Schlüssel zur Validierung der signierten URL verwendet werden soll. CloudFront vergleicht die Informationen in der Signatur mit den Informationen in der Grundsatzerklärung, um sicherzustellen, dass die URL nicht manipuliert wurde.

   Dieser öffentliche Schlüssel muss zu einer Schlüsselgruppe gehören, die ein vertrauenswürdiger Aussteller in der Distribution ist. Weitere Informationen finden Sie unter Festlegen der Signaturgeber, die signierte URLs und signierte Cookies erstellen können.

7. **&Hash-Algorithm=*SHA1 or SHA256***

   (Optional) Der Hash-Algorithmus, der zur Erstellung der Signatur verwendet wurde. Unterstützte Werte sind `SHA1` und `SHA256`. Wenn Sie diesen Parameter nicht angeben, wird CloudFront standardmäßig der Wert verwendet. `SHA1`

Erstellen einer Signatur für eine signierte URL, die eine vordefinierte Richtlinie verwendet

Zum Erstellen der Signatur für eine signierte URL, die eine vordefinierte Richtlinie verwendet, führen Sie die folgenden Schritte aus.

Themen

- Erstellen einer Richtlinienanweisung für eine signierte URL, die eine vordefinierte Richtlinie verwendet
- Erstellen einer Signatur für eine signierte URL, die eine vordefinierte Richtlinie verwendet

Erstellen einer Richtlinienanweisung für eine signierte URL, die eine vordefinierte Richtlinie verwendet

Wenn Sie eine signierte URL mit einer vordefinierten Richtlinie erstellen, ist der `Signature`-Parameter eine gehashte und signierte Version einer Richtlinienanweisung. Bei signierten URLs, die eine vordefinierte Richtlinie verwenden, fügen Sie die Richtlinienanweisung nicht in die URL ein, wie Sie es bei signierten URLs tun, die eine benutzerdefinierte Richtlinie verwenden. Zum Erstellen der Richtlinienanweisung führen Sie die folgenden Schritte aus.

So erstellen Sie die Richtlinienanweisung für eine signierte URL, die eine vordefinierte Richtlinie verwendet

1. Erstellen Sie die Richtlinienerklärung im folgenden JSON-Format und unter Verwendung der UTF-8 Zeichenkodierung. Fügen Sie alle Satzzeichen und andere Literalwerte genau wie angegeben ein. Informationen zu den Parametern `Resource` und `DateLessThan` finden Sie unter Werte, die Sie in der Richtlinienanweisung für eine signierte URL angeben, die eine vordefinierte Richtlinie verwendet.

   ```
   {
       "Statement": [
           {
               "Resource": "base URL or stream name",
               "Condition": {
                   "DateLessThan": {
                       "AWS:EpochTime": ending date and time in Unix time format and
    UTC
                   }
               }
           }
       ]
   }
   ```

2. Entfernen Sie alle Leerzeichen (einschließlich Tabulatoren und Zeilenumbruchzeichen) aus der Richtlinienanweisung. Möglicherweise müssen Sie in der Zeichenfolge im Anwendungscode Escape-Zeichen einfügen.

Werte, die Sie in der Richtlinienanweisung für eine signierte URL angeben, die eine vordefinierte Richtlinie verwendet

Beim Erstellen einer Richtlinienanweisung für eine vordefinierte Richtlinie geben Sie die folgenden Werte an.

Ressource

> **Note**
>
> Sie können nur einen Wert für `Resource` angeben.

Die Basis-URL einschließlich Ihrer Abfragezeichenfolgen, falls vorhanden, jedoch ohne die `Hash-Algorithm` Parameter CloudFront `Expires` `SignatureKey-Pair-Id`,, und, wie zum Beispiel:

```
https://d111111abcdef8.cloudfront.net/images/horizon.jpg?
size=large&license=yes
```

Beachten Sie Folgendes:

- Protokoll – Der Wert muss mit `http://` oder `https://` beginnen.
- Abfragezeichenfolgeparameter – Wenn Sie über keine Abfragezeichenfolgeparameter verfügen, lassen Sie das Fragezeichen weg.
- Alternative Domänennamen – Wenn Sie einen alternativen Domänennamen (CNAME) in der URL angeben, müssen Sie diesen alternativen Domänennamen angeben, wenn Sie auf Ihrer Webseite oder in Ihrer Anwendung auf die Datei verweisen. Geben Sie nicht die Amazon-S3-URL für das Objekt an.
- Sonderzeichen — Wenn die URL für Ihre Ressource ein Sternchen (*) oder ein Fragezeichen (?) enthält (z. B. im Objektschlüsselnamen), müssen Sie URL-encode diese Zeichen (`%2A`für * und `%3F` für?) verwenden. Das liegt daran, dass CloudFront unverschlüsselte ? Zeichen * und Zeichen im Ressourcenpfad als Platzhalterzeichen interpretiert werden. Wenn dies nicht der Fall ist URL-encoded, gewährt die signierte URL möglicherweise einen breiteren Zugriff als beabsichtigt.

DateLessThan

Das Ablaufdatum und die Ablaufzeit für die URL im Unix-Zeitformat (in Sekunden) und in koordinierter Weltzeit (UTC). Beispielsweise wird der 1. Januar 2026, 10:00 Uhr UTC im Unix-Zeitformat in 1767290400 umgewandelt.

Dieser Wert muss mit dem Wert des `Expires`-Abfragezeichenfolgeparameters in der signierten URL übereinstimmen. Setzen Sie den Wert nicht in Anführungszeichen.

Weitere Informationen finden Sie unter Wenn CloudFront überprüft das Ablaufdatum und die Uhrzeit einer signierten URL.

Beispiel-Richtlinienanweisung für eine signierte URL, die eine vordefinierte Richtlinie verwendet

Wenn Sie die folgende Beispiel-Richtlinienanweisung in einer signierten URL verwenden, kann ein Benutzer bis zum 1. Januar 2026, 10:00 Uhr UTC, auf die Datei `https://d111111abcdef8.cloudfront.net/horizon.jpg` zugreifen:

```
{
    "Statement": [
        {
            "Resource": "https://d111111abcdef8.cloudfront.net/horizon.jpg?
size=large&license=yes",
            "Condition": {
                "DateLessThan": {
                    "AWS:EpochTime": 1767290400
                }
            }
        }
    ]
}
```

Erstellen einer Signatur für eine signierte URL, die eine vordefinierte Richtlinie verwendet

Um den Wert für den Parameter `Signature` in einer signierten URL zu erstellen, müssen Sie die in Erstellen einer Richtlinienanweisung für eine signierte URL, die eine vordefinierte Richtlinie verwendet erstellte Richtlinienanweisung hashen und signieren.

Weitere Informationen und Beispiele für das Hashing, Signieren und Codieren der Richtlinienanweisung finden Sie unter:

- Linux-Befehle und OpenSSL für die Base64-Codierung und die Verschlüsselung
- Code-Beispiele für das Erstellen einer Signatur für eine signierte URL

> **Note**
>
> Die verlinkten Beispiele SHA-1 werden standardmäßig verwendet. Um es SHA-256 stattdessen zu verwenden, `sha1` ersetzen Sie es `sha256` in den OpenSSL-Befehlen und fügen Sie den `Hash-Algorithm=SHA256` Abfrageparameter in die signierte URL ein.

Option 1: So erstellen Sie eine Signatur mithilfe einer vordefinierten Richtlinie

1. Verwenden Sie die SHA-256 Hash-Funktion SHA-1 oder und den generierten privaten RSA- oder ECDSA-Schlüssel, um die Policy-Anweisung, die Sie in der Prozedur erstellt haben, zu hashen und zu signieren. So erstellen Sie die Richtlinienanweisung für eine signierte URL, die eine vordefinierte Richtlinie verwendet Verwenden Sie die Version der Richtlinienanweisung, die keine Leerzeichen mehr enthält.

   Wenn Sie es verwenden SHA-256, müssen Sie es `&Hash-Algorithm=SHA256` in die signierte URL aufnehmen.

   Verwenden Sie für den privaten Schlüssel, der für die Hash-Funktion erforderlich ist, einen privaten Schlüssel, dessen öffentlicher Schlüssel sich in einer aktiven vertrauenswürdigen Schlüsselgruppe für die Distribution befindet.

   > **Note**
   >
   > Die Methode, die Sie zum Hashen und Signieren der Richtlinienanweisung verwenden, ist abhängig von Ihrer Programmiersprache und Plattform. Einen Beispiel-Code finden Sie unter Code-Beispiele für das Erstellen einer Signatur für eine signierte URL.

2. Entfernen Sie die Leerzeichen (einschließlich Tabulatoren und Zeilenumbruchzeichen) aus der gehashten und signierten Zeichenfolge.
3. Base64-encode die Zeichenfolge, die die MIME-Base64-Codierung verwendet. Weitere Informationen finden Sie in Abschnitt 6.8, Base64 Content-Transfer-Encoding in RFC 2045, MIME (Multipurpose Internet Mail Extensions), Teil 1: Format von Internetnachrichtentexten.
4. Ersetzen Sie Zeichen, die in einer URL-Abfragezeichenfolge nicht gültig sind, durch gültige Zeichen. In der folgenden Tabelle sind ungültige und gültige Zeichen aufgelistet.

Ersetzen Sie diese ungültigen Zeichen	Durch diese gültigen Zeichen
+	- (Bindestrich)
=	_ (Unterstrich)
/	~ (Tilde)

5. Fügen Sie den resultierenden Wert hinter `&Signature=` zu Ihrer signierten URL hinzu und kehren Sie zu So erstellen Sie eine signierte URL mit einer vordefinierten Richtlinie zurück, um das Verketten der Teile Ihrer signierten URL abzuschließen.

Erstellen einer signierten URL mit einer benutzerdefinierten Richtlinie

Um eine signierte URL mit einer benutzerdefinierten Richtlinie zu erstellen, führen Sie die folgenden Schritte aus.

So erstellen Sie eine signierte URL mit einer benutzerdefinierten Richtlinie

1. Wenn Sie .NET oder Java verwenden, um signierte URLs zu erstellen, und den privaten Schlüssel für Ihr Schlüsselpaar noch nicht vom PEM-Standardformat in ein mit .NET oder Java kompatibles Format neu formatiert haben, holen Sie diesen Schritt jetzt nach. Weitere Informationen finden Sie unter Neuformatierung des privaten Schlüssels (nur .NET und Java).
2. Verketten Sie die folgenden Werte. Sie können das Format in diesem Beispiel für eine signierte URL verwenden.

```
https://d111111abcdef8.cloudfront.net/
image.jpg?color=red&size=medium&Policy=eyANCiAgICEXAMPLEW1lbnQiOiBbeyANCiAgICAgICJSZXNvdXJj
j19DzZrvDh6hQ73lDx~-ar3UocvvRQVw6EkC~GdpGQyyOSKQim-
TxAnW7d8F5Kkai9HVx0FIu-5jcQb0UEmatEXAMPLE3ReXySpLSMj0yCd3ZAB4UcBCAqEijkytL6f3fVYNGQI6&Key-
Pair-Id=K2JCJMDEHXQW5F&Hash-Algorithm=SHA256
```

   Entfernen Sie alle Leerzeichen (einschließlich Tabulatoren und Zeilenumbruchzeichen). Möglicherweise müssen Sie in der Zeichenfolge im Anwendungscode Escape-Zeichen einfügen. Alle Werte verfügen über `String` als Typ.

1. ***Base URL for the file***

   Die Basis-URL ist die CloudFront URL, die Sie für den Zugriff auf die Datei verwenden würden, wenn Sie keine signierten URLs verwenden würden, einschließlich Ihrer eigenen Abfragezeichenfolgenparameter, falls vorhanden. Im vorherigen Beispiel lautet die Basis-URL `https://d111111abcdef8.cloudfront.net/image.jpg`. Weitere Informationen zum Format der URLs für Distributionen finden Sie unter Passen Sie das URL-Format für Dateien in an CloudFront.

   Die folgenden Beispiele zeigen Werte, die Sie für Distributionen angeben.

   - Die folgende CloudFront URL bezieht sich auf eine Bilddatei in einer Distribution (unter Verwendung des CloudFront Domainnamens). Beachten Sie, dass `image.jpg` ein `images`-Verzeichnis ist. Der Pfad zur Datei in der URL muss mit dem Pfad zur Datei auf Ihrem HTTP-Server oder in Ihrem Amazon S3 Bucket übereinstimmen.

     ```
     https://d111111abcdef8.cloudfront.net/images/image.jpg
     ```

   - Die folgende CloudFront URL enthält eine Abfragezeichenfolge:

     ```
     https://d111111abcdef8.cloudfront.net/images/image.jpg?size=large
     ```

   - Die folgenden CloudFront URLs gelten für Bilddateien in einer Distribution.
     Beide verwenden einen alternativen Domänennamen; die zweite enthält eine Abfragezeichenfolge:

     ```
     https://www.example.com/images/image.jpg
     ```

     ```
     https://www.example.com/images/image.jpg?color=red
     ```

   - Die folgende CloudFront URL bezieht sich auf eine Bilddatei in einer Distribution, die einen alternativen Domainnamen und das HTTPS-Protokoll verwendet:

     ```
     https://www.example.com/images/image.jpg
     ```

2. **?**

   ? zeigt an, dass Abfragezeichenfolgeparameter hinter der Basis-URL angegeben sind. Schließen Sie das ? auch dann ein, wenn Sie keine Abfrageparameter angeben.

> **Note**
>
> Sie können die folgenden Abfrageparameter in beliebiger Reihenfolge angeben.

3. ***Your query string parameters, if any*&**

   (Optional) Sie können Ihre eigenen Abfragezeichenfolgenparameter eingeben. Fügen Sie dazu zwischen jedem ein Et-Zeichen (&) hinzu, wie in `color=red&size=medium`. Sie können die Abfragezeichenfolgenparameter in beliebiger Reihenfolge innerhalb der URL angeben.

   > **Important**
   >
   > Ihre Abfragezeichenfolgenparameter können nicht mit`Policy`, `Signature`, `Key-Pair-Id`, oder benannt werden`Hash-Algorithm`.

   Wenn Sie eigene Parameter hinzufügen, fügen Sie nach jedem Parameter ein & an, auch nach dem letzten.

4. **Policy=*base64 encoded version of policy statement***

   Ihre Richtlinienanweisung im JSON-Format, wobei Leerzeichen entfernt wurden und eine Base64-Codierung vorgenommen wurde. Weitere Informationen finden Sie unter Erstellen einer Richtlinienanweisung für eine signierte URL, die eine benutzerdefinierte Richtlinie verwendet.

   Die Richtlinienanweisung steuert den Zugriff, den eine signierte URL einem Benutzer gewährt. Sie enthält die URL der Datei, ein Ablaufdatum und eine Uhrzeit, ein optionales Datum und die Uhrzeit, zu der die URL gültig wird, und eine optionale IP-Adresse oder einen Bereich von IP-Adressen, die auf die Datei zugreifen dürfen.

5. **&Signature=*hashed and signed version of the policy statement***

   Eine gehashte, signierte und Base64-codierte Version der JSON-Richtlinienanweisung. Weitere Informationen finden Sie unter Erstellen einer Signatur für eine signierte URL, die eine benutzerdefinierte Richtlinie verwendet.

6. **&Key-Pair-Id=*public key ID for the CloudFront public key whose corresponding private key you're using to generate the signature***

   Die ID für einen CloudFront öffentlichen Schlüssel, zum BeispielK2JCJMDEHXQW5F. Die öffentliche Schlüssel-ID gibt an CloudFront, welcher öffentliche Schlüssel zur Validierung der signierten URL verwendet werden soll. CloudFrontvergleicht die Informationen in der Signatur mit den Informationen in der Grundsatzerklärung, um sicherzustellen, dass die URL nicht manipuliert wurde.

   Dieser öffentliche Schlüssel muss zu einer Schlüsselgruppe gehören, die ein vertrauenswürdiger Aussteller in der Distribution ist. Weitere Informationen finden Sie unter Festlegen der Signaturgeber, die signierte URLs und signierte Cookies erstellen können.

7. **&Hash-Algorithm=*SHA1 or SHA256***

   (Optional) Der Hash-Algorithmus, der zur Erstellung der Signatur verwendet wurde. Unterstützte Werte sind SHA1 und SHA256. Wenn Sie diesen Parameter nicht angeben, wird CloudFront standardmäßig der Wert verwendet. SHA1

Erstellen einer Richtlinienanweisung für eine signierte URL, die eine benutzerdefinierte Richtlinie verwendet

Führen Sie zum Erstellen einer Richtlinienanweisung für eine signierte URL, die eine benutzerdefinierte Richtlinie verwendet, die folgenden Schritte aus.

Einige Beispiele für Richtlinienanweisungen, die den Zugriff auf Dateien auf verschiedene Weisen kontrollieren, finden Sie unter the section called “Beispiel-Richtlinienanweisungen für eine signierte URL, die eine benutzerdefinierte Richtlinie verwendet”.

So erstellen Sie die Richtlinienanweisung für eine signierte URL, die eine benutzerdefinierte Richtlinie verwendet

1. Erstellen Sie die Richtlinienanweisung unter Verwendung des folgenden JSON-Formats. Ersetzen Sie die Symbole für „kleiner als“ (<) und „größer als“ (>) und die darin enthaltenen Beschreibungen durch Ihre eigenen Werte. Weitere Informationen finden Sie unter the section called “Werte, die Sie in der Richtlinienanweisung für eine signierte URL angeben, die eine benutzerdefinierte Richtlinie verwendet”.

   ```
   {
       "Statement": [
   ```

```
        {
            "Resource": "<Optional but recommended: URL of the file>",
            "Condition": {
                "DateLessThan": {
                 "AWS:EpochTime": <Required: ending date and time in Unix time
 format and UTC>
                },
                "DateGreaterThan": {
                 "AWS:EpochTime": <Optional: beginning date and time in Unix time
 format and UTC>
                },
                "IpAddress": {
                 "AWS:SourceIp": "<Optional: IP address>"
                }
            }
        }
    ]
}
```

Beachten Sie Folgendes:

- Sie können nur eine Anweisung in die Richtlinie aufnehmen.
- Verwenden Sie die UTF-8 Zeichenkodierung.
- Fügen Sie alle Satzzeichen und Parameternamen genau wie angegeben ein. Abkürzungen für Parameternamen werden nicht akzeptiert.
- Die Reihenfolge der Parameter im Bereich `Condition` ist unerheblich.
- Informationen zu den Werten für `Resource`, `DateLessThan`, `DateGreaterThan` und `IpAddress` finden Sie unter the section called “Werte, die Sie in der Richtlinienanweisung für eine signierte URL angeben, die eine benutzerdefinierte Richtlinie verwendet”.

2. Entfernen Sie alle Leerzeichen (einschließlich Tabulatoren und Zeilenumbruchzeichen) aus der Richtlinienanweisung. Möglicherweise müssen Sie in der Zeichenfolge im Anwendungscode Escape-Zeichen einfügen.
3. Base64-encode die Richtlinienanweisung, die die MIME-Base64-Kodierung verwendet. Weitere Informationen finden Sie in Abschnitt 6.8, Base64 Content-Transfer-Encoding in RFC 2045, MIME (Multipurpose Internet Mail Extensions), Teil 1: Format von Internetnachrichtentexten.
4. Ersetzen Sie Zeichen, die in einer URL-Abfragezeichenfolge nicht gültig sind, durch gültige Zeichen. In der folgenden Tabelle sind ungültige und gültige Zeichen aufgelistet.

Ersetzen Sie diese ungültigen Zeichen	Durch diese gültigen Zeichen
+	- (Bindestrich)
=	_ (Unterstrich)
/	~ (Tilde)

5. Fügen Sie den resultierenden Wert hinter zu Ihrer signierten URL hinz `Policy=`.
6. Erstellen Sie eine Signatur für die signierte URL, indem Sie die Richtlinienanweisung hashen, signieren und eine Base64-Codierung vornehmen. Weitere Informationen finden Sie unter the section called “Erstellen einer Signatur für eine signierte URL, die eine benutzerdefinierte Richtlinie verwendet”.

Werte, die Sie in der Richtlinienanweisung für eine signierte URL angeben, die eine benutzerdefinierte Richtlinie verwendet

Beim Erstellen einer Richtlinienanweisung für eine benutzerdefinierte Richtlinie geben Sie die folgenden Werte an.

Ressource

Die URL, einschließlich aller Abfragezeichenfolgen, aber ohne die Parameter CloudFront `Policy`,`Signature`, `Key-Pair-Id` und. `Hash-Algorithm` Beispiel:

```
https://d111111abcdef8.cloudfront.net/images/horizon.jpg\?
size=large&license=yes
```

Sie können nur einen URL-Wert für `Resource` angeben.

> **Important**
>
> Sie können den `Resource`-Parameter in einer Richtlinie weglassen. Dies bedeutet jedoch, dass alle, die über die signierte URL verfügen, auf alle Dateien in jeder Distribution zugreifen können, die mit dem zum Erstellen der signierten URL verwendeten Schlüsselpaar verknüpft ist.

Beachten Sie Folgendes:

- Protokoll – Der Wert muss mit `http://`, `https://` oder `*://` beginnen.
- Parameter für Abfragezeichenfolgen — Wenn die URL Abfragezeichenfolgenparameter enthält, verwenden Sie keinen umgekehrten Schrägstrich (), um das Fragezeichen (`\?`) zu maskieren, mit dem die Abfragezeichenfolge beginnt. Beispiel:

  ```
  https://d111111abcdef8.cloudfront.net/images/horizon.jpg?
  size=large&license=yes
  ```
- Platzhalterzeichen – Sie können Platzhalterzeichen in der URL in der Richtlinie verwenden. Die folgenden Platzhalterzeichen werden unterstützt:
  - Sternchen (`*`) für null oder mehr Zeichen.
  - Fragezeichen (`?`) für genau ein Zeichen

  Wenn die URL in der Richtlinie mit der URL in der HTTP-Anforderung CloudFront übereinstimmt, wird die URL in der Richtlinie wie folgt in vier Abschnitte unterteilt: Protokoll, Domäne, Pfad und Abfragezeichenfolge:

  ```
  [protocol]://[domain]/[path]\?[query string]
  ```

  Wenn Sie in der URL in der Richtlinie ein Platzhalterzeichen verwenden, erfolgt der Platzhalterabgleich nur innerhalb der Grenzen des Abschnitts, der den Platzhalter enthält. Betrachten Sie etwa diese URL in einer Richtlinie:

  ```
  https://www.example.com/hello*world
  ```

  In diesem Beispiel gilt der Sternchen-Platzhalter (`*`) nur innerhalb des Pfadabschnitts, entspricht also den URLs `https://www.example.com/helloworld` und `https://www.example.com/hello-world`, nicht jedoch der URL `https://www.example.net/hello?world`.

  Die folgenden Ausnahmen gelten beim Platzhalterabgleich für die Abschnittsgrenzen:

  - Ein nachfolgendes Sternchen im Pfadabschnitt impliziert ein Sternchen im Abschnitt mit der Abfragezeichenfolge. Beispiel: `http://example.com/hello*` ist gleichbedeutend mit `http://example.com/hello*\?*`.
  - Ein nachfolgendes Sternchen im Domainabschnitt impliziert ein Sternchen sowohl im Pfad- als auch im Abschnitt mit der Abfragezeichenfolge. Beispiel: `http://example.com*` ist gleichbedeutend mit `http://example.com*/*\?*`.

- Eine URL in der Richtlinie kann den Protokollabschnitt weglassen und im Domainabschnitt mit einem Sternchen beginnen. In diesem Fall wird der Protokollabschnitt implizit auf ein Sternchen gesetzt. Beispielsweise entspricht die URL `*example.com` in einer Richtlinie `*://*example.com/`.
- Ein Sternchen an sich (`"Resource": "*"`) entspricht jeder URL.

Beispielsweise entspricht der Wert `https://d111111abcdef8.cloudfront.net/*game_download.zip*` in einer Richtlinie allen folgenden URLs:

- `https://d111111abcdef8.cloudfront.net/game_download.zip`
- `https://d111111abcdef8.cloudfront.net/example_game_download.zip?license=yes`
- `https://d111111abcdef8.cloudfront.net/test_game_download.zip?license=temp`

Percent-encoded Zeichen — CloudFront gleicht die Richtlinie `Resource` mit der unformatierten (nicht dekodierten) Anforderungs-URL ab und leitet diese URL unverändert an den Ursprung weiter. Ein in Prozent kodiertes Zeichen in der Anforderungs-URL, z. B. für, `%3B` für oder `%3F` `%23` für `;` `?#`, wird als Daten und nicht als URL-Trennzeichen behandelt. Weitere Informationen finden Sie unter RFC 3986 auf der IETF-Website. Beispielsweise kann ein Platzhalter nicht mit einem literalen Semikolon (`;`) übereinstimmen, aber er kann mit der gesamten codierten Form () übereinstimmen. `%3B`

- Alternative Domainnamen – Wenn Sie einen alternativen Domainnamen (CNAME) in der URL in der Richtlinie angeben, muss die HTTP-Anfrage diesen alternativen Domainnamen auf Ihrer Webseite oder in Ihrer Anwendung verwenden. Geben Sie nicht die Amazon-S3-URL für die Datei in einer Richtlinie an.

DateLessThan

Das Ablaufdatum und die Ablaufzeit für die URL im Unix-Zeitformat (in Sekunden) und in koordinierter Weltzeit (UTC). Setzen Sie in der Richtlinie den Wert nicht in Anführungszeichen. Weitere Informationen zu UTC finden Sie unter Datum und Uhrzeit im Internet: Zeitstempel.

Beispielsweise wird der 31. Januar 2023, 10:00 Uhr UTC, im Unix-Zeitformat in 1675159200 konvertiert.

Dies ist der einzige erforderliche Parameter in dem Abschnitt. `Condition` CloudFront benötigt diesen Wert, um zu verhindern, dass Benutzer dauerhaft auf Ihre privaten Inhalte zugreifen können.

Weitere Informationen finden Sie unter the section called “Wenn CloudFront überprüft das Ablaufdatum und die Uhrzeit einer signierten URL”.

DateGreaterThan (Fakultativ)

Ein optionales Datum und eine optionale Zeit für die URL im Unix-Zeitformat (in Sekunden) und in koordinierter Weltzeit (UTC). Benutzer sind an oder vor dem angegebenen Datum und der angegebenen Zeit nicht berechtigt, auf die Datei zuzugreifen. Setzen Sie den Wert nicht in Anführungszeichen.

IpAddress (Fakultativ)

Die IP-Adresse des Clients, der die HTTP-Anfrage stellt. Beachten Sie Folgendes:

- Um allen IP-Adressen den Zugriff auf die Datei zu gewähren, lassen Sie den Parameter `IpAddress` weg.
- Sie können entweder eine IP-Adresse oder einen IP-Adressbereich angeben. Sie können die Richtlinie nicht zum Gewähren von Zugriff verwenden, wenn sich die IP-Adresse des Clients in einem von zwei getrennten Bereichen befindet.
- Um den Zugriff von einer einzigen IP-Adresse zu gewähren, geben Sie Folgendes an:

  `"`*`IPv4 IP address`*`/32"`
- Sie müssen IP-Adressbereiche im IPv4-CIDR-Standardformat angeben (z. B, `192.0.2.0/24`). Weitere Informationen finden Sie unter Classless Inter-domain Routing (CIDR): Der Plan zur Zuweisung und Aggregation von Internetadressen.

> **Important**
>
> IP-Adressen im IPv6-Format, wie beispielsweise 2001:0db8:85a3::8a2e:0370:7334, werden nicht unterstützt.

Wenn Sie eine benutzerdefinierte Richtlinie verwenden, die `IpAddress` enthält, aktivieren Sie IPv6 nicht für die Distribution. Wenn Sie den Zugriff auf bestimmte Inhalte nach IP-Adresse einschränken und IPv6-Anforderungen für andere Inhalte unterstützen möchten, können Sie zwei Distributionen erstellen. Weitere Informationen finden Sie unter the section called “Aktivieren von IPv6 (Viewer-Anforderungen)” im Thema the section called “Alle Distributionseinstellungen”.

Beispiel-Richtlinienanweisungen für eine signierte URL, die eine benutzerdefinierte Richtlinie verwendet

Die folgenden Beispiel-Richtlinienanweisungen zeigen, wie der Zugriff auf eine bestimmte Datei, auf alle Dateien in einem Verzeichnis oder auf alle mit einer Schlüsselpaar-ID verknüpften Dateien kontrolliert wird. Die Beispiele zeigen auch, wie der Zugriff von einer einzelnen IP-Adresse oder einem Bereich von IP-Adressen kontrolliert wird und wie Sie verhindern, dass Benutzer die signierte URL nach einem festgelegten Datum und einer festgelegten Zeit verwenden.

Wenn Sie eines dieser Beispiele kopieren und einfügen, entfernen Sie alle Leerzeichen (einschließlich Tabuloren und Zeilenumbruchzeichen), ersetzen Sie die entsprechenden Werte durch Ihre eigenen Werte und fügen Sie ein Zeilenumbruchzeichen hinter der schließenden Klammer (`}`) ein.

Weitere Informationen finden Sie unter the section called “Werte, die Sie in der Richtlinienanweisung für eine signierte URL angeben, die eine benutzerdefinierte Richtlinie verwendet”.

Themen

- Beispiel-Richtlinienanweisung: Zugriff auf eine Datei von einem Bereich von IP-Adressen aus
- Beispiel-Richtlinienanweisung: Zugriff auf alle Dateien in einem Verzeichnis über einen Bereich von IP-Adressen
- Beispiel-Richtlinienanweisung: Zugriff auf alle mit einer Schlüsselpaar-ID verknüpften Dateien über eine IP-Adresse

Beispiel-Richtlinienanweisung: Zugriff auf eine Datei von einem Bereich von IP-Adressen aus

Das folgende Beispiel für eine Richtlinienanweisung in einer signierten URL legt fest, dass ein Benutzer bis zum 31. Januar 2013, 10:00 Uhr UTC, von IP-Adressen im Bereich `192.0.2.0/24` aus auf die Datei `https://d111111abcdef8.cloudfront.net/game_download.zip` zugreifen kann:

```
{
    "Statement": [
        {
            "Resource": "https://d111111abcdef8.cloudfront.net/game_download.zip",
            "Condition": {
                "IpAddress": {
                    "AWS:SourceIp": "192.0.2.0/24"
                },
```

```
                "DateLessThan": {
                    "AWS:EpochTime": 1675159200
                }
            }
        }
    ]
}
```

Beispiel-Richtlinienanweisung: Zugriff auf alle Dateien in einem Verzeichnis über einen Bereich von IP-Adressen

Mit dem folgenden Beispiel für eine benutzerdefinierte Richtlinie können Sie signierte URLs für jede Datei im `training` Verzeichnis erstellen, was durch das Sternchen-Platzhalterzeichen () * im Parameter gekennzeichnet ist. `Resource` Benutzer können bis zum 31. Januar 2013, 10:00 Uhr UTC, von IP-Adressen im Bereich `192.0.2.0/24` aus auf die Datei zugreifen:

```
{
    "Statement": [
        {
            "Resource": "https://d111111abcdef8.cloudfront.net/training/*",
            "Condition": {
                "IpAddress": {
                    "AWS:SourceIp": "192.0.2.0/24"
                },
                "DateLessThan": {
                    "AWS:EpochTime": 1675159200
                }
            }
        }
    ]
}
```

Jede signierte URL, mit der Sie diese Richtlinie verwenden, enthält eine URL, die eine bestimmte Datei identifiziert, zum Beispiel:

`https://d111111abcdef8.cloudfront.net/training/orientation.pdf`

Beispiel-Richtlinienanweisung: Zugriff auf alle mit einer Schlüsselpaar-ID verknüpften Dateien über eine IP-Adresse

Mit dem folgenden Beispiel für eine benutzerdefinierte Richtlinie können Sie signierte URLs für jede Datei erstellen, die einer beliebigen Distribution zugeordnet ist. Dies wird durch das

Sternchen-Platzhalterzeichen (*) im Parameter angezeigt. `Resource` Die signierte URL muss das `https://`-Protokoll verwenden, nicht `http://`. Der Benutzer muss die IP-Adresse verwende `192.0.2.10/32`. (Der Wert `192.0.2.10/32` in CIDR-Notation bezieht sich auf eine einzelne IP-Adresse, `192.0.2.10`.) Die Dateien sind nur vom 31. Januar 2023, 10:00 Uhr UTC, bis zum 2. Februar 2023, 10:00 Uhr UTC, verfügbar:

```
{
    "Statement": [
       {
            "Resource": "https://*",
            "Condition": {
                "IpAddress": {
                    "AWS:SourceIp": "192.0.2.10/32"
                },
                "DateGreaterThan": {
                    "AWS:EpochTime": 1675159200
                },
                "DateLessThan": {
                    "AWS:EpochTime": 1675332000
                }
            }
         }
    ]
}
```

Jede signierte URL, mit der Sie diese Richtlinie verwenden, hat eine URL, die eine bestimmte Datei in einer bestimmten CloudFront Distribution identifiziert, zum Beispiel:

`https://d111111abcdef8.cloudfront.net/training/orientation.pdf`

Die signierte URL enthält auch eine Schlüsselpaar-ID, die mit einer vertrauenswürdigen Schlüsselgruppe in der Distribution (d111111abcdef8.cloudfront.net) verknüpft werden muss, die Sie in der URL angeben.

Erstellen einer Signatur für eine signierte URL, die eine benutzerdefinierte Richtlinie verwendet

Bei der Signatur für eine signierte URL, die eine benutzerdefinierte Richtlinie verwendet, handelt es sich um eine gehashte, signierte und Base64-codierte Version der Richtlinienanweisung. Führen Sie die folgenden Schritte aus, um eine Signatur für eine benutzerdefinierte Richtlinie zu erstellen.

Weitere Informationen und Beispiele für das Hashing, Signieren und Codieren der Richtlinienanweisung finden Sie unter:

- Linux-Befehle und OpenSSL für die Base64-Codierung und die Verschlüsselung
- Code-Beispiele für das Erstellen einer Signatur für eine signierte URL

> **Note**
>
> Die verlinkten Beispiele SHA-1 werden standardmäßig verwendet. Um es SHA-256 stattdessen zu verwenden, `sha1` ersetzen Sie es `sha256` in den OpenSSL-Befehlen und fügen Sie den `Hash-Algorithm=SHA256` Abfrageparameter in die signierte URL ein.

Option 1: So erstellen Sie eine Signatur mithilfe einer benutzerdefinierten Richtlinie

1. Verwenden Sie die SHA-256 Hash-Funktion SHA-1 oder und den generierten privaten RSA- oder ECDSA-Schlüssel, um die JSON-Richtlinienanweisung, die Sie in der Prozedur erstellt haben, zu hashen und zu signieren. So erstellen Sie die Richtlinienanweisung für eine signierte URL, die eine benutzerdefinierte Richtlinie verwendet Verwenden Sie die Version der Richtlinienanweisung, die keine Leerzeichen mehr enthält, jedoch noch nicht Base64-codiert wurde.

   Wenn Sie es verwenden SHA-256, müssen Sie es `&Hash-Algorithm=SHA256` in die signierte URL aufnehmen.

   Verwenden Sie für den privaten Schlüssel, der für die Hash-Funktion erforderlich ist, einen privaten Schlüssel, dessen öffentlicher Schlüssel sich in einer aktiven vertrauenswürdigen Schlüsselgruppe für die Distribution befindet.

   > **Note**
   >
   > Die Methode, die Sie zum Hashen und Signieren der Richtlinienanweisung verwenden, ist abhängig von Ihrer Programmiersprache und Plattform. Einen Beispiel-Code finden Sie unter Code-Beispiele für das Erstellen einer Signatur für eine signierte URL.

2. Entfernen Sie die Leerzeichen (einschließlich Tabulatoren und Zeilenumbruchzeichen) aus der gehashten und signierten Zeichenfolge.
3. Base64-encode die Zeichenfolge, die die MIME-Base64-Codierung verwendet. Weitere Informationen finden Sie in Abschnitt 6.8, Base64 Content-Transfer-Encoding in RFC 2045, MIME (Multipurpose Internet Mail Extensions), Teil 1: Format von Internetnachrichtentexten.

4. Ersetzen Sie Zeichen, die in einer URL-Abfragezeichenfolge nicht gültig sind, durch gültige Zeichen. In der folgenden Tabelle sind ungültige und gültige Zeichen aufgelistet.

Ersetzen Sie diese ungültigen Zeichen	Durch diese gültigen Zeichen
+	- (Bindestrich)
=	_ (Unterstrich)
/	~ (Tilde)

5. Fügen Sie den resultierenden Wert hinter `&Signature=` zu Ihrer signierten URL hinzu und kehren Sie zu So erstellen Sie eine signierte URL mit einer benutzerdefinierten Richtlinie zurück, um das Verketten der Teile Ihrer signierten URL abzuschließen.

Verwenden signierter Cookies

CloudFront Mit signierten Cookies können Sie kontrollieren, wer auf Ihre Inhalte zugreifen kann, wenn Sie Ihre aktuellen URLs nicht ändern möchten oder wenn Sie Zugriff auf mehrere eingeschränkte Dateien gewähren möchten, z. B. auf alle Dateien im Abonnentenbereich einer Website. In diesem Thema werden die Überlegungen bei der Verwendung von signierten Cookies erläutert und es wird beschrieben, wie signierte Cookies mithilfe von vordefinierten und benutzerdefinierten Richtlinien eingerichtet werden.

Themen

- Entscheidung zwischen vordefinierten und benutzerdefinierten Richtlinien für signierte Cookies
- Funktionsweise von signierten Cookies
- Verhindern der missbräuchlichen Verwendung von signierten Cookies
- Wann CloudFront überprüft das Ablaufdatum und die Uhrzeit in einem signierten Cookie
- Beispiel-Code und Drittanbieter-Tools
- Einrichten signierter Cookies mit einer vordefinierten Richtlinie
- Einrichten signierter Cookies mit einer benutzerdefinierten Richtlinie
- Erstellen signierter Cookies mit PHP

Entscheidung zwischen vordefinierten und benutzerdefinierten Richtlinien für signierte Cookies

Wenn Sie ein signiertes Cookie erstellen, schreiben Sie eine Richtlinienanweisung im JSON-Format, welche die Einschränkungen für das signierte Cookie festlegt, z. B. wie lange das Cookie gültig ist. Sie können vordefinierte oder benutzerdefinierte Richtlinien verwenden. In der folgenden Tabelle werden vordefinierte und benutzerdefinierte Richtlinien verglichen:

Beschreibung	Vordefinierte Richtlinie	Benutzerdefinierte Richtlinie
Sie können die Richtlinienanweisung für mehrere Dateien wiederverwenden. Um die Richtlini enanweisung wiederzuverwenden, müssen Sie Platzhalterzeichen im `Resource`-Objekt verwenden . Weitere Informationen finden Sie unter Werte, die Sie in der Richtlinienanweisung für eine benutzerd efinierte Richtlinie für signierte Cookies angeben.)	Nein	Ja
Sie können das Datum und die Zeit festlegen, ab denen Benutzer auf Ihre Inhalte zugreifen können.	Nein	Ja (optional)
Sie können das Datum und die Zeit festlegen, ab denen Benutzer nicht mehr auf Ihre Inhalte zugreifen können.	Ja	Ja
Sie können die IP-Adresse oder den Bereich von IP-Adressen der Benutzer festlegen, die auf Ihre Inhalte zugreifen können.	Nein	Ja (optional)

Informationen zum Erstellen von signierten Cookies mit einer vordefinierten Richtlinie finden Sie unter Einrichten signierter Cookies mit einer vordefinierten Richtlinie.

Informationen zum Erstellen von signierten Cookies mit einer benutzerdefinierten Richtlinie finden Sie unter Einrichten signierter Cookies mit einer benutzerdefinierten Richtlinie.

Funktionsweise von signierten Cookies

Hier finden Sie eine Übersicht darüber, wie Sie signierte Cookies konfigurieren CloudFront und wie CloudFront reagiert wird, wenn ein Benutzer eine Anfrage sendet, die ein signiertes Cookie enthält.

1. Geben Sie in Ihrer CloudFront Distribution eine oder mehrere vertrauenswürdige Schlüsselgruppen an, die die öffentlichen Schlüssel enthalten, die zur Überprüfung der URL-Signatur verwendet werden CloudFront können. Sie verwenden die entsprechenden privaten Schlüssel, um die URLs zu signieren.

   Weitere Informationen finden Sie unter Festlegen der Signaturgeber, die signierte URLs und signierte Cookies erstellen können.
2. Sie entwickeln Ihre Anwendung so, dass ermittelt wird, ob ein Benutzer über Zugriff auf Ihre Inhalte verfügen sollte, und wenn dies der Fall ist, drei `Set-Cookie`-Header an den Viewer zu senden. (Jeder `Set-Cookie` Header kann nur ein Name-Wert-Paar enthalten, und ein CloudFront signiertes Cookie benötigt drei Name-Wert-Paare.) Sie müssen die `Set-Cookie`-Header an den Viewer senden, bevor der Viewer Ihre privaten Inhalte anfragt. Wenn Sie eine kurze Ablaufzeit für das Cookie angeben, möchten Sie vielleicht auch drei weitere `Set-Cookie`-Header als Reaktion auf folgende Anfragen senden, sodass der Benutzer weiterhin Zugriff hat.

   In der Regel verfügt Ihre CloudFront Distribution über mindestens zwei Cache-Verhaltensweisen, eines, für das keine Authentifizierung erforderlich ist, und eines, das eine Authentifizierung erfordert. Die Fehlerseite für den sicheren Teil der Website enthält eine Weiterleitung oder einen Link zu einer Anmeldeseite.

   Wenn Sie Ihre Distribution so konfigurieren, dass Dateien auf der Grundlage von Cookies zwischengespeichert werden, werden CloudFront keine separaten Dateien zwischengespeichert, die auf den Attributen in signierten Cookies basieren.
3. Ein Benutzer meldet sich auf Ihrer Website an und bezahlt entweder für Inhalte oder erfüllt andere Anforderungen für den Zugriff.
4. Ihre Anwendung gibt die `Set-Cookie`-Header in der Antwort zurück und der Viewer speichert die Name-Wert-Paare.
5. Der Benutzer fordert eine Datei an.

   Der Browser des Benutzers oder ein anderer Viewer ruft die Name-Wert-Paare aus Schritt 4 ab und fügt sie zu der Anfrage in einem `Cookie`-Header hinzu. Dies ist das signierte Cookie.

6. CloudFront verwendet den öffentlichen Schlüssel, um die Signatur im signierten Cookie zu validieren und zu bestätigen, dass das Cookie nicht manipuliert wurde. Wenn die Signatur ungültig ist, wird die Anfrage abgelehnt.

   Wenn die Signatur im Cookie gültig ist, überprüft CloudFront die Richtlinienerklärung im Cookie (oder erstellt eine, wenn Sie eine vordefinierte Richtlinie verwenden), um zu bestätigen, dass die Anfrage noch gültig ist. Wenn Sie beispielsweise ein Anfangs- und Enddatum sowie eine Uhrzeit für das Cookie angegeben haben, CloudFront bestätigt dies, dass der Benutzer versucht, während des Zeitraums, den Sie den Zugriff zulassen möchten, auf Ihre Inhalte zuzugreifen.

   Wenn die Anfrage die Anforderungen in der Richtlinienerklärung CloudFront erfüllt, stellt es Ihren Inhalt wie für Inhalte bereit, für die keine Einschränkungen gelten: Es wird festgestellt, ob sich die Datei bereits im Edge-Cache befindet, leitet die Anfrage gegebenenfalls an den Ursprung weiter und gibt die Datei an den Benutzer zurück.

> **Note**
>
> CloudFront erfordert keine signierte URL oder signierte Cookies für `OPTIONS` Anfragen. Dadurch können CORS-Preflight-Anfragen, die Browser ohne Anmeldeinformationen senden, bei Cacheverhalten, für die signierte URLs oder signierte Cookies erforderlich sind, erfolgreich sein. Leitet `OPTIONS` Anfragen ohne Überprüfung einer Signatur an Ihren Ursprung CloudFront weiter`OPTIONS`, sofern das Cache-Verhalten Folgendes zulässt. Stellen Sie sicher, dass Ihr Ursprung als Antwort auf `OPTIONS` Anfragen keine geschützten Inhalte zurückgibt, oder beschränken Sie die zulässigen Methoden auf `GET` und `HEAD` falls Ihre Anwendung CORS nicht verwendet.

Verhindern der missbräuchlichen Verwendung von signierten Cookies

Wenn Sie den `Domain`-Parameter in einem `Set-Cookie`-Header angeben, geben Sie einen möglichst genauen Wert ein, um das Potenzial für den Zugriff durch einen Benutzer mit demselben Stammdomänennamen zu verringern. Beispielsweise ist app.example.com gegenüber example.com vorzuziehen – insbesondere, wenn Sie example.com nicht kontrollieren. Dadurch können Sie verhindern, dass ein Benutzer von www.example.com aus auf Ihre Inhalte zugreift.

Gehen Sie wie folgt vor, um diese Art von Angriff zu verhindern:

- Schließen Sie die Cookie-Attribute `Expires` und `Max-Age` aus, damit der `Set-Cookie`-Header ein Sitzungs-Cookie erstellt. Sitzungs-Cookies werden automatisch gelöscht, wenn der Benutzer den Browser schließt. Dies verringert das Risiko, dass ein Benutzer unbefugten Zugriff auf Ihre Inhalte erhält.
- Fügen Sie das Attribut `Secure` ein, damit das Cookie verschlüsselt wird, wenn ein Viewer es in eine Anfrage einfügt.
- Verwenden Sie wenn möglich eine benutzerdefinierte Richtlinie und fügen Sie die IP-Adresse des Viewers ein.
- Geben Sie auf der Grundlage davon, wie lange Sie Benutzern Zugriff auf Ihre Inhalte gewähren möchten, im Attribut `CloudFront-Expires` eine möglichst kurze, vernünftige Ablaufzeit an.

Wann CloudFront überprüft das Ablaufdatum und die Uhrzeit in einem signierten Cookie

Um festzustellen, ob ein signiertes Cookie noch gültig ist, werden das Ablaufdatum und die Uhrzeit im Cookie zum Zeitpunkt der HTTP-Anfrage CloudFront überprüft. Wenn ein Client unmittelbar vor der Ablaufzeit mit dem Download einer großen Datei beginnt, sollte der Download abgeschlossen werden, auch wenn die Ablaufzeit während des Downloads überschritten wird. Wenn die TCP-Verbindung getrennt wird und der Client nach Überschreitung der Ablaufzeit versucht, den Download erneut zu starten, schlägt der Download fehl.

Wenn ein Client Bereichs-GETs verwendet, um eine Datei in kleineren Teilen abzurufen, schlagen alle GET-Anfragen nach Überschreitung der Ablaufzeit fehl. Weitere Informationen zu Bereichs-GETs finden Sie unter Wie CloudFront verarbeitet Teilanfragen für ein Objekt (Bereichs-GETs).

Beispiel-Code und Drittanbieter-Tools

Der Beispiel-Code für private Inhalte zeigt nur die Erstellung der Signatur für signierte URLs. Das Erstellen einer Signatur für ein signiertes Cookie ist jedoch sehr ähnlich, deshalb ist ein Großteil des Beispiel-Codes auch hier von Bedeutung. Weitere Informationen finden Sie unter den folgenden Themen:

- Erstellen einer URL-Signatur mit Perl
- Erstellen einer URL-Signatur mit PHP
- Erstellen einer URL-Signatur mithilfe von C# und dem .NET Framework
- Erstellen einer URL-Signatur mit Java

Einrichten signierter Cookies mit einer vordefinierten Richtlinie

Um ein signiertes Cookie mit einer vordefinierten Richtlinie einzurichten, führen Sie die folgenden Schritte aus. Zum Erstellen einer Signatur vgl. Erstellen einer Signatur für ein signiertes Cookie, das eine vordefinierte Richtlinie verwendet.

So richten Sie ein signiertes Cookies mit einer vordefinierten Richtlinie ein

1. Wenn Sie .NET oder Java verwenden, um signierte Cookies zu erstellen, und den privaten Schlüssel für Ihr Schlüsselpaar noch nicht vom PEM-Standardformat in ein mit .NET oder Java kompatibles Format neu formatiert haben, holen Sie diesen Schritt jetzt nach. Weitere Informationen finden Sie unter Neuformatierung des privaten Schlüssels (nur .NET und Java).
2. Programmieren Sie Ihre Anwendung so, dass drei `Set-Cookie` Header an zugelassene Zuschauer gesendet werden (oder vier, wenn Sie einen Hash-Algorithmus angeben möchten). Sie benötigen drei `Set-Cookie`-Header, weil jeder `Set-Cookie`-Header nur ein Name-Wert-Paar enthalten kann und ein CloudFront -signiertes Cookie drei Name-Wert-Paare erfordert. Die Name-Wert-Paare sind: `CloudFront-Expires`, `CloudFront-Signature` und `CloudFront-Key-Pair-Id`. Sie können optional ein viertes Name-Wert-Paar hinzufügen`CloudFront-Hash-Algorithm`, um den für die Signatur verwendeten Hash-Algorithmus anzugeben. Die Werte müssen auf dem Viewer vorhanden sein, bevor ein Benutzer die erste Anfrage für eine Datei stellt, bei der der Zugriff kontrolliert werden soll.

   > **Note**
   >
   > Im Allgemeinen empfehlen wir, die Attribute `Expires` und `Max-Age` auszuschließen. Der Ausschluss der Attribute bewirkt, dass der Browser das Cookie löscht, wenn der Benutzer den Browser schließt. Dies verringert das Risiko, dass ein Benutzer unbefugten Zugriff auf Ihre Inhalte erhält. Weitere Informationen finden Sie unter Verhindern der missbräuchlichen Verwendung von signierten Cookies.

   Bei den Namen der Cookie-Attribute muss die Groß- und Kleinschreibung beachtet werden.

   Zeilenumbrüche werden nur hinzugefügt, damit die Attribute besser lesbar sind.

   ```
   Set-Cookie:
   CloudFront-Expires=date and time in Unix time format (in seconds) and Coordinated
    Universal Time (UTC);
   Domain=optional domain name;
   ```

```
Path=/optional directory path;
Secure;
HttpOnly

Set-Cookie:
CloudFront-Signature=hashed and signed version of the policy statement;
Domain=optional domain name;
Path=/optional directory path;
Secure;
HttpOnly

Set-Cookie:
CloudFront-Key-Pair-Id=public key ID for the CloudFront public key whose
 corresponding private key you're using to generate the signature;
Domain=optional domain name;
Path=/optional directory path;
Secure;
HttpOnly

Set-Cookie:
CloudFront-Hash-Algorithm=SHA1 or SHA256;
Domain=optional domain name;
Path=/optional directory path;
Secure;
HttpOnly
```

(Optional) **`Domain`**

Der Domänenname für die angeforderte Datei. Wenn Sie kein `Domain`-Attribut angeben, ist der Standardwert der Domänenname in der URL und dieser gilt nur für die angegebenen Domänennamen, nicht für Unterdomänen. Wenn Sie ein `Domain`-Attribut angeben, gilt dieses auch für Unterdomänen. Ein vorangestellter Punkt im Domänennamen (z. B. `Domain=.example.com`) ist optional. Wenn Sie ein `Domain`-Attribut angeben, müssen darüber hinaus der Domänenname in der URL und der Wert des `Domain`-Attributs übereinstimmen.

Sie können den Domainnamen angeben, der Ihrer Distribution CloudFront zugewiesen wurde, z. B. d111111abcdef8.cloudfront.net, aber Sie können nicht *.cloudfront.net als Domainnamen angeben.

Wenn Sie einen alternativen Domänennamen wie beispielsweise example.com in URLs verwenden möchten, müssen Sie den alternativen Domänennamen zu Ihrer Distribution hinzufügen, unabhängig davon, ob Sie das `Domain`-Attribut angeben. Weitere Informationen finden Sie unter Alternative Domänennamen (CNAMEs) im Thema Referenz für alle Distributionseinstellungen.

(Optional) **`Path`**

Der Pfad für die angeforderte Datei. Wenn Sie kein `Path`-Attribut angeben, ist der Standardwert der Pfad in der URL.

`Secure`

Macht es erforderlich, dass der Viewer Cookies vor dem Senden einer Anfrage verschlüsselt. Wir empfehlen, den `Set-Cookie` Header über eine HTTPS-Verbindung zu senden, um sicherzustellen, dass die Cookie-Attribute vor Man-in-the-Middle-Angriffen geschützt sind.

`HttpOnly`

Definiert, wie der Browser (sofern unterstützt) mit dem Cookie-Wert interagiert. Mit sind `HttpOnly` die Cookie-Werte für nicht zugänglich. JavaScript Diese Vorsichtsmaßnahme kann dazu beitragen, Cross-Site-Scripting (XSS)-Angriffe abzuwehren. Weitere Informationen finden Sie unter Verwenden von HTTP-Cookies.

`CloudFront-Expires`

Geben Sie das Ablaufdatum und die Ablaufzeit im Unix-Zeitformat (in Sekunden) und in koordinierter Weltzeit (UTC) an. Beispielsweise wird der 1. Januar 2026, 10:00 Uhr UTC im Unix-Zeitformat in 1767290400 umgewandelt.

Um die Epochenzeit zu verwenden, geben Sie eine 64-Bit-Ganzzahl für ein Datum an, das nicht nach `9223372036854775807` (Freitag, 11. April 2262 um 23:47:16.854 UTC) liegt.

Weitere Informationen zu UTC finden Sie unter RFC 3339, Datum und Uhrzeit im Internet: Zeitstempel, https://tools.ietf.org/html/rfc3339.

`CloudFront-Signature`

Eine gehashte, signierte und Base64-codierte Version einer JSON-Richtlinienanweisung. Weitere Informationen finden Sie unter Erstellen einer Signatur für ein signiertes Cookie, das eine vordefinierte Richtlinie verwendet.

`CloudFront-Key-Pair-Id`

Die ID für einen CloudFront öffentlichen Schlüssel, zum Beispiel. `K2JCJMDEHXQW5F` Die öffentliche Schlüssel-ID gibt an CloudFront , welcher öffentliche Schlüssel zur Validierung der signierten URL verwendet werden soll. CloudFront vergleicht die Informationen in der Signatur mit den Informationen in der Grundsatzerklärung, um sicherzustellen, dass die URL nicht manipuliert wurde.

Dieser öffentliche Schlüssel muss zu einer Schlüsselgruppe gehören, die ein vertrauenswürdiger Aussteller in der Distribution ist. Weitere Informationen finden Sie unter Festlegen der Signaturgeber, die signierte URLs und signierte Cookies erstellen können.

`CloudFront-Hash-Algorithm`

(Optional) Der Hash-Algorithmus, der zur Erstellung der Signatur verwendet wurde. Unterstützte Werte sind `SHA1` und `SHA256`. Wenn Sie dieses Cookie nicht hinzufügen, ist es CloudFront standardmäßig auf `SHA1` eingestellt.

Das folgende Beispiel zeigt `Set-Cookie`-Header für ein signiertes Cookie bei Verwendung des mit Ihrer Distribution verknüpften Domänennamens in den URLs für Ihre Dateien:

```
Set-Cookie: CloudFront-Expires=1426500000; Domain=d111111abcdef8.cloudfront.net; Path=/
images/*; Secure; HttpOnly
Set-Cookie: CloudFront-Signature=yXrSIgyQoeE4FBI4eMKF6ho~CA8_;
 Domain=d111111abcdef8.cloudfront.net; Path=/images/*; Secure; HttpOnly
Set-Cookie: CloudFront-Key-Pair-Id=K2JCJMDEHXQW5F;
 Domain=d111111abcdef8.cloudfront.net; Path=/images/*; Secure; HttpOnly
Set-Cookie: CloudFront-Hash-Algorithm=SHA256; Domain=d111111abcdef8.cloudfront.net;
 Path=/images/*; Secure; HttpOnly
```

Das folgende Beispiel zeigt `Set-Cookie`-Header für ein signiertes Cookie bei Verwendung des alternativen Domänennamens example.org in den URLs für Ihre Dateien:

```
Set-Cookie: CloudFront-Expires=1426500000; Domain=example.org; Path=/images/*; Secure;
 HttpOnly
Set-Cookie: CloudFront-Signature=yXrSIgyQoeE4FBI4eMKF6ho~CA8_; Domain=example.org;
 Path=/images/*; Secure; HttpOnly
Set-Cookie: CloudFront-Key-Pair-Id=K2JCJMDEHXQW5F; Domain=example.org; Path=/images/*;
 Secure; HttpOnly
```

```
Set-Cookie: CloudFront-Hash-Algorithm=SHA256; Domain=example.org; Path=/images/*;
 Secure; HttpOnly
```

Wenn Sie einen alternativen Domänennamen wie beispielsweise example.com in URLs verwenden möchten, müssen Sie den alternativen Domänennamen zu Ihrer Distribution hinzufügen, unabhängig davon, ob Sie das `Domain`-Attribut angeben. Weitere Informationen finden Sie unter Alternative Domänennamen (CNAMEs) im Thema Referenz für alle Distributionseinstellungen.

Erstellen einer Signatur für ein signiertes Cookie, das eine vordefinierte Richtlinie verwendet

Zum Erstellen der Signatur für ein signiertes Cookie, das eine vordefinierte Richtlinie verwendet, führen Sie die folgenden Schritte aus.

Themen

- Erstellen einer Richtlinienanweisung für ein signiertes Cookie, das eine vordefinierte Richtlinie verwendet
- Signieren der Richtlinienanweisung zum Erstellen einer Signatur für ein signiertes Cookie, das eine vordefinierte Richtlinie verwendet

Erstellen einer Richtlinienanweisung für ein signiertes Cookie, das eine vordefinierte Richtlinie verwendet

Wenn Sie ein signiertes Cookie einrichten, das eine vordefinierte Richtlinie verwendet, ist das `CloudFront-Signature`-Attribut eine gehashte und signierte Version einer Richtlinienanweisung. Bei signierten Cookies, die eine vordefinierte Richtlinie verwenden, fügen Sie die Richtlinienanweisung nicht in den `Set-Cookie`-Header ein, wie Sie es bei signierten Cookies tun, die eine benutzerdefinierte Richtlinie verwenden. Führen Sie die folgenden Schritte aus, um die Richtlinienanweisung zu erstellen.

So erstellen Sie eine Richtlinienanweisung für ein signiertes Cookie, das eine vordefinierte Richtlinie verwendet

1. Erstellen Sie die Richtlinienerklärung im folgenden JSON-Format und unter Verwendung der UTF-8 Zeichenkodierung. Fügen Sie alle Satzzeichen und andere Literalwerte genau wie angegeben ein. Informationen zu den Parametern `Resource` und `DateLessThan` finden Sie unter Werte, die Sie in der Richtlinienanweisung für eine vordefinierte Richtlinie für signierte Cookies angeben.

   ```
   {
   ```

```
    "Statement": [
        {
            "Resource": "base URL or stream name",
            "Condition": {
                "DateLessThan": {
                    "AWS:EpochTime": ending date and time in Unix time format and
 UTC
                }
            }
        }
    ]
}
```

2. Entfernen Sie alle Leerzeichen (einschließlich Tabulatoren und Zeilenumbruchzeichen) aus der Richtlinienanweisung. Möglicherweise müssen Sie in der Zeichenfolge im Anwendungscode Escape-Zeichen einfügen.

Werte, die Sie in der Richtlinienanweisung für eine vordefinierte Richtlinie für signierte Cookies angeben

Beim Erstellen einer Richtlinienanweisung für eine vordefinierte Richtlinie geben Sie die folgenden Werte an:

Ressource

Die Basis-URL einschließlich Ihrer Abfragezeichenfolgen, sofern vorhanden, zum Beispiel:

`https://d111111abcdef8.cloudfront.net/images/horizon.jpg?size=large&license=yes`

Sie können nur einen Wert für `Resource` angeben.

Beachten Sie Folgendes:

- Protokoll – Der Wert muss mit `http://` oder `https://` beginnen.
- Abfragezeichenfolgeparameter – Wenn Sie über keine Abfragezeichenfolgeparameter verfügen, lassen Sie das Fragezeichen weg.
- Alternative Domänennamen – Wenn Sie einen alternativen Domänennamen (CNAME) in der URL angeben, müssen Sie diesen alternativen Domänennamen angeben, wenn Sie auf Ihrer Webseite oder in Ihrer Anwendung auf die Datei verweisen. Geben Sie nicht die Amazon S3-URL für Datei an.

DateLessThan

Das Ablaufdatum und die Ablaufzeit für die URL im Unix-Zeitformat (in Sekunden) und in koordinierter Weltzeit (UTC). Setzen Sie den Wert nicht in Anführungszeichen.

Beispielsweise wird der 16. März 2015, 10:00 Uhr UTC in 1426500000 im Unix-Zeitformat umgewandelt.

Dieser Wert muss mit dem Wert des `CloudFront-Expires`-Attributs im `Set-Cookie`-Header übereinstimmen. Setzen Sie den Wert nicht in Anführungszeichen.

Weitere Informationen finden Sie unter Wann CloudFront überprüft das Ablaufdatum und die Uhrzeit in einem signierten Cookie.

Beispiel-Richtlinienanweisung für eine vordefinierte Richtlinie

Wenn Sie die folgende Beispiel-Richtlinienanweisung in einem signierten Cookie verwenden, kann ein Benutzer bis zum 16. März 2015, 10:00 Uhr UTC, auf die Datei `https://d111111abcdef8.cloudfront.net/horizon.jpg` zugreifen:

```
{
    "Statement": [
        {
            "Resource": "https://d111111abcdef8.cloudfront.net/horizon.jpg?
size=large&license=yes",
            "Condition": {
                "DateLessThan": {
                    "AWS:EpochTime": 1426500000
                }
            }
        }
    ]
}
```

Signieren der Richtlinienanweisung zum Erstellen einer Signatur für ein signiertes Cookie, das eine vordefinierte Richtlinie verwendet

Um den Wert für das `CloudFront-Signature`-Attribut in einem `Set-Cookie`-Header zu erstellen, müssen Sie die in So erstellen Sie eine Richtlinienanweisung für ein signiertes Cookie, das eine vordefinierte Richtlinie verwendet erstellte Richtlinienanweisung hashen und signieren.

Weitere Informationen und Beispiele für das Hashing, Signieren und Codieren der Richtlinienanweisung finden Sie in den folgenden Themen:

- Linux-Befehle und OpenSSL für die Base64-Codierung und die Verschlüsselung
- Code-Beispiele für das Erstellen einer Signatur für eine signierte URL

> **Note**
>
> Die verlinkten Beispiele SHA-1 werden standardmäßig verwendet. Um es SHA-256 stattdessen zu verwenden, `sha1` ersetzen Sie es `sha256` in den OpenSSL-Befehlen und fügen Sie das `CloudFront-Hash-Algorithm` Cookie mit dem Wert von `SHA256` ein.

So erstellen Sie eine Signatur für ein signiertes Cookie mit einer vordefinierten Richtlinie

1. Verwenden Sie die SHA-256 Hash-Funktion SHA-1 oder und RSA, um die Richtlinienanweisung, die Sie in der Prozedur erstellt haben, zu hashen und zu signieren. So erstellen Sie eine Richtlinienanweisung für ein signiertes Cookie, das eine vordefinierte Richtlinie verwendet Verwenden Sie die Version der Richtlinienanweisung, die keine Leerzeichen mehr enthält.

   Wenn Sie es verwenden SHA-256, müssen Sie das `CloudFront-Hash-Algorithm` Cookie mit einem Wert von `SHA256` einschließen.

   Verwenden Sie für den privaten Schlüssel, der für die Hash-Funktion erforderlich ist, einen privaten Schlüssel, dessen öffentlicher Schlüssel sich in einer aktiven vertrauenswürdigen Schlüsselgruppe für die Distribution befindet.

   > **Note**
   >
   > Die Methode, die Sie zum Hashen und Signieren der Richtlinienanweisung verwenden, ist abhängig von Ihrer Programmiersprache und Plattform. Einen Beispiel-Code finden Sie unter Code-Beispiele für das Erstellen einer Signatur für eine signierte URL.

2. Entfernen Sie die Leerzeichen (einschließlich Tabulatoren und Zeilenumbruchzeichen) aus der gehashten und signierten Zeichenfolge.
3. Base64-encode die Zeichenfolge, die die MIME-Base64-Codierung verwendet. Weitere Informationen finden Sie in Abschnitt 6.8, Base64 Content-Transfer-Encoding in RFC 2045, MIME (Multipurpose Internet Mail Extensions), Teil 1: Format von Internetnachrichtentexten.

4. Ersetzen Sie Zeichen, die in einer URL-Abfragezeichenfolge nicht gültig sind, durch gültige Zeichen. In der folgenden Tabelle sind ungültige und gültige Zeichen aufgelistet.

Ersetzen Sie diese ungültigen Zeichen	Durch diese gültigen Zeichen
+	- (Bindestrich)
=	_ (Unterstrich)
/	~ (Tilde)

5. Fügen Sie den resultierenden Wert im `Set-Cookie`-Header für das `CloudFront-Signature`-Name-Wert-Paar ein. Kehren Sie anschließend zu So richten Sie ein signiertes Cookies mit einer vordefinierten Richtlinie ein zurück und fügen Sie den `Set-Cookie`-Header für `CloudFront-Key-Pair-Id` hinzu.

Einrichten signierter Cookies mit einer benutzerdefinierten Richtlinie

Führen Sie die folgenden Schritte aus, um ein signiertes Cookie einzurichten, das eine benutzerdefinierte Richtlinie verwendet:

So richten Sie ein signiertes Cookies mit einer benutzerdefinierten Richtlinie ein

1. Wenn Sie .NET oder Java verwenden, um signierte URLs zu erstellen, und den privaten Schlüssel für Ihr Schlüsselpaar noch nicht vom PEM-Standardformat in ein mit .NET oder Java kompatibles Format neu formatiert haben, holen Sie diesen Schritt jetzt nach. Weitere Informationen finden Sie unter Neuformatierung des privaten Schlüssels (nur .NET und Java).
2. Programmieren Sie Ihre Anwendung so, dass sie drei `Set-Cookie` Header an zugelassene Zuschauer sendet (oder vier, wenn Sie einen Hash-Algorithmus angeben möchten). Sie benötigen drei `Set-Cookie` Header, da jeder `Set-Cookie` Header nur ein Name-Wert-Paar enthalten kann und ein CloudFront signiertes Cookie drei Name-Wert-Paare benötigt. Die Name-Wert-Paare sind: `CloudFront-Policy`, `CloudFront-Signature` und `CloudFront-Key-Pair-Id`. Sie können optional ein viertes Name-Wert-Paar hinzufügen`CloudFront-Hash-Algorithm`, um den für die Signatur verwendeten Hash-Algorithmus anzugeben. Die Werte müssen auf dem Viewer vorhanden sein, bevor ein Benutzer die erste Anfrage für eine Datei stellt, bei der der Zugriff kontrolliert werden soll.

> **Note**
>
> Im Allgemeinen empfehlen wir, die Attribute `Expires` und `Max-Age` auszuschließen. Dies bewirkt, dass der Browser das Cookie löscht, wenn der Benutzer den Browser schließt. Dies verringert das Risiko, dass ein Benutzer unbefugten Zugriff auf Ihre Inhalte erhält. Weitere Informationen finden Sie unter Verhindern der missbräuchlichen Verwendung von signierten Cookies.

Bei den Namen der Cookie-Attribute muss die Groß- und Kleinschreibung beachtet werden.

Zeilenumbrüche werden nur hinzugefügt, damit die Attribute besser lesbar sind.

```
Set-Cookie:
CloudFront-Policy=base64 encoded version of the policy statement;
Domain=optional domain name;
Path=/optional directory path;
Secure;
HttpOnly


Set-Cookie:
CloudFront-Signature=hashed and signed version of the policy statement;
Domain=optional domain name;
Path=/optional directory path;
Secure;
HttpOnly

Set-Cookie:
CloudFront-Key-Pair-Id=public key ID for the CloudFront public key whose
 corresponding private key you're using to generate the signature;
Domain=optional domain name;
Path=/optional directory path;
Secure;
HttpOnly

Set-Cookie:
CloudFront-Hash-Algorithm=SHA1 or SHA256;
Domain=optional domain name;
Path=/optional directory path;
Secure;
```

```
HttpOnly
```

(Optional) **`Domain`**

Der Domänenname für die angeforderte Datei. Wenn Sie kein `Domain`-Attribut angeben, ist der Standardwert der Domänenname in der URL und dieser gilt nur für die angegebenen Domänennamen, nicht für Unterdomänen. Wenn Sie ein `Domain`-Attribut angeben, gilt dieses auch für Unterdomänen. Ein vorangestellter Punkt im Domänennamen (z. B. `Domain=.example.com`) ist optional. Wenn Sie ein `Domain`-Attribut angeben, müssen darüber hinaus der Domänenname in der URL und der Wert des `Domain`-Attributs übereinstimmen.

Sie können den Domainnamen angeben, der Ihrer Distribution CloudFront zugewiesen wurde, z. B. d111111abcdef8.cloudfront.net, aber Sie können nicht *.cloudfront.net als Domainnamen angeben.

Wenn Sie einen alternativen Domänennamen wie beispielsweise example.com in URLs verwenden möchten, müssen Sie den alternativen Domänennamen zu Ihrer Distribution hinzufügen, unabhängig davon, ob Sie das `Domain`-Attribut angeben. Weitere Informationen finden Sie unter Alternative Domänennamen (CNAMEs) im Thema Referenz für alle Distributionseinstellungen.

(Optional) **`Path`**

Der Pfad für die angeforderte Datei. Wenn Sie kein `Path`-Attribut angeben, ist der Standardwert der Pfad in der URL.

`Secure`

Macht es erforderlich, dass der Viewer Cookies vor dem Senden einer Anfrage verschlüsselt. Wir empfehlen, den `Set-Cookie` Header über eine HTTPS-Verbindung zu senden, um sicherzustellen, dass die Cookie-Attribute vor Man-in-the-Middle-Angriffen geschützt sind.

`HttpOnly`

Macht es erforderlich, dass der Viewer das Cookie nur in HTTP- oder HTTPS-Anfragen sendet.

`CloudFront-Policy`

Ihre Richtlinienanweisung im JSON-Format, wobei Leerzeichen entfernt wurden und eine Base64-Codierung vorgenommen wurde. Weitere Informationen finden Sie unter Erstellen einer Signatur für ein signiertes Cookie, das eine benutzerdefinierte Richtlinie verwendet.

Die Richtlinienanweisung steuert den Zugriff, den ein signiertes Cookie einem Benutzer gewährt. Sie enthält die Dateien, auf die der Benutzer zugreifen kann, ein Ablaufdatum und eine Ablaufuhrzeit, ein optionales Datum und die Uhrzeit, zu der die URL gültig wird, und eine optionale IP-Adresse oder einen Bereich von IP-Adressen, die auf die Datei zugreifen dürfen.

`CloudFront-Signature`

Eine gehashte, signierte und Base64-codierte Version der JSON-Richtlinienanweisung. Weitere Informationen finden Sie unter Erstellen einer Signatur für ein signiertes Cookie, das eine benutzerdefinierte Richtlinie verwendet.

`CloudFront-Key-Pair-Id`

Die ID für einen CloudFront öffentlichen Schlüssel, zum Beispiel. `K2JCJMDEHXQW5F` Die öffentliche Schlüssel-ID gibt an CloudFront, welcher öffentliche Schlüssel zur Validierung der signierten URL verwendet werden soll. CloudFrontvergleicht die Informationen in der Signatur mit den Informationen in der Grundsatzerklärung, um sicherzustellen, dass die URL nicht manipuliert wurde.

Dieser öffentliche Schlüssel muss zu einer Schlüsselgruppe gehören, die ein vertrauenswürdiger Aussteller in der Distribution ist. Weitere Informationen finden Sie unter Festlegen der Signaturgeber, die signierte URLs und signierte Cookies erstellen können.

`CloudFront-Hash-Algorithm`

(Optional) Der Hash-Algorithmus, der zur Erstellung der Signatur verwendet wurde. Unterstützte Werte sind `SHA1` und `SHA256`. Wenn Sie dieses Cookie nicht hinzufügen, ist es CloudFront standardmäßig auf `SHA1` eingestellt.

`Set-Cookie`-Beispielheader für benutzerdefinierte Richtlinien

Sehen Sie sich die folgenden Beispiele für `Set-Cookie`-Headerpaare an.

Wenn Sie einen alternativen Domainnamen wie beispielsweise example.org in URLs verwenden möchten, müssen Sie den alternativen Domainnamen zu Ihrer Distribution hinzufügen, unabhängig

davon, ob Sie das `Domain`-Attribut angeben. Weitere Informationen finden Sie unter Alternative Domänennamen (CNAMEs) im Thema Referenz für alle Distributionseinstellungen.

Example Beispiel 1

Sie können die `Set-Cookie`-Header für ein signiertes Cookie bei Verwendung des mit Ihrer Distribution verknüpften Domainnamens in den URLs für Ihre Dateien verwenden.

```
Set-Cookie: CloudFront-
Policy=eyJTdGF0ZW1lbnQiOlt7IlJlc291cmNlIjoiaHR0cDovL2QxMTExMTFhYmNkZWY4LmNsb3VkZnJvbnQubmV0L2dh
 Domain=d111111abcdef8.cloudfront.net; Path=/; Secure; HttpOnly
Set-Cookie: CloudFront-Signature=dtKhpJ3aUYxqDIwepczPiDb9NXQ_;
 Domain=d111111abcdef8.cloudfront.net; Path=/; Secure; HttpOnly
Set-Cookie: CloudFront-Key-Pair-Id=K2JCJMDEHXQW5F;
 Domain=d111111abcdef8.cloudfront.net; Path=/; Secure; HttpOnly
Set-Cookie: CloudFront-Hash-Algorithm=SHA256; Domain=d111111abcdef8.cloudfront.net;
 Path=/; Secure; HttpOnly
```

Example Beispiel 2

Sie können die `Set-Cookie`-Header für ein signiertes Cookie bei Verwendung eines alternativen Domainnamens (example.org) in den URLs für Ihre Dateien verwenden.

```
Set-Cookie: CloudFront-
Policy=eyJTdGF0ZW1lbnQiOlt7IlJlc291cmNlIjoiaHR0cDovL2QxMTExMTFhYmNkZWY4LmNsb3VkZnJvbnQubmV0L2dh
 Domain=example.org; Path=/; Secure; HttpOnly
Set-Cookie: CloudFront-Signature=dtKhpJ3aUYxqDIwepczPiDb9NXQ_; Domain=example.org;
 Path=/; Secure; HttpOnly
Set-Cookie: CloudFront-Key-Pair-Id=K2JCJMDEHXQW5F; Domain=example.org; Path=/; Secure;
 HttpOnly
Set-Cookie: CloudFront-Hash-Algorithm=SHA256; Domain=example.org; Path=/; Secure;
 HttpOnly
```

Example Beispiel 3

Sie können die `Set-Cookie`-Headerpaare für eine signierte Anforderung bei Verwendung des mit Ihrer Distribution verknüpften Domainnamens in den URLs für Ihre Dateien verwenden.

```
Set-Cookie: CloudFront-
Policy=eyJTdGF0ZW1lbnQiOlt7IlJlc291cmNlIjoiaHR0cDovL2QxMTExMTFhYmNkZWY4LmNsb3VkZnJvbnQubmV0L2dh
 Domain=d111111abcdef8.cloudfront.net; Path=/; Secure; HttpOnly
```

```
Set-Cookie: CloudFront-Signature=dtKhpJ3aUYxqDIwepczPiDb9NXQ_;
 Domain=d111111abcdef8.cloudfront.net; Path=/; Secure; HttpOnly
Set-Cookie: CloudFront-Key-Pair-Id=K2JCJMDEHXQW5F;
 Domain=dd111111abcdef8.cloudfront.net; Path=/; Secure; HttpOnly
Set-Cookie: CloudFront-Hash-Algorithm=SHA256; Domain=d111111abcdef8.cloudfront.net;
 Path=/; Secure; HttpOnly
```

Example Beispiel 4

Sie können die `Set-Cookie`-Headerpaare für eine signierte Anforderung bei Verwendung eines mit Ihrer Distribution verknüpften alternativen Domainnamens (example.org) in den URLs für Ihre Dateien verwenden.

```
Set-Cookie: CloudFront-
Policy=eyJTdGF0ZW1lbnQiOlt7IlJlc291cmNlIjoiaHR0cDovL2QxMTExMTFhYmNkZWY4LmNsb3VkZnJvbnQubmV0L2dh
 Domain=example.org; Path=/; Secure; HttpOnly
Set-Cookie: CloudFront-Signature=dtKhpJ3aUYxqDIwepczPiDb9NXQ_; Domain=example.org;
 Path=/; Secure; HttpOnly
Set-Cookie: CloudFront-Key-Pair-Id=K2JCJMDEHXQW5F; Domain=example.org; Path=/; Secure;
 HttpOnly
Set-Cookie: CloudFront-Hash-Algorithm=SHA256; Domain=example.org; Path=/; Secure;
 HttpOnly
```

Erstellen einer Richtlinienanweisung für ein signiertes Cookie, das eine benutzerdefinierte Richtlinie verwendet

Zum Erstellen einer Richtlinienanweisung für eine benutzerdefinierte Richtlinie führen Sie die folgenden Schritte aus. Einige Beispiel-Richtlinienanweisungen, die den Zugriff auf Dateien auf verschiedene Weisen kontrollieren, finden Sie unter Beispiel-Richtlinienanweisungen für ein signiertes Cookie, das eine benutzerdefinierte Richtlinie verwendet.

So erstellen Sie die Richtlinienanweisung für ein signiertes Cookie, das eine benutzerdefinierte Richtlinie verwendet

1. Erstellen Sie die Richtlinienanweisung unter Verwendung des folgenden JSON-Formats.

```
{
    "Statement": [
        {
            "Resource": "URL of the file",
            "Condition": {
                "DateLessThan": {
```

```
                    "AWS:EpochTime":required ending date and time in Unix time
 format and UTC
                },
                "DateGreaterThan": {
                    "AWS:EpochTime":optional beginning date and time in Unix time
 format and UTC
                },
                "IpAddress": {
                    "AWS:SourceIp": "optional IP address"
                }
            }
        }
    ]
}
```

Beachten Sie Folgendes:

- Sie können nur eine Anweisung einschließen.
- Verwenden Sie die UTF-8 Zeichenkodierung.
- Fügen Sie alle Satzzeichen und Parameternamen genau wie angegeben ein. Abkürzungen für Parameternamen werden nicht akzeptiert.
- Die Reihenfolge der Parameter im Bereich `Condition` ist unerheblich.
- Informationen zu den Werten für `Resource`, `DateLessThan`, `DateGreaterThan` und `IpAddress` finden Sie unter Werte, die Sie in der Richtlinienanweisung für eine benutzerdefinierte Richtlinie für signierte Cookies angeben.

2. Entfernen Sie alle Leerzeichen (einschließlich Tabulatoren und Zeilenumbruchzeichen) aus der Richtlinienanweisung. Möglicherweise müssen Sie in der Zeichenfolge im Anwendungscode Escape-Zeichen einfügen.
3. Base64-encode die Richtlinienanweisung, die die MIME-Base64-Kodierung verwendet. Weitere Informationen finden Sie in Abschnitt 6.8, Base64 Content-Transfer-Encoding in RFC 2045, MIME (Multipurpose Internet Mail Extensions), Teil 1: Format von Internetnachrichtentexten.
4. Ersetzen Sie Zeichen, die in einer URL-Abfragezeichenfolge nicht gültig sind, durch gültige Zeichen. In der folgenden Tabelle sind ungültige und gültige Zeichen aufgelistet.

Ersetzen Sie diese ungültigen Zeichen	Durch diese gültigen Zeichen
+	- (Bindestrich)

Ersetzen Sie diese ungültigen Zeichen	Durch diese gültigen Zeichen
=	_ (Unterstrich)
/	~ (Tilde)

5. Fügen Sie den resultierenden Wert im `Set-Cookie`-Header hinter `CloudFront-Policy=` ein.
6. Erstellen Sie eine Signatur für den `Set-Cookie`-Header für `CloudFront-Signature`, indem Sie die Richtlinienanweisung hashen, signieren und eine Base64-Codierung vornehmen. Weitere Informationen finden Sie unter Erstellen einer Signatur für ein signiertes Cookie, das eine benutzerdefinierte Richtlinie verwendet.

Werte, die Sie in der Richtlinienanweisung für eine benutzerdefinierte Richtlinie für signierte Cookies angeben

Beim Erstellen einer Richtlinienanweisung für eine benutzerdefinierte Richtlinie geben Sie die folgenden Werte an.

Ressource

Die Basis-URL einschließlich Ihrer Abfragezeichenfolgen, sofern vorhanden:

```
https://d111111abcdef8.cloudfront.net/images/horizon.jpg?
size=large&license=yes
```

> **Important**
>
> Wenn Sie den `Resource`-Parameter weglassen, können Benutzer auf alle Dateien zugreifen, die mit einer Distribution verknüpft sind, die mit dem zum Erstellen der signierten URL verwendeten Schlüsselpaar verknüpft ist.

Sie können nur einen Wert für `Resource` angeben.

Beachten Sie Folgendes:

- Protokoll – Der Wert muss mit `http://` oder `https://` beginnen.

- Abfragezeichenfolgeparameter – Wenn Sie über keine Abfragezeichenfolgeparameter verfügen, lassen Sie das Fragezeichen weg.
- Platzhalter – Sie können das Platzhalterzeichen, das null oder mehr Zeichen (*), oder das Platzhalterzeichen, das genau einem Zeichen (?) irgendwo in der Zeichenfolge entspricht, verwenden. Beispielsweise würde der Wert:

  `https://d111111abcdef8.cloudfront.net/*game_download.zip*`

  u. a. die folgenden Dateien umfassen:

  - `https://d111111abcdef8.cloudfront.net/game_download.zip`
  - `https://d111111abcdef8.cloudfront.net/example_game_download.zip?license=yes`
  - `https://d111111abcdef8.cloudfront.net/test_game_download.zip?license=temp`

  Ein in Prozent kodiertes Zeichen in der Anforderungs-URL (z. B. `%3B` für`;`, `%3F` für oder `%23` für#) wird als Daten behandelt?, nicht als URL-Trennzeichen. Beispielsweise kann ein Platzhalter nicht in einem literalen Semikolon (`;`) übereinstimmen, aber er kann in der codierten Form () übereinstimmen. `%3B` Weitere Informationen finden Sie unter Werte, die Sie in der Richtlinienanweisung für eine signierte URL angeben, die eine benutzerdefinierte Richtlinie verwendet.
- Alternative Domänennamen – Wenn Sie einen alternativen Domänennamen (CNAME) in der URL angeben, müssen Sie diesen alternativen Domänennamen angeben, wenn Sie auf Ihrer Webseite oder in Ihrer Anwendung auf die Datei verweisen. Geben Sie nicht die Amazon S3-URL für Datei an.

DateLessThan

Das Ablaufdatum und die Ablaufzeit für die URL im Unix-Zeitformat (in Sekunden) und in koordinierter Weltzeit (UTC). Setzen Sie den Wert nicht in Anführungszeichen.

Beispielsweise wird der 16. März 2015, 10:00 Uhr UTC in 1426500000 im Unix-Zeitformat umgewandelt.

Weitere Informationen finden Sie unter Wann CloudFront überprüft das Ablaufdatum und die Uhrzeit in einem signierten Cookie.

DateGreaterThan (Fakultativ)

Ein optionales Datum und eine optionale Zeit für die URL im Unix-Zeitformat (in Sekunden) und in koordinierter Weltzeit (UTC). Benutzer sind an oder vor dem angegebenen Datum und der angegebenen Zeit nicht berechtigt, auf die Datei zuzugreifen. Setzen Sie den Wert nicht in Anführungszeichen.

IpAddress (Fakultativ)

Die IP-Adresse des Clients, der die GET-Anfrage stellt. Beachten Sie Folgendes:

- Um allen IP-Adressen den Zugriff auf die Datei zu gewähren, lassen Sie den Parameter `IpAddress` weg.
- Sie können entweder eine IP-Adresse oder einen IP-Adressbereich angeben. Sie können die Richtlinie beispielsweise nicht so einrichten, dass Zugriff gewährt wird, wenn die IP-Adresse des Clients sich in einem von zwei getrennten Bereichen befindet.
- Um den Zugriff von einer einzigen IP-Adresse zu gewähren, geben Sie Folgendes an:

  `"`*`IPv4 IP address`*`/32"`
- Sie müssen IP-Adressbereiche im IPv4-CIDR-Standardformat angeben (z. B, `192.0.2.0/24`). Weitere Informationen finden Sie unter RFC 4632, Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan,. https://tools.ietf.org/html/rfc4632

> **Important**
>
> IP-Adressen im IPv6-Format, wie beispielsweise 2001:0db8:85a3::8a2e:0370:7334, werden nicht unterstützt.

Wenn Sie eine benutzerdefinierte Richtlinie verwenden, die `IpAddress` enthält, aktivieren Sie IPv6 nicht für die Distribution. Wenn Sie den Zugriff auf bestimmte Inhalte nach IP-Adresse einschränken und IPv6-Anforderungen für andere Inhalte unterstützen möchten, können Sie zwei Distributionen erstellen. Weitere Informationen finden Sie unter Aktivieren von IPv6 (Viewer-Anforderungen) im Thema Referenz für alle Distributionseinstellungen.

Beispiel-Richtlinienanweisungen für ein signiertes Cookie, das eine benutzerdefinierte Richtlinie verwendet

Die folgenden Beispiel-Richtlinienanweisungen zeigen, wie der Zugriff auf eine bestimmte Datei, auf alle Dateien in einem Verzeichnis oder auf alle mit einer Schlüsselpaar-ID verknüpften Dateien kontrolliert wird. Die Beispiele zeigen auch, wie der Zugriff von einer einzelnen IP-Adresse oder einem Bereich von IP-Adressen kontrolliert wird und wie Sie verhindern, dass Benutzer das signierte Cookie nach einem festgelegten Datum und einer festgelegten Zeit verwenden.

Wenn Sie eines dieser Beispiele kopieren und einfügen, entfernen Sie alle Leerzeichen (einschließlich Tabuloren und Zeilenumbruchzeichen), ersetzen Sie die entsprechenden Werte durch Ihre eigenen Werte und fügen Sie ein Zeilenumbruchzeichen hinter der schließenden Klammer (}) ein.

Weitere Informationen finden Sie unter Werte, die Sie in der Richtlinienanweisung für eine benutzerdefinierte Richtlinie für signierte Cookies angeben.

Themen

- Beispiel-Richtlinienanweisung: Zugriff auf eine Datei von einem Bereich von IP-Adressen aus
- Beispiel-Richtlinienanweisung: Zugriff auf alle Dateien in einem Verzeichnis über einen Bereich von IP-Adressen
- Beispiel-Richtlinienanweisung: Zugriff auf alle mit einer Schlüsselpaar-ID verknüpften Dateien über eine IP-Adresse

Beispiel-Richtlinienanweisung: Zugriff auf eine Datei von einem Bereich von IP-Adressen aus

Die folgende Beispiel-Richtlinienanweisung in einem signierten Cookie legt fest, dass ein Benutzer bis zum 1. Januar 2023, 10:00 Uhr UTC, von IP-Adressen im Bereich `https://d111111abcdef8.cloudfront.net/game_download.zip` auf die Datei `192.0.2.0/24` zugreifen kann:

```
{
    "Statement": [
        {
            "Resource": "https://d111111abcdef8.cloudfront.net/game_download.zip",
            "Condition": {
                "IpAddress": {
                    "AWS:SourceIp": "192.0.2.0/24"
                },
```

```
                "DateLessThan": {
                    "AWS:EpochTime": 1767290400
                }
            }
        }
    ]
}
```

Beispiel-Richtlinienanweisung: Zugriff auf alle Dateien in einem Verzeichnis über einen Bereich von IP-Adressen

Mit dem folgenden Beispiel für eine benutzerdefinierte Richtlinie können Sie signierte Cookies für jede Datei im `training` Verzeichnis erstellen, was durch das Platzhalterzeichen * im Parameter gekennzeichnet ist. `Resource` Benutzer können bis zum 1. Januar 2013, 10:00 Uhr UTC, von IP-Adressen im Bereich `192.0.2.0/24` auf die Datei zugreifen:

```
{
    "Statement": [
        {
            "Resource": "https://d111111abcdef8.cloudfront.net/training/*",
            "Condition": {
                "IpAddress": {
                    "AWS:SourceIp": "192.0.2.0/24"
                },
                "DateLessThan": {
                    "AWS:EpochTime": 1767290400
                }
            }
        }
    ]
}
```

Jedes signierte Cookie, in dem Sie diese Richtlinie verwenden, enthält eine Basis-URL, die eine bestimmte Datei kennzeichnet, zum Beispiel:

`https://d111111abcdef8.cloudfront.net/training/orientation.pdf`

Beispiel-Richtlinienanweisung: Zugriff auf alle mit einer Schlüsselpaar-ID verknüpften Dateien über eine IP-Adresse

Mit dem folgenden Beispiel für eine benutzerdefinierte Richtlinie können Sie signierte Cookies für jede Datei festlegen, die einer beliebigen Distribution zugeordnet ist. Dies wird durch das

Platzhalterzeichen * im `Resource` Parameter angezeigt. Der Benutzer muss die IP-Adresse verwende `192.0.2.10/32`. (Der Wert `192.0.2.10/32` in CIDR-Notation bezieht sich auf eine einzelne IP-Adresse, `192.0.2.10`.) Die Dateien sind nur vom 1. Januar 2013, 10:00 Uhr UTC, bis zum 2. Januar 2013, 10:00 Uhr UTC, verfügbar:

```
{
    "Statement": [
        {
            "Resource": "https://*",
            "Condition": {
                "IpAddress": {
                    "AWS:SourceIp": "192.0.2.10/32"
                },
                "DateGreaterThan": {
                    "AWS:EpochTime": 1767290400
                },
                "DateLessThan": {
                    "AWS:EpochTime": 1767376800
                }
            }
        }
    ]
}
```

Jedes signierte Cookie, in dem Sie diese Richtlinie verwenden, enthält eine Basis-URL, die eine bestimmte Datei in einer bestimmten CloudFront Distribution identifiziert, zum Beispiel:

`https://d111111abcdef8.cloudfront.net/training/orientation.pdf`

Das signierte Cookie enthält auch eine Schlüsselpaar-ID, die mit einem vertrauenswürdigen Aussteller in der Distribution (d111111abcdef8.cloudfront.net) verknüpft werden muss, die Sie in der Basis-URL angeben.

Erstellen einer Signatur für ein signiertes Cookie, das eine benutzerdefinierte Richtlinie verwendet

Bei der Signatur für ein signiertes Cookie, das eine benutzerdefinierte Richtlinie verwendet, handelt es sich um eine gehashte, signierte und Base64-codierte Version der Richtlinienanweisung.

Weitere Informationen und Beispiele für das Hashing, Signieren und Codieren der Richtlinienanweisung finden Sie unter:

- Linux-Befehle und OpenSSL für die Base64-Codierung und die Verschlüsselung

- Code-Beispiele für das Erstellen einer Signatur für eine signierte URL

> **Note**
>
> Die verlinkten Beispiele SHA-1 werden standardmäßig verwendet. Um es SHA-256 stattdessen zu verwenden, `sha1` ersetzen Sie es `sha256` in den OpenSSL-Befehlen und fügen Sie das `CloudFront-Hash-Algorithm` Cookie mit dem Wert von `SHA256` ein.

So erstellen Sie eine Signatur für ein signiertes Cookie mithilfe einer benutzerdefinierten Richtlinie

1. Verwenden Sie die SHA-256 Hash-Funktion SHA-1 oder und RSA, um die JSON-Richtlinienanweisung, die Sie in der Prozedur erstellt haben, zu hashen und zu signieren. So erstellen Sie die Richtlinienanweisung für eine signierte URL, die eine benutzerdefinierte Richtlinie verwendet Verwenden Sie die Version der Richtlinienanweisung, die keine Leerzeichen mehr enthält, jedoch noch nicht Base64-codiert wurde.

   Wenn Sie es verwenden SHA-256, müssen Sie das `CloudFront-Hash-Algorithm` Cookie mit einem Wert von `SHA256` angeben.

   Verwenden Sie für den privaten Schlüssel, der für die Hash-Funktion erforderlich ist, einen privaten Schlüssel, dessen öffentlicher Schlüssel sich in einer aktiven vertrauenswürdigen Schlüsselgruppe für die Distribution befindet.

   > **Note**
   >
   > Die Methode, die Sie zum Hashen und Signieren der Richtlinienanweisung verwenden, ist abhängig von Ihrer Programmiersprache und Plattform. Einen Beispiel-Code finden Sie unter Code-Beispiele für das Erstellen einer Signatur für eine signierte URL.

2. Entfernen Sie die Leerzeichen (einschließlich Tabulatoren und Zeilenumbruchzeichen) aus der gehashten und signierten Zeichenfolge.
3. Base64-encode die Zeichenfolge, die die MIME-Base64-Codierung verwendet. Weitere Informationen finden Sie in Abschnitt 6.8, Base64 Content-Transfer-Encoding in RFC 2045, MIME (Multipurpose Internet Mail Extensions), Teil 1: Format von Internetnachrichtentexten.
4. Ersetzen Sie Zeichen, die in einer URL-Abfragezeichenfolge nicht gültig sind, durch gültige Zeichen. In der folgenden Tabelle sind ungültige und gültige Zeichen aufgelistet.

Ersetzen Sie diese ungültigen Zeichen	Durch diese gültigen Zeichen
+	- (Bindestrich)
=	_ (Unterstrich)
/	~ (Tilde)

5. Fügen Sie den resultierenden Wert im `Set-Cookie`-Header für das `CloudFront-Signature`=-Name-Wert-Paar ein und kehren Sie zu So richten Sie ein signiertes Cookies mit einer benutzerdefinierten Richtlinie ein zurück, um den `Set-Cookie`-Header für `CloudFront-Key-Pair-Id` hinzuzufügen.

Erstellen signierter Cookies mit PHP

Das folgende Codebeispiel ähnelt dem Beispiel aus Erstellen einer URL-Signatur mit PHP insofern, als es einen Link zu einem Video erstellt. Anstatt jedoch die URL im Code zu signieren, signiert dieses Beispiel die Cookies mit der Funktion `create_signed_cookies()`. Der clientseitige Player verwendet die Cookies, um jede Anfrage an die Distribution zu authentifizieren. CloudFront

Dieser Ansatz ist nützlich, um Inhalte wie HTTP Live Streaming (HLS) oder Dynamic Adaptive Streaming over HTTP (DASH) zu streamen, bei denen der Client mehrere Anforderungen stellen muss, um das Manifest, die Segmente und die zugehörigen Wiedergabe-Assets abzurufen. Mithilfe signierter Cookies kann der Client jede Anforderung authentifizieren, ohne für jedes Segment eine neue signierte URL generieren zu müssen.

> **Note**
>
> - Das Erstellen einer URL-Signatur ist nur ein Teil der Bereitstellung privater Inhalte über signierte Cookies. Weitere Informationen finden Sie unter Verwenden signierter Cookies.

Themen

- Erstellen Sie den RSA oder die Signatur SHA-1 SHA-256

- Erstellen der signierten Cookies
- Vollständiger Code

In den folgenden Abschnitten wird das Codebeispiel in einzelne Teile unterteilt. Das vollständige Codebeispiel finden Sie unten.

Erstellen Sie den RSA oder die Signatur SHA-1 SHA-256

Dieses Codebeispiel führt Folgendes aus:

1. Die Funktion `rsa_sha1_sign` hasht und signiert die Grundsatzerklärung mithilfe von. SHA-1 Verwenden Sie SHA-256 stattdessen die unten abgebildete Funktion rsa_sha256_sign. Die erforderlichen Argumente sind eine Richtlinienanweisung und der private Schlüssel, der einem öffentlichen Schlüssel entspricht, der sich in einer vertrauenswürdigen Schlüsselgruppe für Ihre Distribution befindet.
2. Als Nächstes erstellt die `url_safe_base64_encode` Funktion eine Version der Signatur. URL-safe

```
function rsa_sha1_sign($policy, $private_key_filename) {
    $signature = "";
    $fp = fopen($private_key_filename, "r");
    $priv_key = fread($fp, 8192);
    fclose($fp);
    $pkeyid = openssl_get_privatekey($priv_key);
    openssl_sign($policy, $signature, $pkeyid);
    openssl_free_key($pkeyid);
    return $signature;
}

function url_safe_base64_encode($value) {
    $encoded = base64_encode($value);
    return str_replace(
        array('+', '=', '/'),
        array('-', '_', '~'),
        $encoded);
}
```

Die folgende Funktion verwendet SHA-256 anstelle von SHA-1:

```
function rsa_sha256_sign($policy, $private_key_filename) {
```

```
    $signature = "";
    $fp = fopen($private_key_filename, "r");
    $priv_key = fread($fp, 8192);
    fclose($fp);
    $pkeyid = openssl_get_privatekey($priv_key);
    openssl_sign($policy, $signature, $pkeyid, OPENSSL_ALGO_SHA256);
    openssl_free_key($pkeyid);
    return $signature;
}
```

Die `rsa_sha256_sign` Funktion ist dieselbe wie`rsa_sha1_sign`, außer dass sie `OPENSSL_ALGO_SHA256` an übergeben wird`openssl_sign`. Wenn Sie es verwenden SHA-256, schließen Sie das `CloudFront-Hash-Algorithm` Cookie mit einem Wert von ein`SHA256`.

Erstellen der signierten Cookies

Der folgende Code konstruiert und erstellt die signierten Cookies unter Verwendung der folgenden Cookie-Attribute: `CloudFront-Expires``CloudFront-Signature`, `CloudFront-Key-Pair-Id` und`CloudFront-Hash-Algorithm`. Der Code verwendet eine benutzerdefinierte Richtlinie.

```
function create_signed_cookies($resource, $private_key_filename, $key_pair_id,
 $expires, $client_ip = null, $hash_algorithm = 'SHA1') {
   $policy = array(
       'Statement' => array(
           array(
               'Resource' => $resource,
               'Condition' => array(
                   'DateLessThan' => array('AWS:EpochTime' => $expires)
               )
           )
       )
   );

   if ($client_ip) {
       $policy['Statement'][0]['Condition']['IpAddress'] = array('AWS:SourceIp' =>
 $client_ip . '/32');
   }

   $policy = json_encode($policy);
   $encoded_policy = url_safe_base64_encode($policy);
   if ($hash_algorithm === 'SHA256') {
       $signature = rsa_sha256_sign($policy, $private_key_filename);
```

```
    } else {
        $signature = rsa_sha1_sign($policy, $private_key_filename);
    }
    $encoded_signature = url_safe_base64_encode($signature);

    $cookies = array(
        'CloudFront-Policy' => $encoded_policy,
        'CloudFront-Signature' => $encoded_signature,
        'CloudFront-Key-Pair-Id' => $key_pair_id
    );

    if ($hash_algorithm === 'SHA256') {
        $cookies['CloudFront-Hash-Algorithm'] = 'SHA256';
    }

    return $cookies;
}
```

Weitere Informationen finden Sie unter Einrichten signierter Cookies mit einer benutzerdefinierten Richtlinie.

Vollständiger Code

Der folgende Beispielcode bietet eine vollständige Demonstration der Erstellung CloudFront signierter Cookies mit PHP. Sie können das vollständige Beispiel aus der Datei demo-php.zip herunterladen.

Im folgenden Beispiel können Sie das Element `$policy Condition` so anpassen, dass sowohl IPv4- und IPv6-Adressbereiche erlaubt sind. Ein Beispiel finden Sie unter Verwenden von IPv6-Adressen in IAM-Richtlinien im Benutzerhandbuch für Amazon Simple Storage Service.

```
<?php

function rsa_sha1_sign($policy, $private_key_filename) {
    $signature = "";
    $fp = fopen($private_key_filename, "r");
    $priv_key = fread($fp, 8192);
    fclose($fp);
    $pkeyid = openssl_get_privatekey($priv_key);
    openssl_sign($policy, $signature, $pkeyid);
    openssl_free_key($pkeyid);
    return $signature;
}
```

```
function url_safe_base64_encode($value) {
    $encoded = base64_encode($value);
    return str_replace(
        array('+', '=', '/'),
        array('-', '_', '~'),
        $encoded);
}

function rsa_sha256_sign($policy, $private_key_filename) {
    $signature = "";
    $fp = fopen($private_key_filename, "r");
    $priv_key = fread($fp, 8192);
    fclose($fp);
    $pkeyid = openssl_get_privatekey($priv_key);
    openssl_sign($policy, $signature, $pkeyid, OPENSSL_ALGO_SHA256);
    openssl_free_key($pkeyid);
    return $signature;
}

function create_signed_cookies($resource, $private_key_filename, $key_pair_id,
 $expires, $client_ip = null, $hash_algorithm = 'SHA1') {
    $policy = array(
        'Statement' => array(
            array(
                'Resource' => $resource,
                'Condition' => array(
                    'DateLessThan' => array('AWS:EpochTime' => $expires)
                )
            )
        )
    );

    if ($client_ip) {
        $policy['Statement'][0]['Condition']['IpAddress'] = array('AWS:SourceIp' =>
 $client_ip . '/32');
    }

    $policy = json_encode($policy);
    $encoded_policy = url_safe_base64_encode($policy);
    if ($hash_algorithm === 'SHA256') {
        $signature = rsa_sha256_sign($policy, $private_key_filename);
    } else {
        $signature = rsa_sha1_sign($policy, $private_key_filename);
    }
```

```
    $encoded_signature = url_safe_base64_encode($signature);

    $cookies = array(
        'CloudFront-Policy' => $encoded_policy,
        'CloudFront-Signature' => $encoded_signature,
        'CloudFront-Key-Pair-Id' => $key_pair_id
    );

    if ($hash_algorithm === 'SHA256') {
        $cookies['CloudFront-Hash-Algorithm'] = 'SHA256';
    }

    return $cookies;
}



$private_key_filename = '/home/test/secure/example-priv-key.pem';
$key_pair_id = 'K2JCJMDEHXQW5F';
$base_url = 'https://d1234.cloudfront.net';

$expires = time() + 3600; // 1 hour from now

// Get the viewer real IP from the x-forward-for header as $_SERVER['REMOTE_ADDR']
 will return viewer facing IP. An alternative option is to use CloudFront-Viewer-
Address header. Note that this header is a trusted CloudFront immutable header. Example
 format: IP:PORT ("CloudFront-Viewer-Address": "1.2.3.4:12345")
$client_ip = $_SERVER['HTTP_X_FORWARDED_FOR'];


// For HLS manifest and segments (using wildcard)
$hls_resource = $base_url . '/sign/*';
$signed_cookies = create_signed_cookies($hls_resource, $private_key_filename,
 $key_pair_id, $expires, $client_ip, 'SHA256');

// Set the cookies
$cookie_domain = parse_url($base_url, PHP_URL_HOST);
foreach ($signed_cookies as $name => $value) {
    setcookie($name, $value, $expires, '/', $cookie_domain, true, true);
}

?>

<!DOCTYPE html>
```

```
<html>
<head>
    <title>CloudFront Signed HLS Stream with Cookies</title>
</head>
<body>
    <h1>Amazon CloudFront Signed HLS Stream with Cookies</h1>
    <h2>Expires at <?php echo gmdate('Y-m-d H:i:s T', $expires); ?> only viewable by IP
 <?php echo $client_ip; ?></h2>

    <div id='hls-video'>
        <video id="video" width="640" height="360" controls></video>
    </div>

    <script src="https://cdn.jsdelivr.net/npm/hls.js@latest"></script>
    <script>
        var video = document.getElementById('video');
        var manifestUrl = '<?php echo $base_url; ?>/sign/manifest.m3u8';

        if (Hls.isSupported()) {
            var hls = new Hls();
            hls.loadSource(manifestUrl);
            hls.attachMedia(video);
        }
        else if (video.canPlayType('application/vnd.apple.mpegurl')) {
            video.src = manifestUrl;
        }
    </script>
</body>
</html>
```

Anstelle von signierten Cookies können Sie signierte URLs verwenden. Weitere Informationen finden Sie unter Erstellen einer URL-Signatur mit PHP.

Linux-Befehle und OpenSSL für die Base64-Codierung und die Verschlüsselung

Sie können den folgenden Linux-Befehlszeilenbefehl und OpenSSL verwenden, um die Richtlinienanweisung zu hashen und zu signieren, die Signatur in Base64 zu codieren und Zeichen, die in URL-Abfragezeichenfolgeparametern nicht gültig sind, durch gültige zu ersetzen.

Informationen zu OpenSSL finden Sie unter https://www.openssl.org.

SHA-1 (Standard):

```
cat policy | tr -d "\n" | tr -d " \t\n\r" | openssl sha1 -sign private_key.pem |
 openssl base64 -A | tr -- '+=/' '-_~'
```

SHA-256:

```
cat policy | tr -d "\n" | tr -d " \t\n\r" | openssl sha256 -sign private_key.pem |
 openssl base64 -A | tr -- '+=/' '-_~'
```

Beim vorhergehenden Befehl:

- `cat`liest die `policy` Datei.
- `tr -d "\n" | tr -d " \t\n\r"` entfernt die Leerzeichen und Zeilenumbruchzeichen, die von `cat` hinzugefügt wurden.
- OpenSSL hasht die Datei mit SHA-1 (oder SHA-256) und signiert sie mit der privaten Schlüsseldatei. `private_key.pem` Die Signatur des privaten Schlüssels kann entweder RSA 2048 oder ECDSA 256 sein. Wenn Sie verwenden SHA-256, fügen Sie den `Hash-Algorithm=SHA256` Abfrageparameter in die signierte URL oder das `CloudFront-Hash-Algorithm=SHA256` Cookie für signierte Cookies ein.
- OpenSSL codiert die gehashte und signierte Richtlinienanweisung in Base64.
- `tr`ersetzt Zeichen, die in URL-Abfragezeichenfolgenparametern nicht gültig sind, durch gültige Zeichen.

Weitere Code-Beispiele, welche die Erstellung einer Signatur veranschaulichen, finden Sie unter Code-Beispiele für das Erstellen einer Signatur für eine signierte URL.

Code-Beispiele für das Erstellen einer Signatur für eine signierte URL

Dieser Abschnitt enthält herunterladbare Anwendungsbeispiele, mit denen vorgeführt wird, wie Signaturen für signierte URLs erstellt werden können. Beispiele sind in Perl, PHP, C# und Java verfügbar. Sie können alle Beispiele zum Erstellen signierter URLs verwenden. Das Perl-Skript wird auf Linux-/macOS-Plattformen ausgeführt. Das PHP-Beispiel funktioniert auf allen Servern, auf denen PHP ausgeführt wird. Das C#-Beispiel verwendet das .NET Framework.

Die Beispiele in diesem Abschnitt werden verwendet SHA-1 , um die Grundsatzerklärung zu hashen und zu signieren. Sie können auch verwenden SHA-256. Um es zu verwenden SHA-256, aktualisieren Sie den Hash-Algorithmus in der Signaturfunktion (`sha1`ersetzen Sie ihn beispielsweise

`sha256` in OpenSSL-Aufrufen durch oder verwenden Sie die entsprechende SHA-256 Konstante in der kryptografischen Bibliothek Ihrer Sprache). Wenn Sie verwenden SHA-256, fügen Sie den `Hash-Algorithm=SHA256` Abfrageparameter in die signierte URL ein.

Ein Beispiel für den Code in JavaScript (Node.js) finden Sie im Node.js AWS Entwickler-Blog unter Mit Amazon CloudFront signierte URLs erstellen.

Beispielcode in Python finden Sie unter Generieren einer signierten URL für Amazon CloudFront in der AWS SDK for Python (Boto3) API-Referenz und in diesem Beispielcode im GitHub Boto3-Repository.

Themen

- Erstellen einer URL-Signatur mit Perl
- Erstellen einer URL-Signatur mit PHP
- Erstellen einer URL-Signatur mithilfe von C# und dem .NET Framework
- Erstellen einer URL-Signatur mit Java

Erstellen einer URL-Signatur mit Perl

Dieser Abschnitt enthält ein Perl-Skript für Linux/Mac Plattformen, mit dem Sie die Signatur für private Inhalte erstellen können. Um die Signatur zu erstellen, führen Sie das Skript mit Befehlszeilenargumenten aus, die die CloudFront URL, den Pfad zum privaten Schlüssel des Unterzeichners, die Schlüssel-ID und ein Ablaufdatum für die URL angeben. Mit diesem Tool lassen sich darüber hinaus signierte URLs dekodieren.

Hinweise

- Das Erstellen einer URL-Signatur ist nur ein Teil der Bereitstellung privater Inhalte über eine signierte URL. Weitere Informationen über den End-to-End-Prozess finden Sie unter Verwenden signierter URLs.
- Beachten Sie beim Signierbefehl, dass dies durch `sha256` im `openssl dgst` Aufruf ersetzt werden `sha1` kann.

Themen

- Quelle für das Perl-Skript zum Erstellen einer signierten URL

Quelle für das Perl-Skript zum Erstellen einer signierten URL

Der folgende Perl-Quellcode kann verwendet werden, um eine signierte URL für CloudFront zu erstellen. Kommentare im Code beinhalten Informationen zu den Befehlszeilen-Switches und den Features des Tools.

```
#!/usr/bin/perl -w

# Copyright 2008 Amazon Technologies, Inc.  Licensed under the Apache License, Version
 2.0 (the "License");
# you may not use this file except in compliance with the License. You may obtain a
 copy of the License at:
#
# https://aws.amazon.com/apache2.0
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
 KIND, either express or implied.
# See the License for the specific language governing permissions and limitations under
 the License.

=head1 cfsign.pl

cfsign.pl - A tool to generate and verify Amazon CloudFront signed URLs

=head1 SYNOPSIS

This script uses an existing RSA key pair to sign and verify Amazon CloudFront signed
 URLs

View the script source for details as to which CPAN packages are required beforehand.

For help, try:

cfsign.pl --help

URL signing examples:

cfsign.pl --action encode --url https://images.my-website.com/gallery1.zip --policy
 sample_policy.json --private-key privkey.pem --key-pair-id mykey

cfsign.pl --action encode --url https://images.my-website.com/gallery1.zip --expires
 1257439868 --private-key privkey.pem --key-pair-id mykey
```

```
URL decode example:

cfsign.pl --action decode --url "http//mydist.cloudfront.net/?Signature=AGO-
PgxkYo99MkJFHvjfGXjG1QDEXeaDb4Qtzmy85wqyJjK7eKojQWa4BCRcow__&Policy=eyJTdGF0ZW1lbnQiOlt7IlJlc29
Pair-Id=mykey"


To generate an RSA key pair, you can use openssl and the following commands:

# Generate a 2048 bit key pair
openssl genrsa -out private-key.pem 2048
openssl rsa -in private-key.pem -pubout -out public-key.pem


=head1 OPTIONS

=over 8

=item B<--help>

Print a help message and exits.

=item B<--action> [action]

The action to execute.  action can be one of:

  encode - Generate a signed URL (using a canned policy or a user policy)
  decode - Decode a signed URL

=item B<--url>

The URL to en/decode

=item B<--stream>

The stream to en/decode

=item B<--private-key>

The path to your private key.

=item B<--key-pair-id>

The key pair identifier.
```

```
=item B<--policy>

The CloudFront policy document.

=item B<--expires>

The Unix epoch time when the URL is to expire. If both this option and
the --policy option are specified, --policy will be used. Otherwise, this
option alone will use a canned policy.

=back

=cut

use strict;
use warnings;

# you might need to use CPAN to get these modules.
# run perl -MCPAN -e "install <module>" to get them.
# The openssl command line will also need to be in your $PATH.
use File::Temp qw/tempfile/;
use File::Slurp;
use Getopt::Long;
use IPC::Open2;
use MIME::Base64 qw(encode_base64 decode_base64);
use Pod::Usage;
use URI;

my $CANNED_POLICY
    = '{"Statement":[{"Resource":"<RESOURCE>","Condition":{"DateLessThan":
{"AWS:EpochTime":<EXPIRES>}}}]}';

my $POLICY_PARAM      = "Policy";
my $EXPIRES_PARAM     = "Expires";
my $SIGNATURE_PARAM   = "Signature";
my $KEY_PAIR_ID_PARAM = "Key-Pair-Id";

my $verbose = 0;
my $policy_filename = "";
my $expires_epoch = 0;
my $action = "";
my $help = 0;
my $key_pair_id = "";
```

```
my $url = "";
my $stream = "";
my $private_key_filename = "";

my $result = GetOptions("action=s"      => \$action,
                        "policy=s"      => \$policy_filename,
                        "expires=i"     => \$expires_epoch,
                        "private-key=s" => \$private_key_filename,
                        "key-pair-id=s" => \$key_pair_id,
                        "verbose"       => \$verbose,
                        "help"          => \$help,
                        "url=s"         => \$url,
                        "stream=s"      => \$stream,
                    );

if ($help or !$result) {
    pod2usage(1);
    exit;
}

if ($url eq "" and $stream eq "") {
    print STDERR "Must include a stream or a URL to encode or decode with the --stream
 or --url option\n";
    exit;
}

if ($url ne "" and $stream ne "") {
    print STDERR "Only one of --url and --stream may be specified\n";
    exit;
}

if ($url ne "" and !is_url_valid($url)) {
    exit;
}

if ($stream ne "") {
    exit unless is_stream_valid($stream);

    # The signing mechanism is identical, so from here on just pretend we're
    # dealing with a URL
    $url = $stream;
}

if ($action eq "encode") {
```

```
    # The encode action will generate a private content URL given a base URL,
    # a policy file (or an expires timestamp) and a key pair id parameter
    my $private_key;
    my $public_key;
    my $public_key_file;

    my $policy;
    if ($policy_filename eq "") {
        if ($expires_epoch == 0) {
            print STDERR "Must include policy filename with --policy argument or an
 expires" .
                         "time using --expires\n";
        }

        $policy = $CANNED_POLICY;
        $policy =~ s/<EXPIRES>/$expires_epoch/g;
        $policy =~ s/<RESOURCE>/$url/g;
    } else {
        if (! -e $policy_filename) {
            print STDERR "Policy file $policy_filename does not exist\n";
            exit;
        }
        $expires_epoch = 0; # ignore if set
        $policy = read_file($policy_filename);
    }

    if ($private_key_filename eq "") {
        print STDERR "You must specific the path to your private key file with --
private-key\n";
        exit;
    }

    if (! -e $private_key_filename) {
        print STDERR "Private key file $private_key_filename does not exist\n";
        exit;
    }

    if ($key_pair_id eq "") {
        print STDERR "You must specify a key pair id with --key-pair-id\n";
        exit;
    }

    my $encoded_policy = url_safe_base64_encode($policy);
    my $signature = rsa_sha1_sign($policy, $private_key_filename);
```

```
    my $encoded_signature = url_safe_base64_encode($signature);

    my $generated_url = create_url($url, $encoded_policy, $encoded_signature,
 $key_pair_id, $expires_epoch);


    if ($stream ne "") {
        print "Encoded stream (for use within a swf):\n" . $generated_url . "\n";
        print "Encoded and escaped stream (for use on a webpage):\n" .
 escape_url_for_webpage($generated_url) . "\n";
    } else {
        print "Encoded URL:\n" . $generated_url . "\n";
    }
} elsif ($action eq "decode") {
    my $decoded = decode_url($url);
    if (!$decoded) {
        print STDERR "Improperly formed URL\n";
        exit;
    }

    print_decoded_url($decoded);
} else {
    # No action specified, print help.  But only if this is run as a program (caller
 will be empty)
    pod2usage(1) unless caller();
}

# Decode a private content URL into its component parts
sub decode_url {
    my $url = shift;

    if ($url =~ /(.*)\?(.*)/) {
        my $base_url = $1;
        my $params = $2;

        my @unparsed_params = split(/&/, $params);
        my %params = ();
        foreach my $param (@unparsed_params) {
            my ($key, $val) = split(/=/, $param);
            $params{$key} = $val;
        }

        my $encoded_signature = "";
        if (exists $params{$SIGNATURE_PARAM}) {
```

```
            $encoded_signature = $params{"Signature"};
        } else {
            print STDERR "Missing Signature URL parameter\n";
            return 0;
        }

        my $encoded_policy = "";
        if (exists $params{$POLICY_PARAM}) {
            $encoded_policy = $params{$POLICY_PARAM};
        } else {
            if (!exists $params{$EXPIRES_PARAM}) {
                print STDERR "Either the Policy or Expires URL parameter needs to be
specified\n";
                return 0;
            }

            my $expires = $params{$EXPIRES_PARAM};

            my $policy = $CANNED_POLICY;
            $policy =~ s/<EXPIRES>/$expires/g;

            my $url_without_cf_params = $url;
            $url_without_cf_params =~ s/$SIGNATURE_PARAM=[^&]*&?//g;
            $url_without_cf_params =~ s/$POLICY_PARAM=[^&]*&?//g;
            $url_without_cf_params =~ s/$EXPIRES_PARAM=[^&]*&?//g;
            $url_without_cf_params =~ s/$KEY_PAIR_ID_PARAM=[^&]*&?//g;

            if ($url_without_cf_params =~ /(.*)\?$/) {
                $url_without_cf_params = $1;
            }

            $policy =~ s/<RESOURCE>/$url_without_cf_params/g;

            $encoded_policy = url_safe_base64_encode($policy);
        }

        my $key = "";
        if (exists $params{$KEY_PAIR_ID_PARAM}) {
            $key = $params{$KEY_PAIR_ID_PARAM};
        } else {
            print STDERR "Missing $KEY_PAIR_ID_PARAM parameter\n";
            return 0;
        }
```

```
        my $policy = url_safe_base64_decode($encoded_policy);

        my %ret = ();
        $ret{"base_url"} = $base_url;
        $ret{"policy"} = $policy;
        $ret{"key"} = $key;

        return \%ret;
    } else {
        return 0;
    }
}

# Print a decoded URL out
sub print_decoded_url {
    my $decoded = shift;

    print "Base URL: \n" . $decoded->{"base_url"} . "\n";
    print "Policy: \n" . $decoded->{"policy"} . "\n";
    print "Key: \n" . $decoded->{"key"} . "\n";
}

# Encode a string with base 64 encoding and replace some invalid URL characters
sub url_safe_base64_encode {
    my ($value) = @_;

    my $result = encode_base64($value);
    $result =~ tr|+=/|-_~|;

    return $result;
}

# Decode a string with base 64 encoding.  URL-decode the string first
# followed by reversing any special character ("+=/") translation.
sub url_safe_base64_decode {
    my ($value) = @_;

    $value =~ s/%([0-9A-Fa-f]{2})/chr(hex($1))/eg;
    $value =~ tr|-_~|+=/|;

    my $result = decode_base64($value);

    return $result;
}
```

```
# Create a private content URL
sub create_url {
    my ($path, $policy, $signature, $key_pair_id, $expires) = @_;

    my $result;
    my $separator = $path =~ /\?/ ? '&' : '?';
    if ($expires) {
        $result = "$path$separator$EXPIRES_PARAM=$expires&$SIGNATURE_PARAM=$signature&
$KEY_PAIR_ID_PARAM=$key_pair_id";
    } else {
        $result = "$path$separator$POLICY_PARAM=$policy&$SIGNATURE_PARAM=$signature&
$KEY_PAIR_ID_PARAM=$key_pair_id";
    }
    $result =~ s/\n//g;

    return $result;
}

# Sign a document with given private key file.
# The first argument is the document to sign
# The second argument is the name of the private key file
sub rsa_sha1_sign {
    my ($to_sign, $pvkFile) = @_;
    print "openssl sha1 -sign $pvkFile $to_sign\n";

    return write_to_program($pvkFile, $to_sign);
}

# Helper function to write data to a program
sub write_to_program {
my ($keyfile, $data) = @_;
unlink "temp_policy.dat" if (-e "temp_policy.dat");
unlink "temp_sign.dat" if (-e "temp_sign.dat");

write_file("temp_policy.dat", $data);

system("openssl dgst -sha1 -sign \"$keyfile\" -out temp_sign.dat temp_policy.dat");

my $output = read_file("temp_sign.dat");

    return $output;
}
```

```
# Read a file into a string and return the string
sub read_file {
    my ($file) = @_;

    open(INFILE, "<$file") or die("Failed to open $file: $!");
    my $str = join('', <INFILE>);
    close INFILE;

    return $str;
}

sub is_url_valid {
    my ($url) = @_;

    # HTTP distributions start with http[s]:// and are the correct thing to sign
    if ($url =~ /^https?:\/\//) {
        return 1;
    } else {
        print STDERR "CloudFront requires absolute URLs for HTTP distributions\n";
        return 0;
    }
}

sub is_stream_valid {
    my ($stream) = @_;

    if ($stream =~ /^rtmp:\/\// or $stream =~ /^\/?cfx\/st/) {
        print STDERR "Streaming distributions require that only the stream name is
 signed.\n";
        print STDERR "The stream name is everything after, but not including, cfx/st/
\n";
        return 0;
    } else {
        return 1;
    }
}

# flash requires that the query parameters in the stream name are url
# encoded when passed in through javascript, etc.  This sub handles the minimal
# required url encoding.
sub escape_url_for_webpage {
    my ($url) = @_;

    $url =~ s/\?/%3F/g;
```

```
    $url =~ s/=/%3D/g;
    $url =~ s/&/%26/g;

    return $url;
}

1;
```

Erstellen einer URL-Signatur mit PHP

Jeder Webserver, auf dem PHP ausgeführt wird, kann diesen PHP-Beispielcode verwenden, um Richtlinienerklärungen und Signaturen für private CloudFront Distributionen zu erstellen. Das vollständige Beispiel erstellt eine funktionierende Webseite mit signierten URL-Links, die einen Videostream per CloudFront Streaming abspielen. Sie können das vollständige Beispiel aus der Datei demo-php.zip herunterladen.

Hinweise

- Das Erstellen einer URL-Signatur ist nur ein Teil der Bereitstellung privater Inhalte über eine signierte URL. Weitere Informationen zum gesamten Prozess finden Sie unter Verwenden signierter URLs.
- Sie können signierte URLs auch mithilfe der `UrlSigner`-Klasse im AWS SDK für PHP erstellen. Weitere Informationen finden Sie unter Class UrlSigner in der AWS SDK für PHP API-Referenz.
- Beachten `openssl_sign` Sie beim Aufruf, dass die Übergabe `OPENSSL_ALGO_SHA256` als viertes Argument zu wechselt SHA-256. (Ein vollständiges Beispiel finden Sie auch unter.) Erstellen signierter Cookies mit PHP

Themen

- Erstellen Sie die RSA-Signatur SHA-1
- Erstellen einer vordefinierten Richtlinie
- Erstellen einer benutzerdefinierten Richtlinie
- Beispiel für vollständigen Code

In den folgenden Abschnitten wird das Codebeispiel in einzelne Teile unterteilt. Sie finden das Beispiel für vollständigen Code unten.

Erstellen Sie die RSA-Signatur SHA-1

Dieses Codebeispiel führt Folgendes aus:

- Die Funktion `rsa_sha1_sign` hasht und signiert die Richtlinienanweisung. Die erforderlichen Argumente sind eine Richtlinienanweisung und der private Schlüssel, der einem öffentlichen Schlüssel entspricht, der sich in einer vertrauenswürdigen Schlüsselgruppe für Ihre Distribution befindet.
- Als Nächstes erstellt die `url_safe_base64_encode` Funktion eine URL-safe Version der Signatur.

```
function rsa_sha1_sign($policy, $private_key_filename) {
    $signature = "";

    // load the private key
    $fp = fopen($private_key_filename, "r");
    $priv_key = fread($fp, 8192);
    fclose($fp);
    $pkeyid = openssl_get_privatekey($priv_key);

    // compute signature
    openssl_sign($policy, $signature, $pkeyid);

    // free the key from memory
    openssl_free_key($pkeyid);

    return $signature;
}

function url_safe_base64_encode($value) {
    $encoded = base64_encode($value);
    // replace unsafe characters +, = and / with
    // the safe characters -, _ and ~
    return str_replace(
        array('+', '=', '/'),
        array('-', '_', '~'),
        $encoded);
}
```

Der folgende Codeausschnitt verwendet die Funktionen `get_canned_policy_stream_name()` und `get_custom_policy_stream_name()` zum Erstellen einer vorgefertigten und

benutzerdefinierten Richtlinie. CloudFront verwendet die Richtlinien, um die URL für das Streamen des Videos zu erstellen, einschließlich der Angabe der Ablaufzeit.

Anschließend können Sie anhand einer vordefinierten Richtlinie oder einer benutzerdefinierten Richtlinie festlegen, wie der Zugriff auf Ihre Inhalte verwaltet werden soll. Weitere Informationen darüber, welche Sie wählen sollten, finden Sie im Abschnitt Entscheidung zwischen vordefinierten und benutzerdefinierten Richtlinien für signierte URLs.

Erstellen einer vordefinierten Richtlinie

Der folgende Beispielcode erstellt eine vordefinierte Richtlinienanweisung für die Signatur.

> **Note**
>
> Die `$expires` Variable ist ein date/time Stempel, der eine Ganzzahl und keine Zeichenfolge sein muss.

```
function get_canned_policy_stream_name($video_path, $private_key_filename,
 $key_pair_id, $expires) {
    // this policy is well known by CloudFront, but you still need to sign it, since it
 contains your parameters
    $canned_policy = '{"Statement":[{"Resource":"' . $video_path . '","Condition":
{"DateLessThan":{"AWS:EpochTime":'. $expires . '}}}]}';
    // the policy contains characters that cannot be part of a URL, so we base64 encode
 it
    $encoded_policy = url_safe_base64_encode($canned_policy);
    // sign the original policy, not the encoded version
    $signature = rsa_sha1_sign($canned_policy, $private_key_filename);
    // make the signature safe to be included in a URL
    $encoded_signature = url_safe_base64_encode($signature);

    // combine the above into a stream name
    $stream_name = create_stream_name($video_path, null, $encoded_signature,
 $key_pair_id, $expires);
    // URL-encode the query string characters
    return $stream_name;
}
```

Weitere Informationen zu vordefinierten Richtlinien finden Sie unter Erstellen einer signierten URL mit einer vordefinierten Richtlinie.

Erstellen einer benutzerdefinierten Richtlinie

Der folgende Beispielcode erstellt eine benutzerdefinierte Richtlinienanweisung für die Signatur.

```
function get_custom_policy_stream_name($video_path, $private_key_filename,
 $key_pair_id, $policy) {
    // the policy contains characters that cannot be part of a URL, so we base64 encode
 it
    $encoded_policy = url_safe_base64_encode($policy);
    // sign the original policy, not the encoded version
    $signature = rsa_sha1_sign($policy, $private_key_filename);
    // make the signature safe to be included in a URL
    $encoded_signature = url_safe_base64_encode($signature);

    // combine the above into a stream name
    $stream_name = create_stream_name($video_path, $encoded_policy, $encoded_signature,
 $key_pair_id, null);
    // URL-encode the query string characters
    return $stream_name;
}
```

Weitere Informationen zu benutzerdefinierten Richtlinien finden Sie unter Erstellen einer signierten URL mit einer benutzerdefinierten Richtlinie.

Beispiel für vollständigen Code

Der folgende Beispielcode bietet eine vollständige Demonstration der Erstellung CloudFront signierter URLs mit PHP. Sie können das vollständige Beispiel aus der Datei demo-php.zip herunterladen.

Im folgenden Beispiel können Sie das Element `$policy` `Condition` so anpassen, dass sowohl IPv4- und IPv6-Adressbereiche erlaubt sind. Ein Beispiel finden Sie unter Verwenden von IPv6-Adressen in IAM-Richtlinien im Benutzerhandbuch für Amazon Simple Storage Service.

```
<?php

function rsa_sha1_sign($policy, $private_key_filename) {
    $signature = "";

    // load the private key
    $fp = fopen($private_key_filename, "r");
    $priv_key = fread($fp, 8192);
    fclose($fp);
    $pkeyid = openssl_get_privatekey($priv_key);
```

```
    // compute signature
    openssl_sign($policy, $signature, $pkeyid);

    // free the key from memory
    openssl_free_key($pkeyid);

    return $signature;
}

function url_safe_base64_encode($value) {
    $encoded = base64_encode($value);
    // replace unsafe characters +, = and / with the safe characters -, _ and ~
    return str_replace(
        array('+', '=', '/'),
        array('-', '_', '~'),
        $encoded);
}

function create_stream_name($stream, $policy, $signature, $key_pair_id, $expires) {
    $result = $stream;
    // if the stream already contains query parameters, attach the new query parameters
 to the end
    // otherwise, add the query parameters
    $separator = strpos($stream, '?') == FALSE ? '?' : '&';
    // the presence of an expires time means we're using a canned policy
    if($expires) {
        $result .= $separator . "Expires=" . $expires . "&Signature=" . $signature .
 "&Key-Pair-Id=" . $key_pair_id;
    }
    // not using a canned policy, include the policy itself in the stream name
    else {
        $result .= $separator . "Policy=" . $policy . "&Signature=" . $signature .
 "&Key-Pair-Id=" . $key_pair_id;
    }

    // new lines would break us, so remove them
    return str_replace('\n', '', $result);
}


function get_canned_policy_stream_name($video_path, $private_key_filename,
 $key_pair_id, $expires) {
```

```
    // this policy is well known by CloudFront, but you still need to sign it, since it
 contains your parameters
    $canned_policy = '{"Statement":[{"Resource":"' . $video_path . '","Condition":
{"DateLessThan":{"AWS:EpochTime":'. $expires . '}}}]}';
    // the policy contains characters that cannot be part of a URL, so we base64 encode
 it
    $encoded_policy = url_safe_base64_encode($canned_policy);
    // sign the original policy, not the encoded version
    $signature = rsa_sha1_sign($canned_policy, $private_key_filename);
    // make the signature safe to be included in a URL
    $encoded_signature = url_safe_base64_encode($signature);

    // combine the above into a stream name
    $stream_name = create_stream_name($video_path, null, $encoded_signature,
 $key_pair_id, $expires);
    // URL-encode the query string characters
    return $stream_name;
}

function get_custom_policy_stream_name($video_path, $private_key_filename,
 $key_pair_id, $policy) {
    // the policy contains characters that cannot be part of a URL, so we base64 encode
 it
    $encoded_policy = url_safe_base64_encode($policy);
    // sign the original policy, not the encoded version
    $signature = rsa_sha1_sign($policy, $private_key_filename);
    // make the signature safe to be included in a URL
    $encoded_signature = url_safe_base64_encode($signature);

    // combine the above into a stream name
    $stream_name = create_stream_name($video_path, $encoded_policy, $encoded_signature,
 $key_pair_id, null);
    // URL-encode the query string characters
    return $stream_name;
}


// Path to your private key.  Be very careful that this file is not accessible
// from the web!

$private_key_filename = '/home/test/secure/example-priv-key.pem';
$key_pair_id = 'K2JCJMDEHXQW5F';
```

```
// Make sure you have "Restrict viewer access" enabled on this path behaviour and using
 the above Trusted key groups (recommended).
$video_path = 'https://example.com/secure/example.mp4';

$expires = time() + 300; // 5 min from now
$canned_policy_stream_name = get_canned_policy_stream_name($video_path,
 $private_key_filename, $key_pair_id, $expires);

// Get the viewer real IP from the x-forward-for header as $_SERVER['REMOTE_ADDR']
 will return viewer facing IP. An alternative option is to use CloudFront-Viewer-
Address header. Note that this header is a trusted CloudFront immutable header. Example
 format: IP:PORT ("CloudFront-Viewer-Address": "1.2.3.4:12345")
$client_ip = $_SERVER['HTTP_X_FORWARDED_FOR'];
$policy =
'{'.
    '"Statement":['.
        '{'.
            '"Resource":"'. $video_path . '",'.
            '"Condition":{'.
                '"IpAddress":{"AWS:SourceIp":"' . $client_ip . '/32"},'.
                '"DateLessThan":{"AWS:EpochTime":' . $expires . '}'.
            '}'.
        '}'.
    ']' .
    '}';
$custom_policy_stream_name = get_custom_policy_stream_name($video_path,
 $private_key_filename, $key_pair_id, $policy);

?>

<html>

<head>
    <title>CloudFront</title>
</head>

<body>
    <h1>Amazon CloudFront</h1>
    <h2>Canned Policy</h2>
    <h3>Expires at <?php echo gmdate('Y-m-d H:i:s T', $expires); ?></h3>
    <br />

    <div id='canned'>The canned policy video will be here: <br>
```

```
        <video width="640" height="360" autoplay muted controls>
        <source src="<?php echo $canned_policy_stream_name; ?>" type="video/mp4">
        Your browser does not support the video tag.
        </video>
    </div>

    <h2>Custom Policy</h2>
    <h3>Expires at <?php echo gmdate('Y-m-d H:i:s T', $expires); ?> only viewable by IP
 <?php echo $client_ip; ?></h3>
    <div id='custom'>The custom policy video will be here: <br>

         <video width="640" height="360" autoplay muted controls>
         <source src="<?php echo $custom_policy_stream_name; ?>" type="video/mp4">
         Your browser does not support the video tag.
        </video>
    </div>

</body>

</html>
```

Zusätzliche Beispiele für URL-Signaturen finden Sie in den folgenden Themen:

- Erstellen einer URL-Signatur mit Perl
- Erstellen einer URL-Signatur mithilfe von C# und dem .NET Framework
- Erstellen einer URL-Signatur mit Java

Anstatt signierte URLs zur Erstellung der Signatur zu verwenden, können Sie signierte Cookies verwenden. Weitere Informationen finden Sie unter Erstellen signierter Cookies mit PHP.

Erstellen einer URL-Signatur mithilfe von C# und dem .NET Framework

Die C#-Beispiele in diesem Abschnitt implementieren eine Beispielanwendung, die zeigt, wie die Signaturen für CloudFront private Distributionen mithilfe vorgefertigter und benutzerdefinierter Richtlinienanweisungen erstellt werden. Die Beispiele enthalten Dienstprogrammfunktionen auf der Grundlage des AWS SDK für .NET, das in .NET-Anwendungen nützlich sein kann.

Darüber hinaus können Sie signierte URLs und signierte Cookies mit dem SDK für .NET erstellen. Schlagen Sie unter den folgenden Themen in der API-Referenz für SDK für .NET nach:

- Signierte URLs — AmazonCloudFrontUrlSigner

- Signierte Cookies — AmazonCloudFrontCookieSigner

Um den Code herunterzuladen, navigieren Sie zu Signaturcode in C #.

> **Hinweise**
>
> - Die Klassen `AmazonCloudFrontCookieSigner` und `AmazonCloudFrontUrlSigner` wurden in ein separates Paket verschoben. Weitere Informationen zu ihrer Verwendung finden Sie unter CookieSigner und UrlSigner im AWS SDK für .NET (V4) Developer Guide.
> - Das Erstellen einer URL-Signatur ist nur ein Teil der Bereitstellung privater Inhalte über eine signierte URL. Weitere Informationen finden Sie unter Verwenden signierter URLs. Weitere Informationen über die Verwendung von signierten Cookies finden Sie unter Verwenden signierter Cookies.
> - Beachten Sie beim RSA-Signaturaufruf, dass dies durch den Parameter `SHA256` im Hash-Algorithmus ersetzt werden `SHA1` kann.

Verwenden eines RSA-Schlüssels im .NET-Framework

Um einen RSA-Schlüssel im.NET Framework zu verwenden, müssen Sie die AWS bereitgestellte PEM-Datei in das vom .NET Framework verwendete XML-Format konvertieren.

Nach der Konvertierung weist die Datei mit dem privaten RSA-Schlüssel das folgende Format auf:

Example: Privater RSA-Schlüssel im .NET Framework-XML-Format

```
<RSAKeyValue>
  <Modulus>
    wO5IvYCP5UcoCKDo1dcspoMehWBZcyfs9QEzGi6Oe5y+ewGr1oW+vB2GPB
    ANBiVPcUHTFWhwaIBd3oglmF0lGQljP/jOfmXHUK2kUUnLnJp+oOBL2NiuFtqcW6h/L5lIpD8Yq+NRHg
    Ty4zDsyr2880MvXv88yEFURCkqEXAMPLE=
  </Modulus>
  <Exponent>AQAB</Exponent>
  <P>
    5bmKDaTz
    npENGVqz4Cea8XPH+sxt+2VaAwYnsarVUoSBeVt8WLloVuZGG9IZYmH5KteXEu7fZveYd9UEXAMPLE==
  </P>
  <Q>
```

```
    1v9l/WN1a1N3rOK4VGoCokx7kR2SyTMSbZgF9IWJNOugR/WZw7HTnjipO3c9dy1Ms9pUKwUF4
    6d7049EXAMPLE==
  </Q>
  <DP>
    RgrSKuLWXMyBH+/l1Dx/I4tXuAJIrlPyo+VmiOc7b5NzHptkSHEPfR9s1
    OK0VqjknclqCJ3Ig86OMEtEXAMPLE==
  </DP>
  <DQ>
    pjPjvSFw+RoaTu0pgCA/jwW/FGyfN6iim1RFbkT4
    z49DZb2IM885f3vf35eLTaEYRYUHQgZtChNEV0TEXAMPLE==
  </DQ>
  <InverseQ>
    nkvOJTg5QtGNgWb9i
    cVtzrL/1pFEOHbJXwEJdU99N+7sMK+1066DL/HSBUCD63qD4USpnf0myc24in0EXAMPLE==</InverseQ>
  <D>
      Bc7mp7XYHynuPZxChjWNJZIq+A73gm0ASDv6At7F8Vi9r0xUlQe/v0AQS3ycN8QlyR4XMbzMLYk
      3yjxFDXo4ZKQtOGzLGteCU2srANiLv26/imXA8FVidZftTAtLviWQZBVPTeYIA69ATUYPEq0a5u5wjGy
      UOij9OWyuEXAMPLE=
   </D>
</RSAKeyValue>
```

Signaturmethode für vordefinierte Richtlinien in C#

Der folgende C#-Code erstellt eine signierte URL, die eine vordefinierte Richtlinie verwendet, indem die folgenden Schritte ausgeführt werden:

- Es wird eine Richtlinienanweisung erstellt.
- Hasht die Richtlinienanweisung mit SHA1 und signiert das Ergebnis mit RSA und dem privaten Schlüssel, dessen entsprechender öffentlicher Schlüssel sich in einer vertrauenswürdigen Schlüsselgruppe befindet.
- Base64-encodes die gehashte und signierte Policy-Anweisung und ersetzt Sonderzeichen, damit die Zeichenfolge sicher als URL-Anforderungsparameter verwendet werden kann.
- Verkettet die Werte.

Ein Beispiel für die vollständige Implementierung finden Sie unter Signaturcode in C#.

> **Note**
>
> Der `keyId` wird zurückgegeben, wenn Sie einen öffentlichen Schlüssel auf CloudFront hochladen. Weitere Informationen finden Sie unter

6

& Key-Pair-Id.

Example: Signaturmethode für vordefinierte Richtlinien in C#

```
public static string ToUrlSafeBase64String(byte[] bytes)
{
    return System.Convert.ToBase64String(bytes)
        .Replace('+', '-')
        .Replace('=', '_')
        .Replace('/', '~');
}

public static string CreateCannedPrivateURL(string urlString,
    string durationUnits, string durationNumber, string pathToPolicyStmnt,
    string pathToPrivateKey, string keyId)
{
    // args[] 0-thisMethod, 1-resourceUrl, 2-seconds-minutes-hours-days
    // to expiration, 3-numberOfPreviousUnits, 4-pathToPolicyStmnt,
    // 5-pathToPrivateKey, 6-keyId

    TimeSpan timeSpanInterval = GetDuration(durationUnits, durationNumber);

    // Create the policy statement.
    string strPolicy = CreatePolicyStatement(pathToPolicyStmnt,
        urlString,
        DateTime.Now,
        DateTime.Now.Add(timeSpanInterval),
        "0.0.0.0/0");
    if ("Error!" == strPolicy) return "Invalid time frame." +
        "Start time cannot be greater than end time.";

    // Copy the expiration time defined by policy statement.
    string strExpiration = CopyExpirationTimeFromPolicy(strPolicy);

    // Read the policy into a byte buffer.
    byte[] bufferPolicy = Encoding.ASCII.GetBytes(strPolicy);

    // Initialize the SHA1CryptoServiceProvider object and hash the policy data.
    using (SHA1CryptoServiceProvider
        cryptoSHA1 = new SHA1CryptoServiceProvider())
    {
```

```
        bufferPolicy = cryptoSHA1.ComputeHash(bufferPolicy);

        // Initialize the RSACryptoServiceProvider object.
        RSACryptoServiceProvider providerRSA = new RSACryptoServiceProvider();
        XmlDocument xmlPrivateKey = new XmlDocument();

        // Load your private key, which you created by converting your
        // .pem file to the XML format that the .NET framework uses.
        // Several tools are available.
        xmlPrivateKey.Load(pathToPrivateKey);

        // Format the RSACryptoServiceProvider providerRSA and
        // create the signature.
        providerRSA.FromXmlString(xmlPrivateKey.InnerXml);
        RSAPKCS1SignatureFormatter rsaFormatter =
            new RSAPKCS1SignatureFormatter(providerRSA);
        rsaFormatter.SetHashAlgorithm("SHA1");
        byte[] signedPolicyHash = rsaFormatter.CreateSignature(bufferPolicy);

        // Convert the signed policy to URL-safe base64 encoding and
        // replace unsafe characters + = / with the safe characters - _ ~
        string strSignedPolicy = ToUrlSafeBase64String(signedPolicyHash);

        // Concatenate the URL, the timestamp, the signature,
        // and the key pair ID to form the signed URL.
        return urlString +
            "?Expires=" +
            strExpiration +
            "&Signature=" +
            strSignedPolicy +
            "&Key-Pair-Id=" +
            keyId;
    }
}
```

Signaturmethode für benutzerdefinierte Richtlinien in C#

Der folgende C#-Code erstellt eine signierte URL, die eine benutzerdefinierte Richtlinie verwendet, indem die folgenden Schritte ausgeführt werden:

1. Es wird eine Richtlinienanweisung erstellt.
2. Base64-encodes die Richtlinienanweisung und ersetzt Sonderzeichen, damit die Zeichenfolge sicher als URL-Anforderungsparameter verwendet werden kann.

3. Hasht die Richtlinienanweisung mit SHA1 und verschlüsselt das Ergebnis mit RSA und dem privaten Schlüssel, dessen entsprechender öffentlicher Schlüssel sich in einer vertrauenswürdigen Schlüsselgruppe befindet.
4. Base64-encodes die Hash-Richtlinienanweisung und das Ersetzen von Sonderzeichen, damit die Zeichenfolge sicher als URL-Anforderungsparameter verwendet werden kann.
5. Verkettet die Werte.

Ein Beispiel für die vollständige Implementierung finden Sie unter Signaturcode in C#.

> **Note**
>
> Der `keyId` wird zurückgegeben, wenn Sie einen öffentlichen Schlüssel auf CloudFront hochladen. Weitere Informationen finden Sie unter
>
> 
>
> & Key-Pair-Id.

Example: Signaturmethode für benutzerdefinierte Richtlinien in C#

```
public static string ToUrlSafeBase64String(byte[] bytes)
{
    return System.Convert.ToBase64String(bytes)
        .Replace('+', '-')
        .Replace('=', '_')
        .Replace('/', '~');
}

public static string CreateCustomPrivateURL(string urlString,
    string durationUnits, string durationNumber, string startIntervalFromNow,
    string ipaddress, string pathToPolicyStmnt, string pathToPrivateKey,
    string keyId)
{
    // args[] 0-thisMethod, 1-resourceUrl, 2-seconds-minutes-hours-days
    // to expiration, 3-numberOfPreviousUnits, 4-starttimeFromNow,
    // 5-ip_address, 6-pathToPolicyStmt, 7-pathToPrivateKey, 8-keyId

    TimeSpan timeSpanInterval = GetDuration(durationUnits, durationNumber);
    TimeSpan timeSpanToStart = GetDurationByUnits(durationUnits,
        startIntervalFromNow);
    if (null == timeSpanToStart)
```

```
        return "Invalid duration units." +
            "Valid options: seconds, minutes, hours, or days";

    string strPolicy = CreatePolicyStatement(
        pathToPolicyStmnt, urlString, DateTime.Now.Add(timeSpanToStart),
        DateTime.Now.Add(timeSpanInterval), ipaddress);

    // Read the policy into a byte buffer.
    byte[] bufferPolicy = Encoding.ASCII.GetBytes(strPolicy);

    // Convert the policy statement to URL-safe base64 encoding and
    // replace unsafe characters + = / with the safe characters - _ ~

    string urlSafePolicy = ToUrlSafeBase64String(bufferPolicy);

    // Initialize the SHA1CryptoServiceProvider object and hash the policy data.
    byte[] bufferPolicyHash;
    using (SHA1CryptoServiceProvider cryptoSHA1 =
        new SHA1CryptoServiceProvider())
    {
        bufferPolicyHash = cryptoSHA1.ComputeHash(bufferPolicy);

        // Initialize the RSACryptoServiceProvider object.
        RSACryptoServiceProvider providerRSA = new RSACryptoServiceProvider();
        XmlDocument xmlPrivateKey = new XmlDocument();

        // Load your private key, which you created by converting your
        // .pem file to the XML format that the .NET framework uses.
        // Several tools are available.
        xmlPrivateKey.Load(pathToPrivateKey);

        // Format the RSACryptoServiceProvider providerRSA
        // and create the signature.
        providerRSA.FromXmlString(xmlPrivateKey.InnerXml);
        RSAPKCS1SignatureFormatter RSAFormatter =
            new RSAPKCS1SignatureFormatter(providerRSA);
        RSAFormatter.SetHashAlgorithm("SHA1");
        byte[] signedHash = RSAFormatter.CreateSignature(bufferPolicyHash);

        // Convert the signed policy to URL-safe base64 encoding and
        // replace unsafe characters + = / with the safe characters - _ ~
        string strSignedPolicy = ToUrlSafeBase64String(signedHash);

        return urlString +
```

```
            "?Policy=" +
            urlSafePolicy +
            "&Signature=" +
            strSignedPolicy +
            "&Key-Pair-Id=" +
            keyId;
    }
}
```

Dienstprogrammmethoden für die Generierung von Signaturen

Mit den folgenden Methoden lassen sich die Richtlinienanweisung aus einer Datei abrufen und Zeitintervalle für die Generierung von Signaturen parsen.

Example: Dienstprogrammmethoden für die Generierung von Signaturen

```
public static string CreatePolicyStatement(string policyStmnt,
   string resourceUrl,
   DateTime startTime,
   DateTime endTime,
   string ipAddress)

{
   // Create the policy statement.
   FileStream streamPolicy = new FileStream(policyStmnt, FileMode.Open,
 FileAccess.Read);
   using (StreamReader reader = new StreamReader(streamPolicy))
   {
      string strPolicy = reader.ReadToEnd();

      TimeSpan startTimeSpanFromNow = (startTime - DateTime.Now);
      TimeSpan endTimeSpanFromNow = (endTime - DateTime.Now);
      TimeSpan intervalStart =
         (DateTime.UtcNow.Add(startTimeSpanFromNow)) -
         new DateTime(1970, 1, 1, 0, 0, 0, DateTimeKind.Utc);
      TimeSpan intervalEnd =
         (DateTime.UtcNow.Add(endTimeSpanFromNow)) -
         new DateTime(1970, 1, 1, 0, 0, 0, DateTimeKind.Utc);

      int startTimestamp = (int)intervalStart.TotalSeconds; // START_TIME
      int endTimestamp = (int)intervalEnd.TotalSeconds;  // END_TIME

      if (startTimestamp > endTimestamp)
         return "Error!";
```

```
      // Replace variables in the policy statement.
      strPolicy = strPolicy.Replace("RESOURCE", resourceUrl);
      strPolicy = strPolicy.Replace("START_TIME", startTimestamp.ToString());
      strPolicy = strPolicy.Replace("END_TIME", endTimestamp.ToString());
      strPolicy = strPolicy.Replace("IP_ADDRESS", ipAddress);
      strPolicy = strPolicy.Replace("EXPIRES", endTimestamp.ToString());
      return strPolicy;
   }
}

public static TimeSpan GetDuration(string units, string numUnits)
{
   TimeSpan timeSpanInterval = new TimeSpan();
   switch (units)
   {
      case "seconds":
         timeSpanInterval = new TimeSpan(0, 0, 0, int.Parse(numUnits));
         break;
      case "minutes":
         timeSpanInterval = new TimeSpan(0, 0, int.Parse(numUnits), 0);
         break;
      case "hours":
         timeSpanInterval = new TimeSpan(0, int.Parse(numUnits), 0 ,0);
         break;
      case "days":
         timeSpanInterval = new TimeSpan(int.Parse(numUnits),0 ,0 ,0);
         break;
      default:
         Console.WriteLine("Invalid time units;" +
            "use seconds, minutes, hours, or days");
         break;
   }
   return timeSpanInterval;
}

private static TimeSpan GetDurationByUnits(string durationUnits,
   string startIntervalFromNow)
{
   switch (durationUnits)
   {
      case "seconds":
         return new TimeSpan(0, 0, int.Parse(startIntervalFromNow));
      case "minutes":
```

```
            return new TimeSpan(0, int.Parse(startIntervalFromNow), 0);
         case "hours":
            return new TimeSpan(int.Parse(startIntervalFromNow), 0, 0);
         case "days":
            return new TimeSpan(int.Parse(startIntervalFromNow), 0, 0, 0);
         default:
            return new TimeSpan(0, 0, 0, 0);
      }
   }

   public static string CopyExpirationTimeFromPolicy(string policyStatement)
   {
      int startExpiration = policyStatement.IndexOf("EpochTime");
      string strExpirationRough = policyStatement.Substring(startExpiration +
         "EpochTime".Length);
      char[] digits = { '0', '1', '2', '3', '4', '5', '6', '7', '8', '9' };

      List<char> listDigits = new List<char>(digits);
      StringBuilder buildExpiration = new StringBuilder(20);

      foreach (char c in strExpirationRough)
      {
         if (listDigits.Contains(c))
            buildExpiration.Append(c);
      }
      return buildExpiration.ToString();
   }
```

Weitere Informationen finden Sie auch unter

- Erstellen einer URL-Signatur mit Perl
- Erstellen einer URL-Signatur mit PHP
- Erstellen einer URL-Signatur mit Java

Erstellen einer URL-Signatur mit Java

Zusätzlich zum folgenden Codebeispiel können Sie die `CloudFrontUrlSigner` Utility-Klasse in AWS SDK für Java (Version 1) verwenden, um CloudFront signierte URLs zu erstellen.

Weitere Beispiele finden Sie unter Erstellen signierter URLs und Cookies mithilfe eines AWS SDK in der Codebibliothek für AWS SDK-Codebeispiele.

Hinweise

- Das Erstellen einer signierten URL ist nur ein Teil des Prozesses, mit dem private Inhalte bereitgestellt CloudFront werden. Weitere Informationen zum gesamten Prozess finden Sie unter Verwenden signierter URLs.
- Beachten `Signature.getInstance` Sie im Anruf, dass dies durch ersetzt werden `SHA1withRSA` kann`SHA256withRSA`.

Example Java-Richtlinie und Verschlüsselungsmethoden für Signaturen

```
package org.example;

import java.time.Instant;
import java.time.temporal.ChronoUnit;
import software.amazon.awssdk.services.cloudfront.CloudFrontUtilities;
import software.amazon.awssdk.services.cloudfront.model.CannedSignerRequest;
import software.amazon.awssdk.services.cloudfront.url.SignedUrl;

public class Main {

    public static void main(String[] args) throws Exception {
        CloudFrontUtilities cloudFrontUtilities = CloudFrontUtilities.create();
        Instant expirationDate = Instant.now().plus(7, ChronoUnit.DAYS);
        String resourceUrl = "https://a1b2c3d4e5f6g7.cloudfront.net";
        String keyPairId = "K1UA3WV15I7JSD";
        CannedSignerRequest cannedRequest = CannedSignerRequest.builder()
                .resourceUrl(resourceUrl)
                .privateKey(new java.io.File("/path/to/private_key.pem").toPath())
                .keyPairId(keyPairId)
                .expirationDate(expirationDate)
                .build();
        SignedUrl signedUrl =
 cloudFrontUtilities.getSignedUrlWithCannedPolicy(cannedRequest);
        String url = signedUrl.url();
        System.out.println(url);

    }
}
```

Example Beispiel für vordefinierte Policy-Signaturen mit SHA256 in Java

```
package org.example;

import java.io.File;
import java.nio.file.Files;
import java.security.KeyFactory;
import java.security.PrivateKey;
import java.security.Signature;
import java.security.spec.PKCS8EncodedKeySpec;
import java.time.Instant;
import java.time.temporal.ChronoUnit;
import java.util.Base64;

public class Main {

    public static void main(String[] args) throws Exception {
        String resourceUrl = "https://a1b2c3d4e5f6g7.cloudfront.net/myfile.html";
        String keyPairId = "K1UA3WV15I7JSD";
        Instant expiration = Instant.now().plus(7, ChronoUnit.DAYS);
        PrivateKey privateKey = loadPrivateKey("/path/to/private_key.der");

        System.out.println(createSignedUrl(resourceUrl, keyPairId, privateKey,
 expiration, "SHA1"));
        System.out.println(createSignedUrl(resourceUrl, keyPairId, privateKey,
 expiration, "SHA256"));
    }

    static String createSignedUrl(String resourceUrl, String keyPairId,
                                  PrivateKey privateKey, Instant expiration,
                                  String hashAlgorithm) throws Exception {
        long epochSeconds = expiration.getEpochSecond();

        String policy = "{\"Statement\":[{\"Resource\":\"" + resourceUrl
                + "\",\"Condition\":{\"DateLessThan\":{\"AWS:EpochTime\":" +
 epochSeconds + "}}}]}";

        String jcaAlgorithm = hashAlgorithm.equals("SHA256") ? "SHA256withRSA" :
 "SHA1withRSA";

        Signature sig = Signature.getInstance(jcaAlgorithm);
        sig.initSign(privateKey);
        sig.update(policy.getBytes("UTF-8"));
        String signature = base64UrlEncode(sig.sign());
```

```
        String url = resourceUrl
                + (resourceUrl.contains("?") ? "&" : "?")
                + "Expires=" + epochSeconds
                + "&Signature=" + signature
                + "&Key-Pair-Id=" + keyPairId;

        if (hashAlgorithm.equals("SHA256")) {
            url += "&Hash-Algorithm=SHA256";
        }

        return url;
    }

    static String base64UrlEncode(byte[] bytes) {
        return Base64.getEncoder().encodeToString(bytes)
                .replace('+', '-')
                .replace('=', '_')
                .replace('/', '~');
    }

    static PrivateKey loadPrivateKey(String path) throws Exception {
        byte[] keyBytes = Files.readAllBytes(new File(path).toPath());
        return KeyFactory.getInstance("RSA")
                .generatePrivate(new PKCS8EncodedKeySpec(keyBytes));
    }
}
```

Weitere Informationen finden Sie auch unter:

- Erstellen einer URL-Signatur mit Perl
- Erstellen einer URL-Signatur mit PHP
- Erstellen einer URL-Signatur mithilfe von C# und dem .NET Framework

Beschränken Sie den Zugriff auf ein AWS Ursprungsserver

Sie können einige AWS Ursprünge so konfigurieren CloudFront , dass sie die folgenden Vorteile bieten:

- Schränkt den Zugriff auf den AWS Ursprung ein, sodass er nicht öffentlich zugänglich ist.

- Stellt sicher, dass Zuschauer (Benutzer) nur über die angegebene CloudFront Distribution auf die AWS Inhalte im Quelltext zugreifen können. Dadurch wird verhindert, dass Zuschauer direkt von der Quelle aus oder über eine unbeabsichtigte CloudFront Verbreitung auf die Inhalte zugreifen.

Stellen Sie dazu so ein, CloudFront dass authentifizierte Anfragen an Ihren AWS Absender gesendet werden, und konfigurieren Sie den AWS Absender so, dass nur authentifizierte Anfragen von zugestellt werden. CloudFront Weitere Informationen zu kompatiblen Herkunftstypen finden Sie in den folgenden Themen. AWS

Topics

- Beschränken Sie den Zugriff auf einen AWS Elemental MediaPackage v2-Ursprung
- Beschränken Sie den Zugriff auf ein AWS Elemental MediaStore Ursprungsserver
- Beschränken Sie den Zugriff auf ein AWS Lambda Herkunft der Funktions-URL
- Beschränken Sie den Zugriff auf ein AWS Lambda Ursprung der Webfunktion
- Beschränken des Zugriffs auf einen Amazon-S3-Ursprung
- Zugriffsbeschränkung mit VPC-Ursprüngen
- Beschränken Sie den Zugriff auf einen Amazon S3 Multi-Region Access Point-Ursprung

Beschränken Sie den Zugriff auf einen AWS Elemental MediaPackage v2-Ursprung

CloudFront bietet Origin Access Control (OAC) zur Beschränkung des Zugriffs auf einen MediaPackage v2-Ursprung.

> **Note**
>
> CloudFront OAC unterstützt nur v2. MediaPackage MediaPackage Version 1 wird nicht unterstützt.

Themen

- Erstellen einer neuen OAC
- Erweiterte Einstellungen für die Ursprungszugriffssteuerung

Erstellen einer neuen OAC

Führen Sie die in den folgenden Themen beschriebenen Schritte aus, um ein neues OAC in CloudFront einzurichten.

Themen

- Voraussetzungen
- Erteilen Sie CloudFront die Erlaubnis, auf den MediaPackage v2-Ursprung zuzugreifen
- Erstellen der OAC

Voraussetzungen

Bevor Sie OAC erstellen und einrichten, müssen Sie über eine CloudFront Distribution mit MediaPackage v2-Ursprung verfügen. Weitere Informationen finden Sie unter Verwenden Sie einen MediaStore Container oder einen MediaPackage Kanal.

Erteilen Sie CloudFront die Erlaubnis, auf den MediaPackage v2-Ursprung zuzugreifen

Bevor Sie ein OAC erstellen oder es in einer CloudFront Distribution einrichten, stellen Sie sicher, dass es über die Zugriffsrechte für den MediaPackage v2-Ursprung CloudFront verfügt. Tun Sie dies, nachdem Sie eine CloudFront Distribution erstellt haben, aber bevor Sie das OAC dem MediaPackage v2-Ursprung in der Distributionskonfiguration hinzufügen.

Verwenden Sie eine IAM-Richtlinie, um dem CloudFront Dienstprinzipal (`cloudfront.amazonaws.com`) den Zugriff auf den Ursprung zu ermöglichen. Das `Condition` Element in der Richtlinie ermöglicht den CloudFront Zugriff auf den MediaPackage v2-Ursprung nur, wenn die Anfrage im Namen der CloudFront Distribution erfolgt, die den MediaPackage v2-Ursprung enthält. Dies ist die Distribution mit dem MediaPackage v2-Ursprung, zu der Sie OAC hinzufügen möchten.

Example: IAM-Richtlinie, die nur Lesezugriff für eine CloudFront Distribution mit aktiviertem OAC ermöglicht

Die folgende Richtlinie ermöglicht der CloudFront Distribution (*`E1PDK09ESKHJWT`*) den Zugriff auf den v2-Ursprung. MediaPackage Der Ursprung ist der für das `Resource`-Element angegebene ARN.

JSON

```
{
```

```
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowCloudFrontServicePrincipal",
            "Effect": "Allow",
            "Principal": {"Service": "cloudfront.amazonaws.com"},
            "Action": "mediapackagev2:GetObject",
            "Resource": "arn:aws:mediapackagev2:us-
east-1:123456789012:channelGroup/channel-group-name/channel/channel-name/
originEndpoint/origin_endpoint_name",
            "Condition": {
                "StringEquals": {"AWS:SourceArn":
 "arn:aws:cloudfront::123456789012:distribution/E1PDK09ESKHJWT"}
            }
        }
    ]
}
```

Hinweise

- Wenn Sie die MQAR-Funktion und die Ursprungszugriffssteuerung (OAC) aktiviert haben, fügen Sie die Aktion `mediapackagev2:GetHeadObject` der IAM-Richtlinie hinzu. MQAR benötigt diese Berechtigung, um HEAD Anfragen an den MediaPackage v2-Ursprung zu senden. Weitere Informationen zu MQAR finden Sie unter Media Quality-Aware Resiliency.
- Wenn Sie eine Distribution erstellen, die keine Berechtigungen für Ihren MediaPackage v2-Ursprung hat, können Sie in der CloudFront Konsole „Richtlinie kopieren“ und dann „Endpunktberechtigungen aktualisieren“ auswählen. Anschließend können Sie die kopierte Berechtigung an den Endpunkt anhängen. Weitere Informationen finden Sie unter Endpunktrichtlinienfelder im Benutzerhandbuch für AWS Elemental MediaPackage .

Erstellen der OAC

Um ein OAC zu erstellen, können Sie die AWS-Managementkonsole, CloudFormation AWS CLI, oder die CloudFront API verwenden.

Console

So erstellen Sie eine OAC

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home.
2. Wählen Sie im Navigationsbereich Origin access (Ursprungszugriff) aus.
3. Wählen Sie Create control setting (Kontrolleinstellung erstellen) aus.
4. Führen Sie im Formular Neue OAC erstellen die folgenden Schritte aus:
   a. Geben Sie für die OAC einen Namen und (optional) eine Beschreibung ein.
   b. Es empfiehlt sich, unter Signierverhalten die Standardeinstellung Anforderungen signieren (empfohlen) zu belassen. Weitere Informationen finden Sie unter the section called “Erweiterte Einstellungen für die Ursprungszugriffssteuerung”.
5. Wählen Sie als Origin-Typ MediaPackage V2 aus.
6. Wählen Sie Erstellen aus.

> **Tip**
>
> Nachdem Sie die OAC erstellt haben, notieren Sie sich den Namen. Sie benötigen diesen im folgenden Verfahren.

Um einem MediaPackage v2-Ursprung in einer Distribution ein OAC hinzuzufügen

1. Öffnen Sie die CloudFront Konsole unter. https://console.aws.amazon.com/cloudfront/v4/home
2. Wählen Sie eine Distribution mit einem MediaPackage V2-Ursprung aus, zu der Sie das OAC hinzufügen möchten, und wählen Sie dann den Tab Origins.
3. Wählen Sie den MediaPackage v2-Ursprung aus, zu dem Sie das OAC hinzufügen möchten, und wählen Sie dann Bearbeiten.
4. Wählen Sie HTTPS only (Nur HTTPS) für Protocol (Protokoll) Ihres Ursprungs aus.
5. Wählen Sie im Dropdown-Menü Ursprungszugriffssteuerung den zu verwendenden OAC-Namen aus.
6. Wählen Sie Änderungen speichern aus.

Die Verteilung beginnt mit der Bereitstellung an allen CloudFront Edge-Standorten. Wenn ein Edge-Standort die neue Konfiguration empfängt, signiert er alle Anfragen, die er an den MediaPackage v2-Ursprung sendet.

CloudFormation

Verwenden Sie den `AWS::CloudFront::OriginAccessControl` Ressourcentyp CloudFormation, um ein OAC mit zu erstellen. Das folgende Beispiel zeigt die CloudFormation Vorlagensyntax im YAML-Format für die Erstellung eines OAC.

```
Type: AWS::CloudFront::OriginAccessControl
Properties:
  OriginAccessControlConfig:
      Description: An optional description for the origin access control
      Name: ExampleOAC
      OriginAccessControlOriginType: mediapackagev2
      SigningBehavior: always
      SigningProtocol: sigv4
```

Weitere Informationen finden Sie im AWS CloudFormation Benutzerhandbuch unter AWS::CloudFront::OriginAccessSteuerung.

CLI

Verwenden Sie den aws cloudfront create-origin-access-control Befehl, um eine Origin-Zugriffskontrolle mit dem AWS Command Line Interface (AWS CLI) zu erstellen. Sie können eine Eingabedatei verwenden, um die Eingabeparameter für den Befehl bereitzustellen, anstatt jeden einzelnen Parameter als Befehlszeileneingabe anzugeben.

So erstellen Sie eine Ursprungszugriffssteuerung (CLI mit Eingabedatei)

1. Verwenden Sie den folgenden Befehl zum Erstellen einer Datei mit dem Namen `origin-access-control.yaml`. Diese Datei enthält alle Eingabeparameter für den Befehl create-origin-access-control.

   ```
   aws cloudfront create-origin-access-control --generate-cli-skeleton yaml-input >
    origin-access-control.yaml
   ```

2. Öffnen Sie die Datei `origin-access-control.yaml`, die Sie gerade erstellt haben. Bearbeiten Sie die Datei, um einen Namen für die OAC und eine Beschreibung (optional)

hinzuzufügen, und ändern Sie `SigningBehavior` zu `always`. Speichern Sie dann die Datei.

Weitere Informationen zu anderen OAC-Einstellungen finden Sie unter the section called “Erweiterte Einstellungen für die Ursprungszugriffssteuerung”.

3. Verwenden Sie den folgenden Befehl, um die Ursprungszugriffssteuerung mit Eingabeparametern aus der Datei `origin-access-control.yaml` zu erstellen.

```
aws cloudfront create-origin-access-control --cli-input-yaml file://origin-
access-control.yaml
```

Notieren Sie den `Id`-Wert in der Befehlsausgabe. Sie benötigen ihn, um das OAC zu einem MediaPackage v2-Ursprung in einer CloudFront Distribution hinzuzufügen.

Um ein OAC an einen MediaPackage v2-Ursprung in einer vorhandenen Distribution anzuhängen (CLI mit Eingabedatei)

1. Verwenden Sie den folgenden Befehl, um die Verteilungskonfiguration für die CloudFront Distribution zu speichern, zu der Sie das OAC hinzufügen möchten. Die Distribution muss einen MediaPackage v2-Ursprung haben.

```
aws cloudfront get-distribution-config --id <CloudFront distribution ID> --
output yaml > dist-config.yaml
```

2. Öffnen Sie die Datei mit dem Namen `dist-config.yaml`, die Sie gerade erstellt haben. Bearbeiten Sie die Datei und nehmen Sie die folgenden Änderungen vor:

   - Fügen Sie im Objekt `Origins` die ID der OAC dem Feld namens `OriginAccessControlId` hinzu.
   - Entfernen Sie den Wert aus dem Feld namens `OriginAccessIdentity`, sofern vorhanden.
   - Benennen Sie das Feld `ETag` in `IfMatch` um, ändern Sie jedoch nicht den Wert des Feldes.

   Speichern Sie die Datei, wenn Sie fertig sind.

3. Verwenden Sie den folgenden Befehl, um die Distribution zu aktualisieren und die Ursprungszugriffssteuerung zu verwenden.

```
aws cloudfront update-distribution --id <CloudFront distribution ID> --cli-
input-yaml file://dist-config.yaml
```

Die Verteilung beginnt mit der Bereitstellung an allen CloudFront Edge-Standorten. Wenn ein Edge-Standort die neue Konfiguration empfängt, signiert er alle Anfragen, die er an den MediaPackage v2-Ursprung sendet.

API

Um ein OAC mit der CloudFront API zu erstellen, verwenden Sie CreateOriginAccessControl. Weitere Informationen zu den Feldern, die Sie in diesem API-Aufruf angeben, finden Sie in der API-Referenzdokumentation für Ihr AWS SDK oder einen anderen API-Client.

Nachdem Sie ein OAC erstellt haben, können Sie es mithilfe eines der folgenden API-Aufrufe an einen MediaPackage v2-Ursprung in einer Distribution anhängen:

- Um es an eine bestehende Distribution anzuhängen, verwenden Sie UpdateDistribution.
- Um es an eine neue Distribution anzuhängen, verwenden Sie CreateDistribution.

Geben Sie für beide API-Aufrufe die ID der OAC im Feld `OriginAccessControlId` innerhalb eines Ursprungs an. Weitere Informationen zu den anderen Feldern, die Sie in diesen API-Aufrufen angeben, finden Sie unter Referenz für alle Distributionseinstellungen und in der API-Referenzdokumentation für Ihr AWS SDK oder einen anderen API-Client.

Erweiterte Einstellungen für die Ursprungszugriffssteuerung

Die CloudFront OAC-Funktion umfasst erweiterte Einstellungen, die nur für bestimmte Anwendungsfälle vorgesehen sind. Verwenden Sie die empfohlenen Einstellungen, sofern Sie die erweiterten Einstellungen nicht speziell benötigen.

OAC enthält eine Einstellung mit dem Namen Signaturverhalten (in der Konsole) oder `SigningBehavior` (in der API, CLI und CloudFormation). Diese Einstellung bietet die folgenden Optionen:

Ursprungsanforderungen immer signieren (empfohlene Einstellung)

Wir empfehlen die Verwendung dieser Einstellung mit der Bezeichnung Sign requests (recommended) (Anforderungen signieren (empfohlen)) in der Konsole bzw. `always` in der API, CLI und CloudFormation. Mit dieser Einstellung werden CloudFront immer alle Anfragen signiert, die an den MediaPackage v2-Ursprung gesendet werden.

Ursprungsanforderungen nie signieren

Diese Einstellung heißt Do not sign requests (Anforderungen nicht signieren) in der Konsole bzw. `never` in der API, CLI und CloudFormation. Verwenden Sie diese Einstellung, um die OAC für alle Ursprünge in allen Distributionen zu deaktivieren, die diese OAC verwenden. Auf diese Weise lässt sich im Vergleich zum einzelnen Entfernen einer OAC aus allen Ursprüngen und Distributionen, die diese verwenden, Zeit und Aufwand sparen. Mit dieser Einstellung signiert es CloudFront keine Anfragen, die es an den MediaPackage v2-Ursprung sendet.

> **Warning**
>
> Um diese Einstellung verwenden zu können, muss der MediaPackage v2-Ursprung öffentlich zugänglich sein. Wenn Sie diese Einstellung mit einem MediaPackage v2-Ursprung verwenden, der nicht öffentlich zugänglich ist, CloudFront können Sie nicht auf den Ursprung zugreifen. Der MediaPackage v2-Ursprung gibt Fehler an die Zuschauer zurück CloudFront und CloudFront leitet diese Fehler an sie weiter. Weitere Informationen finden Sie im AWS Elemental MediaPackage Benutzerhandbuch MediaPackage im Beispiel für eine MediaPackage v2-Richtlinie für Richtlinien und Berechtigungen.

Viewer (Client)-**`Authorization`**-Header nicht überschreiben

Diese Einstellung heißt Do not override authorization header (Autorisierungsheader nicht überschreiben) in der Konsole bzw. `no-override` in der API, CLI und CloudFormation. Verwenden Sie diese Einstellung, wenn Sie Originalanfragen nur signieren CloudFront möchten, wenn die entsprechende Viewer-Anfrage keinen `Authorization` Header enthält. Mit dieser Einstellung wird der `Authorization` Header der Viewer-Anfrage weitergegeben, CloudFront wenn eine vorhanden ist, signiert aber die ursprüngliche Anfrage (fügt einen eigenen `Authorization` Header hinzu), wenn die Viewer-Anfrage keinen `Authorization` Header enthält.

> **Warning**
>
> Um den `Authorization` Header aus der Viewer-Anfrage weiterzugeben, müssen Sie den `Authorization` Header zu einer Cache-Richtlinie für alle Cache-Verhaltensweisen hinzufügen, die MediaPackage v2-Ursprünge verwenden, die mit dieser ursprünglichen Zugriffskontrolle verknüpft sind.

Beschränken Sie den Zugriff auf ein AWS Elemental MediaStore Ursprungsserver

CloudFront bietet Origin Access Control (OAC), um den Zugriff auf einen Ursprung einzuschränken. AWS Elemental MediaStore

Themen

- Erstellen einer neuen Ursprungszugriffssteuerung
- Erweiterte Einstellungen für die Ursprungszugriffssteuerung

Erstellen einer neuen Ursprungszugriffssteuerung

Führen Sie die in den folgenden Themen beschriebenen Schritte aus, um eine neue Origin-Zugriffskontrolle in einzurichten. CloudFront

Themen

- Voraussetzungen
- Erteilen Sie die CloudFront Erlaubnis, auf den MediaStore Ursprung zuzugreifen
- Erstellen der Ursprungszugriffssteuerung

Voraussetzungen

Bevor Sie die Origin-Zugriffskontrolle erstellen und einrichten, müssen Sie über eine CloudFront Distribution mit einem MediaStore Ursprung verfügen.

Erteilen Sie die CloudFront Erlaubnis, auf den MediaStore Ursprung zuzugreifen

Bevor du eine Zugriffskontrolle für den Ursprung erstellst oder sie in einer CloudFront Distribution einrichtest, vergewissere dich, dass diese CloudFront Person über Zugriffsberechtigungen für den

MediaStore Ursprung verfügt. Tun Sie dies, nachdem Sie eine CloudFront Distribution erstellt haben, aber bevor Sie das OAC zum MediaStore Ursprung in der Distributionskonfiguration hinzufügen.

Verwenden Sie eine MediaStore Container-Richtlinie, um dem CloudFront Dienstprinzipal (`cloudfront.amazonaws.com`) den Zugriff auf den Ursprung zu ermöglichen. Verwenden Sie ein `Condition` Element in der Richtlinie, um nur dann Zugriff auf den MediaStore Container CloudFront zu gewähren, wenn die Anfrage im Namen der CloudFront Distribution erfolgt, die den MediaStore Ursprung enthält. Dies ist die Distribution mit dem MediaStore Ursprung, zu der Sie OAC hinzufügen möchten.

Im Folgenden finden Sie Beispiele für MediaStore Container-Richtlinien, die es einer CloudFront Distribution ermöglichen, auf einen MediaStore Ursprung zuzugreifen.

Example MediaStore Container-Richtlinie, die schreibgeschützten Zugriff für eine CloudFront Distribution mit aktiviertem OAC ermöglicht

JSON

```
{
        "Version":"2012-10-17",
        "Statement": [
            {
                "Sid": "AllowCloudFrontServicePrincipalReadOnly",
                "Effect": "Allow",
                "Principal": {
                  "Service": "cloudfront.amazonaws.com"
                },
                "Action": [
                  "mediastore:GetObject"
                ],
                "Resource": "arn:aws:mediastore:us-
east-1:111122223333:container/<container name>/*",
                "Condition": {
                    "StringEquals": {
                      "AWS:SourceArn":
 "arn:aws:cloudfront::111122223333:distribution/CloudFront-distribution-ID"
                    },
                    "Bool": {
                      "aws:SecureTransport": "true"
                    }
                }
```

```
            }
        ]
}
```

Example MediaStore Container-Richtlinie, die Lese- und Schreibzugriff für eine CloudFront Distribution mit aktiviertem OAC ermöglicht

JSON

```
{
        "Version":"2012-10-17",
        "Statement": [
            {
                "Sid": "AllowCloudFrontServicePrincipalReadWrite",
                "Effect": "Allow",
                "Principal": {
                  "Service": "cloudfront.amazonaws.com"
                },
                "Action": [
                  "mediastore:GetObject",
                  "mediastore:PutObject"
                ],
                "Resource": "arn:aws:mediastore:us-
east-1:111122223333:container/container-name/*",
                "Condition": {
                    "StringEquals": {
                      "AWS:SourceArn":
 "arn:aws:cloudfront::111122223333:distribution/CloudFront-distribution-ID"
                    },
                    "Bool": {
                      "aws:SecureTransport": "true"
                    }
                }
            }
        ]
}
```

> **Note**
>
> Um Schreibzugriff zu ermöglichen, müssen Sie die zulässigen HTTP-Methoden so konfigurieren, dass sie PUT in die Verhaltenseinstellungen Ihrer CloudFront Distribution aufgenommen werden.

Erstellen der Ursprungszugriffssteuerung

Um ein OAC zu erstellen, können Sie die AWS-Managementkonsole, CloudFormation AWS CLI, oder die CloudFront API verwenden.

Console

So erstellen Sie eine Ursprungszugriffssteuerung

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home.
2. Wählen Sie im Navigationsbereich Origin access (Ursprungszugriff) aus.
3. Wählen Sie Create control setting (Kontrolleinstellung erstellen) aus.
4. Gehen Sie im Formular Create control setting (Kontrolleinstellung erstellen) wie folgt vor:
   a. Geben Sie im Bereich Details einen Namen (Name) und (optional) eine Beschreibung (Description) für die Ursprungszugriffssteuerung ein.
   b. Es empfiehlt sich, im Bereich Settings (Einstellungen) die Standardeinstellung (Sign requests (recommended)) (Anforderungen signieren (empfohlen)) zu belassen. Weitere Informationen finden Sie unter the section called “Erweiterte Einstellungen für die Ursprungszugriffssteuerung”.
5. Wählen Sie MediaStore aus der Drop-down-Liste „Origin-Typ“ aus.
6. Wählen Sie Erstellen aus.

   Nachdem die OAC erstellt wurde, notieren Sie sich den Namen. Sie benötigen diesen im folgenden Verfahren.

Um einem Ursprung in einer Distribution eine MediaStore Ursprungszugriffskontrolle hinzuzufügen

1. Öffnen Sie die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home.

2. Wählen Sie eine Distribution mit einem MediaStore Ursprung aus, zu dem Sie das OAC hinzufügen möchten, und wählen Sie dann den Tab Origins.
3. Wählen Sie den MediaStore Ursprung aus, dem Sie das OAC hinzufügen möchten, und klicken Sie dann auf Bearbeiten.
4. Wählen Sie HTTPS only (Nur HTTPS) für Protocol (Protokoll) Ihres Ursprungs aus.
5. Wählen Sie im Dropdown-Menü Origin access control (Ursprungszugriffssteuerung) die OAC aus, die Sie verwenden möchten.
6. Wählen Sie Änderungen speichern aus.

Die Verteilung beginnt mit der Bereitstellung an allen CloudFront Edge-Standorten. Wenn ein Edge-Standort die neue Konfiguration empfängt, signiert er alle Anfragen, die er an den MediaStore Bucket-Ursprung sendet.

CloudFormation

Verwenden Sie den `AWS::CloudFront::OriginAccessControl` Ressourcentyp CloudFormation, um eine Origin-Zugriffskontrolle (OAC) zu erstellen. Das folgende Beispiel zeigt die CloudFormation Vorlagensyntax im YAML-Format für die Erstellung einer Origin-Zugriffskontrolle.

```
Type: AWS::CloudFront::OriginAccessControl
Properties:
  OriginAccessControlConfig:
      Description: An optional description for the origin access control
      Name: ExampleOAC
      OriginAccessControlOriginType: mediastore
      SigningBehavior: always
      SigningProtocol: sigv4
```

Weitere Informationen finden Sie unter AWS::CloudFront:: OriginAccessControl im AWS CloudFormation Benutzerhandbuch.

CLI

Verwenden Sie den aws cloudfront create-origin-access-control Befehl, um eine Origin-Zugriffskontrolle mit dem AWS Command Line Interface (AWS CLI) zu erstellen. Sie können eine Eingabedatei verwenden, um die Eingabeparameter für den Befehl bereitzustellen, anstatt jeden einzelnen Parameter als Befehlszeileneingabe anzugeben.

So erstellen Sie eine Ursprungszugriffssteuerung (CLI mit Eingabedatei)

1. Verwenden Sie den folgenden Befehl zum Erstellen einer Datei mit dem Namen `origin-access-control.yaml`. Diese Datei enthält alle Eingabeparameter für den Befehl create-origin-access-control.

   ```
   aws cloudfront create-origin-access-control --generate-cli-skeleton yaml-input >
    origin-access-control.yaml
   ```

2. Öffnen Sie die Datei `origin-access-control.yaml`, die Sie gerade erstellt haben. Bearbeiten Sie die Datei, um einen Namen für die OAC und eine Beschreibung (optional) hinzuzufügen, und ändern Sie `SigningBehavior` zu `always`. Speichern Sie dann die Datei.

   Weitere Informationen zu anderen OAC-Einstellungen finden Sie unter the section called “Erweiterte Einstellungen für die Ursprungszugriffssteuerung”.

3. Verwenden Sie den folgenden Befehl, um die Ursprungszugriffssteuerung mit Eingabeparametern aus der Datei `origin-access-control.yaml` zu erstellen.

   ```
   aws cloudfront create-origin-access-control --cli-input-yaml file://origin-
   access-control.yaml
   ```

   Notieren Sie den `Id`-Wert in der Befehlsausgabe. Sie benötigen ihn, um das OAC zu einem MediaStore Ursprung in einer CloudFront Distribution hinzuzufügen.

Um ein OAC an einen MediaStore Ursprung in einer vorhandenen Distribution anzuhängen (CLI mit Eingabedatei)

1. Verwenden Sie den folgenden Befehl, um die Verteilungskonfiguration für die CloudFront Distribution zu speichern, zu der Sie das OAC hinzufügen möchten. Die Verteilung muss einen MediaStore Ursprung haben.

   ```
   aws cloudfront get-distribution-config --id <CloudFront distribution ID> --
   output yaml > dist-config.yaml
   ```

2. Öffnen Sie die Datei mit dem Namen `dist-config.yaml`, die Sie gerade erstellt haben. Bearbeiten Sie die Datei und nehmen Sie die folgenden Änderungen vor:

   - Fügen Sie im Objekt `Origins` die ID der OAC dem Feld namens `OriginAccessControlId` hinzu.
   - Entfernen Sie den Wert aus dem Feld namens `OriginAccessIdentity`, sofern vorhanden.
   - Benennen Sie das Feld `ETag` in `IfMatch` um, ändern Sie jedoch nicht den Wert des Feldes.

   Speichern Sie die Datei, wenn Sie fertig sind.

3. Verwenden Sie den folgenden Befehl, um die Distribution zu aktualisieren und die Ursprungszugriffssteuerung zu verwenden.

```
aws cloudfront update-distribution --id <CloudFront distribution ID> --cli-
input-yaml file://dist-config.yaml
```

Die Verteilung beginnt mit der Bereitstellung an allen CloudFront Edge-Standorten. Wenn ein Edge-Standort die neue Konfiguration empfängt, signiert er alle Anfragen, die er an den MediaStore Ursprung sendet.

API

Um mit der CloudFront API eine Origin-Zugriffskontrolle zu erstellen, verwenden Sie CreateOriginAccessControl. Weitere Informationen zu den Feldern, die Sie in diesem API-Aufruf angeben, finden Sie in der API-Referenzdokumentation für Ihr AWS SDK oder einen anderen API-Client.

Nachdem Sie eine Origin-Zugriffskontrolle erstellt haben, können Sie sie mithilfe eines der folgenden API-Aufrufe an einen MediaStore Ursprung in einer Distribution anhängen:

- Um sie an eine bestehende Distribution anzuhängen, verwenden Sie UpdateDistribution.
- Um es an eine neue Distribution anzuhängen, verwenden Sie CreateDistribution.

Geben Sie für beide API-Aufrufe die ID der Ursprungszugriffssteuerung im Feld `OriginAccessControlId` innerhalb eines Ursprungs an. Weitere Informationen zu den

anderen Feldern, die Sie in diesen API-Aufrufen angeben, finden Sie unter Referenz für alle Distributionseinstellungen und in der API-Referenzdokumentation für Ihr AWS SDK oder einen anderen API-Client.

Erweiterte Einstellungen für die Ursprungszugriffssteuerung

Die CloudFront Origin-Zugriffskontrollfunktion umfasst erweiterte Einstellungen, die nur für bestimmte Anwendungsfälle vorgesehen sind. Verwenden Sie die empfohlenen Einstellungen, sofern Sie die erweiterten Einstellungen nicht speziell benötigen.

Die Origin-Zugriffskontrolle enthält eine Einstellung namens Signaturverhalten (in der Konsole) oder `SigningBehavior` (in der API, CLI und CloudFormation). Diese Einstellung bietet die folgenden Optionen:

Ursprungsanforderungen immer signieren (empfohlene Einstellung)

Wir empfehlen die Verwendung dieser Einstellung mit der Bezeichnung Sign requests (recommended) (Anforderungen signieren (empfohlen)) in der Konsole bzw. `always` in der API, CLI und CloudFormation. Mit dieser Einstellung signiert es CloudFront immer alle Anfragen, die es an den MediaStore Ursprung sendet.

Ursprungsanforderungen nie signieren

Diese Einstellung heißt Do not sign requests (Anforderungen nicht signieren) in der Konsole bzw. `never` in der API, CLI und CloudFormation. Verwenden Sie diese Einstellung, um die Ursprungszugriffssteuerung für alle Ursprünge in allen Distributionen zu deaktivieren, die diese Ursprungszugriffssteuerung verwenden. Auf diese Weise lässt sich im Vergleich zum einzelnen Entfernen einer Ursprungszugriffssteuerung aus allen Ursprüngen und Distributionen, die diese verwenden, Zeit und Aufwand sparen. Mit dieser Einstellung signiert es CloudFront keine Anfragen, die es an den MediaStore Ursprung sendet.

> **Warning**
>
> Um diese Einstellung verwenden zu können, muss der MediaStore Ursprung öffentlich zugänglich sein. Wenn Sie diese Einstellung mit einem MediaStore Ursprung verwenden, der nicht öffentlich zugänglich ist, CloudFront können Sie nicht auf den Ursprung zugreifen. Der MediaStore Ursprung gibt Fehler zurück CloudFront und CloudFront leitet diese Fehler an die Zuschauer weiter. Weitere Informationen finden Sie in der MediaStore Beispiel-Container-Richtlinie für öffentlichen Lesezugriff über HTTPS.

Viewer (Client)-**`Authorization`**-Header nicht überschreiben

Diese Einstellung heißt Do not override authorization header (Autorisierungsheader nicht überschreiben) in der Konsole bzw. `no-override` in der API, CLI und CloudFormation. Verwenden Sie diese Einstellung, wenn Sie ursprüngliche Anfragen nur signieren CloudFront möchten, wenn die entsprechende Viewer-Anfrage keinen `Authorization` Header enthält. Mit dieser Einstellung wird der `Authorization` Header der Viewer-Anfrage weitergegeben, CloudFront wenn eine vorhanden ist, signiert aber die ursprüngliche Anfrage (fügt einen eigenen `Authorization` Header hinzu), wenn die Viewer-Anfrage keinen `Authorization` Header enthält.

> **Warning**
>
> Um den `Authorization` Header aus der Viewer-Anfrage weiterzugeben, müssen Sie den `Authorization` Header zu einer Cache-Richtlinie für alle Cache-Verhaltensweisen hinzufügen, die MediaStore Ursprünge verwenden, die mit dieser ursprünglichen Zugriffskontrolle verknüpft sind.

Beschränken Sie den Zugriff auf ein AWS Lambda Herkunft der Funktions-URL

CloudFront bietet Origin Access Control (OAC), um den Zugriff auf den URL-Ursprung einer Lambda-Funktion einzuschränken.

> **Note**
>
> Wenn Sie eine Lambda-Webfunktion als Ausgangspunkt verwenden, finden Sie CloudFront weitere Informationen unterBeschränken Sie den Zugriff auf ein AWS Lambda Ursprung der Webfunktion. Für die Ursprünge der Lambda-Webfunktionen ist eine andere OAC-Konfiguration erforderlich.

Themen

- Erstellen einer neuen OAC
- Erweiterte Einstellungen für die Ursprungszugriffssteuerung
- Beispielvorlagencode

Erstellen einer neuen OAC

Führen Sie die in den folgenden Themen beschriebenen Schritte aus, um ein neues OAC in einzurichten. CloudFront

> **Important**
>
> Wenn Sie mit Ihrer Lambda-Funktions-URL `POST` Methoden verwenden`PUT`, müssen Ihre Benutzer den SHA256-Wert des Hauptteils berechnen und den Payload-Hashwert des Anforderungstexts in den `x-amz-content-sha256` Header aufnehmen, wenn sie die Anfrage an senden. CloudFront Lambda unterstützt keine unsignierten Nutzdaten.

Themen

- Voraussetzungen
- Erteilen Sie die CloudFront Erlaubnis, auf die URL der Lambda-Funktion zuzugreifen
- Erstellen der OAC

Voraussetzungen

Bevor Sie OAC erstellen und einrichten, benötigen Sie eine CloudFront Distribution mit einer Lambda-Funktions-URL als Ursprung. Für die Verwendung von OAC müssen Sie `AWS_IAM` als Wert für den `AuthType`-Parameter angeben. Weitere Informationen finden Sie unter Verwenden einer Lambda-Funktions-URL.

Erteilen Sie die CloudFront Erlaubnis, auf die URL der Lambda-Funktion zuzugreifen

Bevor Sie ein OAC erstellen oder es in einer CloudFront Distribution einrichten, stellen Sie sicher, dass es über die Zugriffsberechtigung für die Lambda-Funktions-URL CloudFront verfügt. Tun Sie dies, nachdem Sie eine CloudFront Distribution erstellt haben, aber bevor Sie das OAC zur Lambda-Funktions-URL in der Verteilungskonfiguration hinzufügen.

> **Note**
>
> Um die IAM-Richtlinie für die Lambda-Funktions-URL zu aktualisieren, müssen Sie die AWS Command Line Interface (AWS CLI) verwenden. Die Bearbeitung der IAM-Richtlinie in der Lambda-Konsole wird derzeit nicht unterstützt.

Der folgende AWS CLI Befehl gewährt dem CloudFront Service Principal (`cloudfront.amazonaws.com`) Zugriff auf Ihre Lambda-Funktions-URL. Das `Condition` Element in der Richtlinie ermöglicht den CloudFront Zugriff auf Lambda nur, wenn die Anfrage im Namen der CloudFront Distribution erfolgt, die die URL der Lambda-Funktion enthält. Dies ist die Distribution mit der Lambda-Funktions-URL als Ursprung, der Sie OAC hinzufügen möchten.

Example: AWS CLI Befehl zum Aktualisieren einer Richtlinie, um für eine Distribution mit aktiviertem OAC nur Lesezugriff zu CloudFront gewähren

Die folgenden AWS CLI Befehle ermöglichen der CloudFront Distribution (*E1PDK09ESKHJWT*) den Zugriff auf Ihr Lambda*FUNCTION_URL_NAME*.

```
aws lambda add-permission \
--statement-id "AllowCloudFrontServicePrincipal" \
--action "lambda:InvokeFunctionUrl" \
--principal "cloudfront.amazonaws.com" \
--source-arn "arn:aws:cloudfront::123456789012:distribution/E1PDK09ESKHJWT" \
--function-name FUNCTION_URL_NAME
```

```
aws lambda add-permission \
--statement-id "AllowCloudFrontServicePrincipalInvokeFunction" \
--action "lambda:InvokeFunction" \
--principal "cloudfront.amazonaws.com" \
--source-arn "arn:aws:cloudfront::123456789012:distribution/E1PDK09ESKHJWT" \
--function-name FUNCTION_URL_NAME
```

> **Note**
>
> Wenn Sie eine Distribution erstellen und diese keine Berechtigung für Ihre Lambda-Funktions-URL hat, können Sie in der CloudFront Konsole den Befehl CLI kopieren auswählen und diesen Befehl dann über Ihr Befehlszeilenterminal eingeben. Weitere Informationen finden Sie unter Gewähren des Zugriffs auf Funktionen für AWS-Services im Entwicklerhandbuch für AWS Lambda .

Erstellen der OAC

Um ein OAC zu erstellen, können Sie die AWS-Managementkonsole, CloudFormation AWS CLI, oder die CloudFront API verwenden.

Console

So erstellen Sie eine OAC

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home.
2. Wählen Sie im Navigationsbereich Origin access (Ursprungszugriff) aus.
3. Wählen Sie Create control setting (Kontrolleinstellung erstellen) aus.
4. Führen Sie im Formular Neue OAC erstellen die folgenden Schritte aus:
   a. Geben Sie für die OAC einen Namen und (optional) eine Beschreibung ein.
   b. Es empfiehlt sich, unter Signierverhalten die Standardeinstellung Anforderungen signieren (empfohlen) zu belassen. Weitere Informationen finden Sie unter the section called “Erweiterte Einstellungen für die Ursprungszugriffssteuerung”.
5. Wählen Sie als Ursprungstyp die Option Lambda aus.
6. Wählen Sie Erstellen aus.

> **Tip**
>
> Nachdem Sie die OAC erstellt haben, notieren Sie sich den Namen. Sie benötigen diesen im folgenden Verfahren.

So fügen Sie eine Ursprungszugriffssteuerung einer Lambda-Funktions-URL in einer Distribution hinzu

1. Öffnen Sie die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home.
2. Wählen Sie eine Distribution mit einer Lambda-Funktions-URL, der Sie die OAC hinzufügen möchten, und dann die Registerkarte Ursprünge aus.
3. Wählen Sie die Lambda-Funktions-URL, der Sie die OAC hinzufügen möchten, und dann Bearbeiten aus.
4. Wählen Sie HTTPS only (Nur HTTPS) für Protocol (Protokoll) Ihres Ursprungs aus.
5. Wählen Sie im Dropdown-Menü Ursprungszugriffssteuerung den zu verwendenden OAC-Namen aus.
6. Wählen Sie Änderungen speichern aus.

Die Verteilung beginnt mit der Bereitstellung an allen CloudFront Edge-Standorten. Wenn ein Edge-Standort die neue Konfiguration erhält, signiert er alle Anforderungen, die er an die Lambda-Funktions-URL sendet.

CloudFormation

Verwenden Sie den `AWS::CloudFront::OriginAccessControl` Ressourcentyp CloudFormation, um ein OAC mit zu erstellen. Das folgende Beispiel zeigt die CloudFormation Vorlagensyntax im YAML-Format für die Erstellung eines OAC.

```
Type: AWS::CloudFront::OriginAccessControl
Properties:
  OriginAccessControlConfig:
      Description: An optional description for the origin access control
      Name: ExampleOAC
      OriginAccessControlOriginType: lambda
      SigningBehavior: always
      SigningProtocol: sigv4
```

Weitere Informationen finden Sie unter AWS::CloudFront:: OriginAccessControl im AWS CloudFormation Benutzerhandbuch.

CLI

Verwenden Sie den aws cloudfront create-origin-access-control Befehl, um eine Origin-Zugriffskontrolle mit dem AWS Command Line Interface (AWS CLI) zu erstellen. Sie können eine Eingabedatei verwenden, um die Eingabeparameter für den Befehl bereitzustellen, anstatt jeden einzelnen Parameter als Befehlszeileneingabe anzugeben.

So erstellen Sie eine Ursprungszugriffssteuerung (CLI mit Eingabedatei)

1. Verwenden Sie den folgenden Befehl zum Erstellen einer Datei mit dem Namen `origin-access-control.yaml`. Diese Datei enthält alle Eingabeparameter für den Befehl create-origin-access-control.

   ```
   aws cloudfront create-origin-access-control --generate-cli-skeleton yaml-input >
    origin-access-control.yaml
   ```

2. Öffnen Sie die Datei `origin-access-control.yaml`, die Sie gerade erstellt haben. Bearbeiten Sie die Datei, um einen Namen für die OAC und eine Beschreibung (optional)

hinzuzufügen, und ändern Sie `SigningBehavior` zu `always`. Speichern Sie dann die Datei.

Weitere Informationen zu anderen OAC-Einstellungen finden Sie unter the section called “Erweiterte Einstellungen für die Ursprungszugriffssteuerung”.

3. Verwenden Sie den folgenden Befehl, um die Ursprungszugriffssteuerung mit Eingabeparametern aus der Datei `origin-access-control.yaml` zu erstellen.

```
aws cloudfront create-origin-access-control --cli-input-yaml file://origin-
access-control.yaml
```

Notieren Sie den `Id`-Wert in der Befehlsausgabe. Sie benötigen es, um das OAC zu einer Lambda-Funktions-URL in einer CloudFront Distribution hinzuzufügen.

So fügen Sie eine OAC einer Lambda-Funktions-URL in einer vorhandenen Distribution an (CLI mit Eingabedatei)

1. Verwenden Sie den folgenden Befehl, um die Verteilungskonfiguration für die CloudFront Distribution zu speichern, zu der Sie das OAC hinzufügen möchten. Die Distribution muss eine Lambda-Funktions-URL als Ursprung haben.

```
aws cloudfront get-distribution-config --id <CloudFront distribution ID> --
output yaml > dist-config.yaml
```

2. Öffnen Sie die Datei mit dem Namen `dist-config.yaml`, die Sie gerade erstellt haben. Bearbeiten Sie die Datei und nehmen Sie die folgenden Änderungen vor:

   - Fügen Sie im Objekt `Origins` die ID der OAC dem Feld namens `OriginAccessControlId` hinzu.
   - Entfernen Sie den Wert aus dem Feld namens `OriginAccessIdentity`, sofern vorhanden.
   - Benennen Sie das Feld `ETag` in `IfMatch` um, ändern Sie jedoch nicht den Wert des Feldes.

   Speichern Sie die Datei, wenn Sie fertig sind.

3. Verwenden Sie den folgenden Befehl, um die Distribution zu aktualisieren und die Ursprungszugriffssteuerung zu verwenden.

```
aws cloudfront update-distribution --id <CloudFront distribution ID> --cli-
input-yaml file://dist-config.yaml
```

Die Verteilung beginnt mit der Bereitstellung an allen CloudFront Edge-Standorten. Wenn ein Edge-Standort die neue Konfiguration erhält, signiert er alle Anforderungen, die er an die Lambda-Funktions-URL sendet.

API

Um ein OAC mit der CloudFront API zu erstellen, verwenden Sie CreateOriginAccessControl. Weitere Informationen zu den Feldern, die Sie in diesem API-Aufruf angeben, finden Sie in der API-Referenzdokumentation für Ihr AWS SDK oder einen anderen API-Client.

Nachdem Sie eine OAC erstellt haben, können Sie diese mit einem der folgenden API-Aufrufe an eine Lambda-Funktions-URL in einer Distribution anfügen:

- Um es an eine bestehende Distribution anzuhängen, verwenden Sie UpdateDistribution.
- Um es an eine neue Distribution anzuhängen, verwenden Sie CreateDistribution.

Geben Sie für beide API-Aufrufe die ID der OAC im Feld `OriginAccessControlId` innerhalb eines Ursprungs an. Weitere Informationen zu den anderen Feldern, die Sie in diesen API-Aufrufen angeben, finden Sie unter und in der API-Referenzdokumentation für Ihr AWS SDK oder einen anderen API-Client.

Erweiterte Einstellungen für die Ursprungszugriffssteuerung

Die CloudFront OAC-Funktion umfasst erweiterte Einstellungen, die nur für bestimmte Anwendungsfälle vorgesehen sind. Verwenden Sie die empfohlenen Einstellungen, sofern Sie die erweiterten Einstellungen nicht speziell benötigen.

OAC enthält eine Einstellung mit dem Namen Signaturverhalten (in der Konsole) oder `SigningBehavior` (in der API, CLI und CloudFormation). Diese Einstellung bietet die folgenden Optionen:

Ursprungsanforderungen immer signieren (empfohlene Einstellung)

Wir empfehlen die Verwendung dieser Einstellung mit der Bezeichnung Sign requests (recommended) (Anforderungen signieren (empfohlen)) in der Konsole bzw. `always` in der API, CLI und CloudFormation. Mit dieser Einstellung signiert es CloudFront immer alle Anfragen, die es an die URL der Lambda-Funktion sendet.

Ursprungsanforderungen nie signieren

Diese Einstellung heißt Do not sign requests (Anforderungen nicht signieren) in der Konsole bzw. `never` in der API, CLI und CloudFormation. Verwenden Sie diese Einstellung, um die OAC für alle Ursprünge in allen Distributionen zu deaktivieren, die diese OAC verwenden. Auf diese Weise lässt sich im Vergleich zum einzelnen Entfernen einer OAC aus allen Ursprüngen und Distributionen, die diese verwenden, Zeit und Aufwand sparen. Signiert mit dieser Einstellung CloudFront keine Anfragen, die an die URL der Lambda-Funktion gesendet werden.

> **Warning**
>
> Um diese Einstellung verwenden zu können, muss die Lambda-Funktions-URL öffentlich zugänglich sein. Wenn Sie diese Einstellung mit einer Lambda-Funktions-URL verwenden, die nicht öffentlich zugänglich ist, CloudFront können Sie nicht auf den Ursprung zugreifen. Die URL der Lambda-Funktion gibt Fehler zurück CloudFront und CloudFront leitet diese Fehler an die Betrachter weiter. Weitere Informationen finden Sie unter Sicherheits- und Authentifizierungsmodell für Lambda-Funktions-URLs im Benutzerhandbuch für AWS Lambda .

Viewer (Client)-**`Authorization`**-Header nicht überschreiben

Diese Einstellung heißt Do not override authorization header (Autorisierungsheader nicht überschreiben) in der Konsole bzw. `no-override` in der API, CLI und CloudFormation. Verwenden Sie diese Einstellung, wenn Sie ursprüngliche Anfragen nur signieren CloudFront möchten, wenn die entsprechende Viewer-Anfrage keinen `Authorization` Header enthält. Mit dieser Einstellung wird der `Authorization` Header der Viewer-Anfrage weitergegeben, CloudFront wenn eine vorhanden ist, signiert aber die ursprüngliche Anfrage (fügt einen eigenen `Authorization` Header hinzu), wenn die Viewer-Anfrage keinen `Authorization` Header enthält.

⚠ Warning

- Wenn Sie diese Einstellung verwenden, müssen Sie die Signature Version 4-Signatur für die Lambda-Funktions-URL anstelle des Namens oder des CNAME Ihrer CloudFront Distribution angeben. Wenn der `Authorization` Header von der Viewer-Anfrage an die URL der Lambda-Funktion weitergeleitet wird, validiert Lambda die Signatur anhand des Hosts der CloudFront Lambda-URL-Domain. Wenn die Signatur nicht auf der Lambda-URL-Domain basiert, stimmt der Host in der Signatur nicht mit dem Host überein, der vom Lambda-URL-Ursprung verwendet wird. Dies hat zur Folge, dass die Anforderung fehlschlägt, was zu einem Fehler bei der Signaturvalidierung führt.
- Um den `Authorization`-Header aus der Viewer-Anforderung zu übergeben, müssen Sie den `Authorization`-Header einer Cache-Richtlinie für alle Cache-Verhaltensweisen hinzufügen, die Lambda-Funktions-URLs verwenden, die dieser Ursprungszugriffssteuerung zugeordnet sind.

Beispielvorlagencode

Wenn Ihr CloudFront Ursprung eine Lambda-Funktions-URL ist, die mit einem OAC verknüpft ist, können Sie das folgende Python-Skript verwenden, um Dateien mit der Methode in die Lambda-Funktion hochzuladen. `POST`

Dieser Code setzt voraus, dass Sie die OAC so konfiguriert haben, dass das standardmäßige Signaturverhalten auf Ursprungsanforderungen immer signieren eingestellt ist und dass Sie die Einstellung Autorisierungsheader nicht überschreiben nicht ausgewählt haben.

Diese Konfiguration ermöglicht es der OAC, die SigV4-Autorisierung mit Lambda mithilfe des Lambda-Hostnamens korrekt zu verwalten. Die Nutzdaten werden mithilfe von SigV4 aus der IAM-Identität signiert, die für die Lambda-Funktions-URL autorisiert ist, die als `IAM_AUTH`-Typ festgelegt ist.

Die Vorlage zeigt, wie signierte Nutzdaten-Hashwerte im x-amz-content-sha256-Header für clientseitige `POST`-Anforderungen behandelt werden. Diese Vorlage wurde speziell für die Verwaltung von Formulardaten-Nutzdaten entwickelt. Die Vorlage ermöglicht sichere Datei-Uploads an eine Lambda-Funktions-URL und verwendet AWS Authentifizierungsmechanismen CloudFront, um sicherzustellen, dass nur autorisierte Anfragen auf die Lambda-Funktion zugreifen können.

Der Code umfasst folgende Funktionen:

- erfüllt die Anforderung, den Nutzdaten-Hash in den x-amz-content-sha256-Header aufzunehmen
- Verwendet die SigV4-Authentifizierung für den sicheren Zugriff AWS-Service
- unterstützt Datei-Uploads mithilfe von mehrteiligen Formulardaten
- beinhaltet die Fehlerbehandlung für Anforderungsausnahmen

```
import boto3
from botocore.auth import SigV4Auth
from botocore.awsrequest import AWSRequest
import requests
import hashlib
import os


def calculate_body_hash(body):
    return hashlib.sha256(body).hexdigest()


def sign_request(request, credentials, region, service):
    sigv4 = SigV4Auth(credentials, service, region)
    sigv4.add_auth(request)


def upload_file_to_lambda(cloudfront_url, file_path, region):
    # AWS credentials
    session = boto3.Session()
    credentials = session.get_credentials()

    # Prepare the multipart form-data
    boundary = "------------------------boundary"

    # Read file content
    with open(file_path, 'rb') as file:
        file_content = file.read()

    # Get the filename from the path
    filename = os.path.basename(file_path)
```

```
    # Prepare the multipart body
    body = (
        f'--{boundary}\r\n'
        f'Content-Disposition: form-data; name="file"; filename="{filename}"\r\n'
        f'Content-Type: application/octet-stream\r\n\r\n'
    ).encode('utf-8')
    body += file_content
    body += f'\r\n--{boundary}--\r\n'.encode('utf-8')

    # Calculate SHA256 hash of the entire body
    body_hash = calculate_body_hash(body)

    # Prepare headers
    headers = {
        'Content-Type': f'multipart/form-data; boundary={boundary}',
        'x-amz-content-sha256': body_hash
    }

    # Create the request
    request = AWSRequest(
        method='POST',
        url=cloudfront_url,
        data=body,
        headers=headers
    )

    # Sign the request
    sign_request(request, credentials, region, 'lambda')

    # Get the signed headers
    signed_headers = dict(request.headers)

    # Print request headers before sending
    print("Request Headers:")
    for header, value in signed_headers.items():
        print(f"{header}: {value}")

    try:
        # Send POST request with signed headers
        response = requests.post(
            cloudfront_url,
            data=body,
            headers=signed_headers
```

```
        )

        # Print response status and content
        print(f"\nStatus code: {response.status_code}")
        print("Response:", response.text)

        # Print response headers
        print("\nResponse Headers:")
        for header, value in response.headers.items():
            print(f"{header}: {value}")

    except requests.exceptions.RequestException as e:
        print(f"An error occurred: {e}")


# Usage
cloudfront_url = "https://d111111abcdef8.cloudfront.net"
file_path = r"filepath"
region = "us-east-1"  # example: "us-west-2"

upload_file_to_lambda(cloudfront_url, file_path, region)
```

Beschränken Sie den Zugriff auf ein AWS Lambda Ursprung der Webfunktion

Sie können Origin Access Control (OAC) verwenden, um den Zugriff auf einen AWS Lambda Web Function-Ursprung einzuschränken. Eine Lambda-Webfunktion bietet ihren eigenen regionalen oder globalen HTTPS-Endpunkt, den Sie als Ursprungsdomain für Ihre CloudFront Distribution verwenden. Weitere Informationen zu Lambda Web Functions finden Sie unter AWS Lambda Web Functions im AWS Lambda Developer Guide.

> **Note**
>
> OAC mit Lambda-Webfunktionen wird für globale Endpunkte, die Opt-in-Regionen enthalten, nicht unterstützt. Wenn der Endpunkt eine Anfrage an eine Opt-in-Region weiterleitet, CloudFront kann die Anfrage für diese Region nicht signiert werden. Der Ursprung gibt dann einen HTTP 403-Fehler (Forbidden) zurück. Bei allen Regionen, die dem globalen Endpunkt zugeordnet sind, muss es sich um Regionen handeln, die standardmäßig aktiviert sind.

Hinweise zur Verwendung von OAC mit einem URL-Ursprung für eine Lambda-Funktion finden Sie unter. Beschränken Sie den Zugriff auf ein AWS Lambda Herkunft der Funktions-URL

Voraussetzungen

Bevor Sie OAC erstellen und einrichten, benötigen Sie eine CloudFront Distribution mit einer Lambda-Webfunktion als Ursprung. Der Ursprungsdomänenname ist der Hostname des Endpunkts der Lambda-Webfunktion. CloudFront identifiziert den Endpunkttyp anhand des Hostnamens:

- Regionaler Endpunkt (`HomeRegion`und `PerRegion` Endpunkttypen) — Der Hostname endet auf `.lambda-web.`*`AWS-Region`*`.on.aws`
- Globaler Endpunkt (`MultiRegion`Endpunkttyp) — Der Hostname endet auf `.lambda-web.global.on.aws`

Ihre Konfiguration muss außerdem die folgenden Anforderungen erfüllen:

- Authentifizierungstyp des Endpunkts — Der Authentifizierungstyp des Endpunkts muss sein. `IamAuth` Mit verwendet der Endpunkt AWS Identity and Access Management (IAM)`IamAuth`, um Anfragen zu authentifizieren und zu autorisieren. Bei Anfragen von CloudFront basiert die Autorisierung auf der ressourcenbasierten Richtlinie der Lambda-Webfunktion. CloudFront signiert Anfragen im Namen des Betrachters, und der Endpunkt validiert die Signatur. Weitere Informationen finden Sie im AWS Lambda Developer Guide unter CloudFront Integration.
- Cache-Richtlinie — Wenn Ihre Lambda-Webfunktion den `Authorization` Header des Viewers verwendet, müssen Sie den Header zur Cache-Richtlinie hinzufügen. Diese Anforderung gilt für jedes Cache-Verhalten, das den Ursprung verwendet. Für `GET` und `HEAD` Anfragen CloudFront entfernt den `Authorization` Header, bevor die Anfrage weitergeleitet wird, es sei denn, der Header ist in der Cache-Richtlinie enthalten. Weitere Informationen finden Sie unter Header und CloudFront Verhalten von HTTP-Anfragen (benutzerdefiniert und Amazon S3-Ursprünge).
- Richtlinie für ursprüngliche Anfragen — Verwenden Sie eine Richtlinie für ursprüngliche Anfragen, die den `Host` Header des Betrachters ausschließt. Lambda Web Function-Endpunkte erwarten, dass der `Host` Header den ursprünglichen Domainnamen und nicht den Domainnamen der CloudFront Distribution enthält. Wenn der `Host` Header den Domainnamen der Distribution enthält, gibt der Endpunkt einen HTTP 404-Fehler (Not Found) zurück. Wenn Sie eine Richtlinie für verwaltete Ursprungsanfragen verwenden, verwenden Sie diese `AllViewerExceptHostHeader` Option, nicht`AllViewer`. Weitere Informationen finden Sie unter AllViewerExceptHostHeader.

> **Note**
>
> Lambda @Edge -Origin-Anfrage- und Origin-Antwort-Trigger werden bei Cache-Verhaltensweisen, die einen Lambda-Webfunktions-Ursprung mit OAC verwenden, nicht unterstützt. Dies gilt für regionale und globale Endpunkte sowie für Ursprünge, auf die direkt oder über eine Ursprungsgruppe zugegriffen wird. CloudFront lehnt eine Verteilungskonfiguration ab, die beide enthält.

Erteilen Sie die CloudFront Erlaubnis für den Zugriff auf die Lambda-Webfunktion

Fügen Sie der Lambda-Webfunktion eine ressourcenbasierte Richtlinie hinzu, die es dem CloudFront Service Principal (`cloudfront.amazonaws.com`) ermöglicht, den Endpunkt aufzurufen. Die IAM-Aktion zum Aufrufen eines Endpunkts ist. `lambda:InvokeWebFunctionEndpoint` Die Richtlinie ist an die Lambda-Webfunktion angehängt, aber jede Anweisung kann sich `Resource` auf einen bestimmten Endpunkt-ARN ARN. Verwenden Sie ein `Condition` Element, um den Zugriff auf den Endpunkt nur dann CloudFront zu ermöglichen, wenn die Anfrage im Namen der CloudFront Distribution erfolgt, die den Ursprung enthält.

Example Resource-based Richtlinie für CloudFront OAC

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowCloudFrontOACAccess",
            "Effect": "Allow",
            "Principal": {
                "Service": "cloudfront.amazonaws.com"
            },
            "Action": "lambda:InvokeWebFunctionEndpoint",
            "Resource": "arn:aws:lambda:us-east-1:111122223333:web-function/my-
function/endpoint/prod",
            "Condition": {
                "StringEquals": {
                    "aws:SourceArn":
 "arn:aws:cloudfront::111122223333:distribution/EDFDVBD6EXAMPLE"
                }
            }
        }
    ]
```

```
}
```

Um die ressourcenbasierte Richtlinie anzuhängen, verwenden Sie den put-resource-policy Befehl mit der Funktion ARN (kein Endpunkt-ARN):

```
aws lambda-web put-resource-policy \
  --resource-arn "arn:aws:lambda:us-east-1:111122223333:web-function/my-function" \
  --policy file://policy.json \
  --region us-east-1
```

Sie können die Richtlinie mit den Befehlen get-resource-policy und delete-resource-policy überprüfen oder entfernen.

> **Note**
>
> Für Endgeräte, die sich über mehr als eine Region erstrecken, repliziert Lambda Richtlinienänderungen auf alle Regionen, die den Endpunkt bedienen. Es kann eine kurze Zeit dauern, bis eine neue oder aktualisierte Richtlinie in jeder Region wirksam wird. Weitere Informationen finden Sie unter Replikation und eventuelle Konsistenz im AWS Lambda Entwicklerhandbuch.

Erstellen der Ursprungszugriffssteuerung

Um eine Origin Access Control (OAC) zu erstellen, können Sie die AWS-Managementkonsole, CloudFormation AWS CLI, oder die CloudFront API verwenden.

Console

So erstellen Sie eine Ursprungszugriffssteuerung

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home.
2. Wählen Sie im Navigationsbereich Origin access (Ursprungszugriff) aus.
3. Wählen Sie Create control setting (Kontrolleinstellung erstellen) aus.
4. Gehen Sie im Formular Create control setting (Kontrolleinstellung erstellen) wie folgt vor:

    a. Geben Sie im Bereich Details einen Namen (Name) und (optional) eine Beschreibung (Description) für die Ursprungszugriffssteuerung ein.

b. Wählen Sie Lambda aus der Dropdownliste Origin-Typ aus. Dieser Originstyp wird sowohl für Lambda-Funktions-URLs als auch für Lambda-Webfunktionen verwendet.

c. Wenn Ihr Ursprung ein globaler Endpunkt ist (der Hostname endet auf `.lambda-web.global.on.aws`), wählen Sie die Option SigV4A-Signaturprotokoll verwenden. Lassen Sie diese Option für einen regionalen Endpunkt (der Hostname endet auf `.lambda-web.`*`AWS-Region`*`.on.aws`) deaktiviert.

d. Wählen Sie unter Signierverhalten die Option Immer signieren, Zuschauerautorisierung beibehalten aus. Diese Option wird unter Anfragen signieren (empfohlen) angezeigt und ist standardmäßig ausgewählt. Die Ursprünge der Lambda-Webfunktionen unterstützen nur dieses Signaturverhalten. Weitere Informationen finden Sie unter Verhalten beim Signieren.

5. Wählen Sie Erstellen aus.

So fügen Sie einem Lambda Web Function-Ursprung in einer Distribution eine Origin-Zugriffskontrolle hinzu

1. Öffnen Sie die CloudFront Konsole unter. https://console.aws.amazon.com/cloudfront/v4/home
2. Wählen Sie eine Distribution mit einem Lambda-Webfunktions-Ursprung aus, zu dem Sie das OAC hinzufügen möchten, und wählen Sie dann die Registerkarte Origins.
3. Wählen Sie den Ursprung der Lambda-Webfunktion aus, zu dem Sie das OAC hinzufügen möchten, und wählen Sie dann Bearbeiten.
4. Wählen Sie unter Ursprungszugriff die Option Einstellungen der Ursprungszugriffssteuerung (empfohlen) aus.
5. Wählen Sie im Dropdown-Menü Origin access control (Ursprungszugriffssteuerung) die OAC aus, die Sie verwenden möchten.
6. Wählen Sie Änderungen speichern aus.

Die Verteilung beginnt mit der Bereitstellung an allen CloudFront Edge-Standorten. Wenn ein Edge-Standort die neue Konfiguration empfängt, signiert er alle Anfragen, die er an den Ursprung der Lambda-Webfunktion sendet.

CLI

Verwenden Sie den Befehl create-origin-access-control. Verwenden Sie für einen regionalen Endpunkt (der Hostname endet auf `.lambda-web.`*`AWS-Region`*`.on.aws`) das `sigv4` Signaturprotokoll:

```
aws cloudfront create-origin-access-control \
    --origin-access-control-config '{
        "Name": "my-lambda-web-oac",
        "Description": "OAC for Lambda Web Function origin",
        "SigningProtocol": "sigv4",
        "SigningBehavior": "always-amz-auth",
        "OriginAccessControlOriginType": "lambda"
    }'
```

Verwenden Sie für einen globalen Endpunkt (der Hostname endet auf `.lambda-web.global.on.aws`) das `sigv4a` Signaturprotokoll:

```
aws cloudfront create-origin-access-control \
    --origin-access-control-config '{
        "Name": "my-lambda-web-global-oac",
        "Description": "OAC for global Lambda Web Function origin",
        "SigningProtocol": "sigv4a",
        "SigningBehavior": "always-amz-auth",
        "OriginAccessControlOriginType": "lambda"
    }'
```

CloudFormation

Geben Sie die folgenden Werte in der `OriginAccessControlConfig` ein:

- `SigningProtocol`: `sigv4` (regionaler Endpunkt) oder `sigv4a` (globaler Endpunkt)
- `SigningBehavior`: `always-amz-auth`
- `OriginAccessControlOriginType`: `lambda`

Example CloudFormation Vorlage

```
Type: AWS::CloudFront::OriginAccessControl
Properties:
  OriginAccessControlConfig:
```

```
    Description: An optional description for the origin access control
    Name: my-lambda-web-oac
    OriginAccessControlOriginType: lambda
    SigningBehavior: always-amz-auth
    SigningProtocol: sigv4a
```

Weitere Informationen finden Sie unter AWS::CloudFront:: OriginAccessControl im AWS CloudFormation Benutzerhandbuch.

OAC wird für jede Anfrage erst wirksam, wenn die Bereitstellung der Distribution abgeschlossen ist. Bei einem Endpunkt, der sich über mehr als eine Region erstreckt, muss die zuvor angefügte ressourcenbasierte Richtlinie auch die Replikation in alle Regionen abschließen, die den Endpunkt bedienen. Bis beide abgeschlossen sind, können einige Anfragen eine vorübergehende HTTP-403-Antwort vom Ursprung erhalten. Diese Fehler werden automatisch behoben, nachdem die Bereitstellung der Distribution und die Replikation der Richtlinie abgeschlossen ist.

Verhalten beim Signieren

Die Ursprünge der Lambda-Webfunktionen unterstützen nur das `always-amz-auth` Signierungsverhalten (Immer signieren, Zuschauerautorisierung in der Konsole beibehalten). CloudFront signiert immer alle Anfragen, die es an die Lambda-Webfunktion sendet, ersetzt aber nicht den `Authorization` Header der Viewer-Anfrage. CloudFront Fügt stattdessen die Signatur in einem speziellen CloudFront Header hinzu und übergibt den `Authorization` Header des Viewers unverändert an den Ursprung. Mit diesem Verhalten kann Ihre Lambda-Webfunktion den ursprünglichen `Authorization` Header des Viewers empfangen (z. B. um eine eigene Authentifizierung oder Autorisierung durchzuführen). CloudFront signiert immer noch jede Anfrage, die sie an den Ursprung sendet. Um den `Authorization` Header des Viewers weiterzuleiten, müssen Sie ihn einer Cache-Richtlinie für jedes Cache-Verhalten hinzufügen, das den Ursprung verwendet.

Die anderen Signierverhalten (`always`, `never`, und`no-override`) werden für Lambda-Webfunktions-Ursprünge nicht unterstützt. Das `always-amz-auth` Signaturverhalten wird für keinen anderen Quelltyp unterstützt, auch nicht für URL-Ursprünge von Lambda-Funktionen.

Signiert für `PUT`, `POST`, und `PATCH` -Anfragen die CloudFront ursprüngliche Anfrage mit einer unsignierten Payload (). `UNSIGNED-PAYLOAD` Der Hauptteil der Anfrage ist nicht durch die CloudFront Signatur abgedeckt. Die Signatur deckt weiterhin die Anforderungsmethode, den Pfad, die Abfragezeichenfolge und die Header ab, die Ihre Distribution identifizieren. Der Endpunkt kann

daher überprüfen, ob jede Anfrage CloudFront im Namen Ihrer Distribution stammt, bevor er Ihre Lambda-Webfunktion aufruft. Der Anfragetext ist während der Übertragung durch die HTTPS-Verbindung zwischen CloudFront und dem Endpunkt geschützt. Sie müssen in Anfragen an keinen `x-amz-content-sha256` Header angeben CloudFront. Wenn ein Betrachter einen sendet, verwendet er ihn CloudFront nicht beim Signieren der ursprünglichen Anfrage.

Beschränken des Zugriffs auf einen Amazon-S3-Ursprung

CloudFront bietet zwei Möglichkeiten, authentifizierte Anfragen an einen Amazon S3 S3-Ursprung zu senden: Origin Access Control (OAC) und Origin Access Identity (OAI). OAC hilft Ihnen, Ihre Ursprünge wie Amazon S3 zu sichern.

Wir empfehlen die Verwendung von OAC aufgrund der Unterstützung folgender Funktionen:

- Alle Amazon S3 S3-Buckets insgesamt AWS-Regionen, einschließlich der Opt-in-Regionen, die nach Dezember 2022 eingeführt wurden
- Serverseitige Amazon S3 S3-Verschlüsselung mit AWS KMS () SSE-KMS
- Dynamische Anforderungen (`PUT` und `DELETE`) an Amazon S3

OAI unterstützt diese Funktionen nicht oder erfordert in diesen Szenarien zusätzliche Problemumgehungen. Wenn Sie OAI bereits verwenden und zu OAC migrieren möchten, finden Sie weitere Informationen unter the section called “Migration von der Ursprungszugriffsidentität (OAI) zur Ursprungszugriffssteuerung (OAC)”.

Hinweise

- Wenn Sie CloudFront OAC mit Amazon S3-Bucket-Ursprüngen verwenden, müssen Sie Amazon S3 Object Ownership auf Bucket owner enforced setzen, die Standardeinstellung für neue Amazon S3 S3-Buckets. Wenn Sie ACLs benötigen, verwenden Sie die bevorzugte Einstellung des Bucket-Besitzers, um die Kontrolle über Objekte zu behalten, die mithilfe von CloudFront
- Wenn Ihr Ursprung ein Amazon S3 S3-Bucket ist, der als Website-Endpunkt konfiguriert ist, müssen Sie ihn CloudFront als benutzerdefinierten Ursprung einrichten. Das bedeutet, dass Sie OAC (oder OAI) nicht verwenden können. OAC unterstützt keine Ursprungsweiterleitung mit Lambda@Edge.
- Wenn Sie einen Amazon S3 Multi-Region Access Point als Ausgangspunkt verwenden, finden Sie CloudFront weitere Informationen unterthe section called “Beschränken Sie den

Zugriff auf einen Amazon S3 Multi-Region Access Point-Ursprung". Für S3 Multi-Region Access Points ist eine andere OAC-Konfiguration erforderlich.

In den folgenden Themen wird die Verwendung von OAC mit einem Amazon-S3-Ursprung beschrieben.

Topics

- the section called "Erstellen einer neuen Ursprungszugriffssteuerung"
- the section called "Löschen einer Distribution mit einer OAC, die an einen S3-Bucket angehängt ist"
- the section called "Migration von der Ursprungszugriffsidentität (OAI) zur Ursprungszugriffssteuerung (OAC)"
- the section called "Erweiterte Einstellungen für die Ursprungszugriffssteuerung"

Erstellen einer neuen Ursprungszugriffssteuerung

Führen Sie die in den folgenden Themen beschriebenen Schritte aus, um eine neue Origin-Zugriffskontrolle in CloudFront einzurichten.

Themen

- Voraussetzungen
- Erteilen Sie die CloudFront Erlaubnis, auf den S3-Bucket zuzugreifen
- Erstellen der Ursprungszugriffssteuerung

Voraussetzungen

Bevor Sie Origin Access Control (OAC) erstellen und einrichten, benötigen Sie eine CloudFront Distribution mit einem Amazon S3 S3-Bucket-Ursprung. Dieser Ursprung muss ein regulärer S3-Bucket sein, kein Bucket, der als Website-Endpunkt konfiguriert wurde. Weitere Informationen zum Einrichten einer CloudFront Distribution mit einem S3-Bucket-Ursprung finden Sie unterthe section called "Erste Schritte mit einer Standarddistribution".

Important

Wenn Sie OAC verwenden, um Ihren Amazon S3-Ursprung zu sichern, erfolgt die Kommunikation zwischen Amazon S3 CloudFront und Amazon S3 immer über HTTPS,

aber nur, wenn Sie sich dafür entscheiden, Anfragen immer zu signieren. Sie müssen in der Konsole die Option Anfragen signieren (empfohlen) auswählen oder `always` in der CloudFront API angeben, AWS CLI, oder CloudFormation.
Wenn Sie stattdessen entweder die Option Anfragen nicht signieren oder Autorisierungsheader nicht überschreiben wählen, CloudFront wird das Verbindungsprotokoll verwendet, das Sie in den folgenden Richtlinien angegeben haben:

- Viewer-Protokollrichtlinie
- Ursprungsprotokollrichtlinie (nur benutzerdefinierte Ursprünge)

Wenn Sie beispielsweise „Autorisierungsheader nicht überschreiben“ wählen und HTTPS zwischen CloudFront und Ihrem Amazon S3 S3-Ursprung verwenden möchten, verwenden Sie „HTTP zu HTTPS umleiten“ oder „Nur HTTPS“ für die Viewer-Protokollrichtlinie.

Erteilen Sie die CloudFront Erlaubnis, auf den S3-Bucket zuzugreifen

Bevor Sie eine Origin-Zugriffskontrolle (OAC) erstellen oder sie in einer CloudFront Distribution einrichten, stellen Sie sicher, dass diese Person über die Zugriffsberechtigung für den S3-Bucket-Ursprung CloudFront verfügt. Tun Sie dies, nachdem Sie eine CloudFront Distribution erstellt haben, aber bevor Sie das OAC zum S3-Ursprung in der Distributionskonfiguration hinzufügen.

Verwenden Sie eine S3-Bucket-Richtlinie, um dem CloudFront Dienstprinzipal (`cloudfront.amazonaws.com`) den Zugriff auf den Bucket zu ermöglichen. Verwenden Sie ein `Condition` Element in der Richtlinie, CloudFront um nur dann auf den Bucket zuzugreifen, wenn die Anfrage im Namen der CloudFront Distribution erfolgt, die den S3-Ursprung enthält. Dies ist die Distribution mit dem S3-Ursprung, dem Sie OAC hinzufügen möchten.

Informationen zum Hinzufügen oder Ändern einer Bucket-Richtlinie finden Sie unter Hinzufügen einer Bucket-Richtlinie mit der Amazon-S3-Konsole im Amazon-S3-Benutzerhandbuch.

Im Folgenden finden Sie Beispiele für S3-Bucket-Richtlinien, die einer CloudFront Distribution mit OAC-aktiviertem Zugriff auf einen S3-Ursprung ermöglichen.

Example S3-Bucket-Richtlinie, die nur Lesezugriff für eine CloudFront Distribution mit aktiviertem OAC ermöglicht

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "AllowCloudFrontServicePrincipalReadOnly",
      "Effect": "Allow",
      "Principal": {
        "Service": "cloudfront.amazonaws.com"
      },
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
      "Condition": {
        "StringEquals": {
          "AWS:SourceArn":
 "arn:aws:cloudfront::111122223333:distribution/<CloudFront distribution ID>"
        }
      }
    }
  ]
}
```

Example S3-Bucket-Richtlinie, die Lese- und Schreibzugriff für eine CloudFront Distribution mit aktiviertem OAC ermöglicht

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "AllowCloudFrontServicePrincipalReadWrite",
      "Effect": "Allow",
      "Principal": {
        "Service": "cloudfront.amazonaws.com"
      },
```

```
      "Action": [
        "s3:GetObject",
        "s3:PutObject"
      ],
      "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
      "Condition": {
        "StringEquals": {
          "AWS:SourceArn":
 "arn:aws:cloudfront::111122223333:distribution/CloudFront-distribution-ID>"
        }
      }
    }
  ]
}
```

SSE-KMS

Wenn die Objekte im S3-Bucket-Ursprung mit serverseitiger Verschlüsselung mit AWS Key Management Service (SSE-KMS) verschlüsselt sind, müssen Sie sicherstellen, dass die CloudFront Distribution über die Berechtigung zur Verwendung des AWS KMS Schlüssels verfügt. Um der CloudFront Distribution die Erlaubnis zur Verwendung des KMS-Schlüssels zu erteilen, fügen Sie der KMS-Schlüsselrichtlinie eine Anweisung hinzu. Weitere Informationen zum Ändern einer Schlüsselrichtlinie finden Sie unter Ändern einer Schlüsselrichtlinie im AWS Key Management Service -Entwicklerhandbuch.

Example Anweisung für die KMS-Schlüsselrichtlinie

Das folgende Beispiel zeigt eine AWS KMS Richtlinienanweisung, die es der CloudFront Distribution mit OAC ermöglicht, auf einen KMS-Schlüssel für SSE-KMS zuzugreifen.

```
{
    "Sid": "AllowCloudFrontServicePrincipalSSE-KMS",
    "Effect": "Allow",
    "Principal": {
        "Service": [
            "cloudfront.amazonaws.com"
        ]
     },
    "Action": [
        "kms:Decrypt",
        "kms:Encrypt",
```

```
        "kms:GenerateDataKey*"
    ],
    "Resource": "*",
    "Condition": {
            "StringEquals": {
                "AWS:SourceArn":
 "arn:aws:cloudfront::111122223333:distribution/<CloudFront distribution ID>"
            }
        }
}
```

Erstellen der Ursprungszugriffssteuerung

Um eine Origin Access Control (OAC) zu erstellen, können Sie die AWS-Managementkonsole, CloudFormation AWS CLI, oder die CloudFront API verwenden.

Console

So erstellen Sie eine Ursprungszugriffssteuerung

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home.
2. Wählen Sie im Navigationsbereich Origin access (Ursprungszugriff) aus.
3. Wählen Sie Create control setting (Kontrolleinstellung erstellen) aus.
4. Gehen Sie im Formular Create control setting (Kontrolleinstellung erstellen) wie folgt vor:
   a. Geben Sie im Bereich Details einen Namen (Name) und (optional) eine Beschreibung (Description) für die Ursprungszugriffssteuerung ein.
   b. Es empfiehlt sich, im Bereich Settings (Einstellungen) die Standardeinstellung (Sign requests (recommended)) (Anforderungen signieren (empfohlen)) zu belassen. Weitere Informationen finden Sie unter the section called “Erweiterte Einstellungen für die Ursprungszugriffssteuerung”.
5. Wählen Sie „S3“ aus der Dropdown-Liste Origin type (Ursprungstyp) aus.
6. Wählen Sie Erstellen aus.

   Nachdem die OAC erstellt wurde, notieren Sie sich den Namen. Sie benötigen diesen im folgenden Verfahren.

So fügen Sie eine Ursprungszugriffssteuerung einem S3-Ursprung in einer Distribution hinzu

1. Öffnen Sie die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home.
2. Wählen Sie eine Distribution mit einem S3-Ursprung aus, der Sie die OAC hinzufügen möchten, und wählen Sie dann den Tab Origins (Ursprünge) aus.
3. Wählen Sie den S3-Ursprung aus, dem Sie die OAC hinzufügen möchten, und wählen Sie dann Edit (Bearbeiten) aus.
4. Wählen Sie unter Ursprungszugriff die Option Einstellungen der Ursprungszugriffssteuerung (empfohlen) aus.
5. Wählen Sie im Dropdown-Menü Origin access control (Ursprungszugriffssteuerung) die OAC aus, die Sie verwenden möchten.
6. Wählen Sie Änderungen speichern aus.

Die Verteilung beginnt mit der Bereitstellung an allen CloudFront Edge-Standorten. Wenn ein Edge-Standort die neue Konfiguration erhält, signiert er alle Anforderungen, die er an den S3-Bucket-Ursprung sendet.

CloudFormation

Verwenden Sie den `AWS::CloudFront::OriginAccessControl` Ressourcentyp CloudFormation, um eine Origin Access Control (OAC) mit zu erstellen. Das folgende Beispiel zeigt die CloudFormation Vorlagensyntax im YAML-Format für die Erstellung einer Origin-Zugriffskontrolle.

```
Type: AWS::CloudFront::OriginAccessControl
Properties:
  OriginAccessControlConfig:
      Description: An optional description for the origin access control
      Name: ExampleOAC
      OriginAccessControlOriginType: s3
      SigningBehavior: always
      SigningProtocol: sigv4
```

Weitere Informationen finden Sie unter AWS::CloudFront:: OriginAccessControl im AWS CloudFormation Benutzerhandbuch.

CLI

Verwenden Sie den aws cloudfront create-origin-access-control Befehl, um eine Origin-Zugriffskontrolle mit dem AWS Command Line Interface (AWS CLI) zu erstellen. Sie können eine Eingabedatei verwenden, um die Eingabeparameter für den Befehl bereitzustellen, anstatt jeden einzelnen Parameter als Befehlszeileneingabe anzugeben.

So erstellen Sie eine Ursprungszugriffssteuerung (CLI mit Eingabedatei)

1. Verwenden Sie den folgenden Befehl zum Erstellen einer Datei mit dem Namen `origin-access-control.yaml`. Diese Datei enthält alle Eingabeparameter für den Befehl create-origin-access-control.

```
aws cloudfront create-origin-access-control --generate-cli-skeleton yaml-input >
 origin-access-control.yaml
```

2. Öffnen Sie die Datei `origin-access-control.yaml`, die Sie gerade erstellt haben. Bearbeiten Sie die Datei, um einen Namen für die OAC und eine Beschreibung (optional) hinzuzufügen, und ändern Sie `SigningBehavior` zu `always`. Speichern Sie dann die Datei.

   Weitere Informationen zu anderen OAC-Einstellungen finden Sie unter the section called “Erweiterte Einstellungen für die Ursprungszugriffssteuerung”.

3. Verwenden Sie den folgenden Befehl, um die Ursprungszugriffssteuerung mit Eingabeparametern aus der Datei `origin-access-control.yaml` zu erstellen.

```
aws cloudfront create-origin-access-control --cli-input-yaml file://origin-
access-control.yaml
```

   Notieren Sie den `Id`-Wert in der Befehlsausgabe. Sie benötigen ihn, um das OAC zu einem S3-Bucket-Ursprung in einer CloudFront Distribution hinzuzufügen.

So fügen Sie eine OAC einem S3-Bucket-Ursprung in einer vorhandenen Distribution an (CLI mit Eingabedatei)

1. Verwenden Sie den folgenden Befehl, um die Verteilungskonfiguration für die CloudFront Distribution zu speichern, zu der Sie das OAC hinzufügen möchten. Die Distribution muss über einen S3-Bucket-Ursprung verfügen.

```
aws cloudfront get-distribution-config --id <CloudFront distribution ID> --
output yaml > dist-config.yaml
```

2. Öffnen Sie die Datei mit dem Namen `dist-config.yaml`, die Sie gerade erstellt haben. Bearbeiten Sie die Datei und nehmen Sie die folgenden Änderungen vor:

   - Fügen Sie im Objekt `Origins` die ID der OAC dem Feld namens `OriginAccessControlId` hinzu.
   - Entfernen Sie den Wert aus dem Feld namens `OriginAccessIdentity`, sofern vorhanden.
   - Benennen Sie das Feld `ETag` in `IfMatch` um, ändern Sie jedoch nicht den Wert des Feldes.

   Speichern Sie die Datei, wenn Sie fertig sind.

3. Verwenden Sie den folgenden Befehl, um die Distribution zu aktualisieren und die Ursprungszugriffssteuerung zu verwenden.

```
aws cloudfront update-distribution --id <CloudFront distribution ID> --cli-
input-yaml file://dist-config.yaml
```

Die Verteilung beginnt mit der Bereitstellung an allen CloudFront Edge-Standorten. Wenn ein Edge-Standort die neue Konfiguration erhält, signiert er alle Anforderungen, die er an den S3-Bucket-Ursprung sendet.

API

Um eine Origin-Zugriffskontrolle mit der CloudFront API zu erstellen, verwenden Sie CreateOriginAccessControl. Weitere Informationen zu den Feldern, die Sie in diesem API-Aufruf

angeben, finden Sie in der API-Referenzdokumentation für Ihr AWS SDK oder einen anderen API-Client.

Nachdem Sie eine Ursprungszugriffssteuerung erstellt haben, können Sie diese mit einem der folgenden API-Aufrufe an einen S3-Bucket-Ursprung in einer Distribution anfügen:

- Um es an eine bestehende Distribution anzuhängen, verwenden Sie UpdateDistribution.
- Um es an eine neue Distribution anzuhängen, verwenden Sie CreateDistribution.

Geben Sie für beide API-Aufrufe die ID der Ursprungszugriffssteuerung im Feld `OriginAccessControlId` innerhalb eines Ursprungs an. Weitere Informationen zu den anderen Feldern, die Sie in diesen API-Aufrufen angeben, finden Sie unter Referenz für alle Distributionseinstellungen und in der API-Referenzdokumentation für Ihr AWS SDK oder einen anderen API-Client.

Löschen einer Distribution mit einer OAC, die an einen S3-Bucket angehängt ist

Wenn Sie eine Distribution löschen müssen, deren OAC an einen S3-Bucket angehängt ist, sollten Sie die Distribution löschen, bevor Sie den S3-Bucket-Ursprung löschen. Alternativ können Sie die Region in den ursprünglichen Domainnamen aufnehmen. Wenn dies nicht möglich ist, können Sie die OAC aus der Distribution entfernen, indem Sie vor dem Löschen auf den öffentlichen Zugriff wechseln. Weitere Informationen finden Sie unter Löschen einer -Verteilung.

Migration von der Ursprungszugriffsidentität (OAI) zur Ursprungszugriffssteuerung (OAC)

Um von einer Legacy-Ursprungszugriffsidentität (OAI) zu einer Ursprungszugriffssteuerung (OAC) zu migrieren, aktualisieren Sie zunächst den S3-Bucket-Ursprung so, dass sowohl die OAI als auch die Distribution mit aktivierter OAC auf den Inhalt des Buckets zugreifen können. Dadurch wird sichergestellt, dass während der Umstellung CloudFront nie der Zugriff auf den Bucket verloren geht. Um sowohl OAI als auch der Distribution mit aktivierter OAC den Zugriff auf einen S3-Bucket zu ermöglichen, aktualisieren Sie die Bucket-Richtlinie und nehmen Sie zwei Anweisungen auf, eine für jede Art von Prinzipal.

Die folgende Beispiel-S3-Bucket-Richtlinie ermöglicht sowohl einer OAI als auch einer Distribution mit aktivierter OAC den Zugriff auf einen S3-Ursprung.

Example S3-Bucket-Richtlinie, die nur Lesezugriff für eine OAI und eine CloudFront Distribution mit aktiviertem OAC ermöglicht

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowCloudFrontServicePrincipalReadOnly",
            "Effect": "Allow",
            "Principal": {
                "Service": "cloudfront.amazonaws.com"
            },
            "Action": "s3:GetObject",
            "Resource": "arn:aws:s3:::<S3 bucket name>/*",
            "Condition": {
                "StringEquals": {
                    "AWS:SourceArn":
 "arn:aws:cloudfront::111122223333:distribution/<CloudFront distribution ID>"
                }
            }
        },
        {
            "Sid": "AllowLegacyOAIReadOnly",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::cloudfront:user/CloudFront Origin Access
 Identity <origin access identity ID>"
            },
            "Action": "s3:GetObject",
            "Resource": "arn:aws:s3:::<S3 bucket name>/*"
        }
    ]
}
```

Nachdem Sie die Bucket-Richtlinie des S3-Ursprungs aktualisiert haben, um den Zugriff sowohl für OAI als auch für OAC zu ermöglichen, können Sie die Distributionskonfiguration so aktualisieren, dass OAC anstelle von OAI verwendet wird. Weitere Informationen finden Sie unter the section called “Erstellen einer neuen Ursprungszugriffssteuerung”.

Nachdem die Distribution vollständig bereitgestellt wurde, können Sie die Anweisung in der Bucket-Richtlinie, die den Zugriff für die OAI ermöglicht, entfernen. Weitere Informationen finden Sie unter the section called “Erteilen Sie die CloudFront Erlaubnis, auf den S3-Bucket zuzugreifen”.

Erweiterte Einstellungen für die Ursprungszugriffssteuerung

Die CloudFront Origin-Zugriffskontrollfunktion umfasst erweiterte Einstellungen, die nur für bestimmte Anwendungsfälle vorgesehen sind. Verwenden Sie die empfohlenen Einstellungen, sofern Sie die erweiterten Einstellungen nicht speziell benötigen.

Die Origin-Zugriffskontrolle enthält eine Einstellung namens Signaturverhalten (in der Konsole) oder `SigningBehavior` (in der API, CLI und CloudFormation). Diese Einstellung bietet die folgenden Optionen:

Ursprungsanforderungen immer signieren (empfohlene Einstellung)

Wir empfehlen die Verwendung dieser Einstellung mit der Bezeichnung Sign requests (recommended) (Anforderungen signieren (empfohlen)) in der Konsole bzw. `always` in der API, CLI und CloudFormation. Mit dieser Einstellung signiert es CloudFront immer alle Anfragen, die es an den S3-Bucket-Ursprung sendet.

Ursprungsanforderungen nie signieren

Diese Einstellung heißt Do not sign requests (Anforderungen nicht signieren) in der Konsole bzw. `never` in der API, CLI und CloudFormation. Verwenden Sie diese Einstellung, um die Ursprungszugriffssteuerung für alle Ursprünge in allen Distributionen zu deaktivieren, die diese Ursprungszugriffssteuerung verwenden. Auf diese Weise lässt sich im Vergleich zum einzelnen Entfernen einer Ursprungszugriffssteuerung aus allen Ursprüngen und Distributionen, die diese verwenden, Zeit und Aufwand sparen. Mit dieser Einstellung signiert es CloudFront keine Anfragen, die es an den S3-Bucket-Ursprung sendet.

> **Warning**
>
> Um diese Einstellung verwenden zu können, muss der S3-Bucket-Ursprung öffentlich zugänglich sein. Wenn Sie diese Einstellung mit einem S3-Bucket-Ursprung verwenden, der nicht öffentlich zugänglich ist, CloudFront können Sie nicht auf den Ursprung zugreifen. Der S3-Bucket-Ursprung gibt Fehler zurück CloudFront und CloudFront leitet diese Fehler an die Zuschauer weiter.

Viewer (Client)-**`Authorization`**-Header nicht überschreiben

Diese Einstellung heißt Do not override authorization header (Autorisierungsheader nicht überschreiben) in der Konsole bzw. `no-override` in der API, CLI und CloudFormation. Verwenden Sie diese Einstellung, wenn Sie ursprüngliche Anfragen nur signieren CloudFront möchten, wenn die entsprechende Viewer-Anfrage keinen `Authorization` Header enthält. Mit dieser Einstellung wird der `Authorization` Header der Viewer-Anfrage weitergegeben, CloudFront wenn eine vorhanden ist, signiert aber die ursprüngliche Anfrage (fügt einen eigenen `Authorization` Header hinzu), wenn die Viewer-Anfrage keinen `Authorization` Header enthält.

> **Warning**
>
> Um den `Authorization`-Header aus der Viewer-Anforderung zu übergeben, müssen Sie den `Authorization`-Header zu einer Cache-Richtlinie für alle Cache-Verhaltensweisen hinzufügen, die S3-Bucket-Ursprünge verwenden, die dieser Ursprungszugriffssteuerung zugeordnet sind.

Verwenden einer Ursprungszugriffsidentität (Legacy, nicht empfohlen)

Übersicht über das die Ursprungszugriffsidentität

CloudFront Origin Access Identity (OAI) bietet ähnliche Funktionen wie Origin Access Control (OAC), funktioniert aber nicht für alle Szenarien. Insbesondere unterstützt OAI Folgendes nicht:

- Amazon S3 S3-Buckets in allen Regionen AWS-Regionen, einschließlich optionaler Regionen
- Serverseitige Amazon S3 S3-Verschlüsselung mit AWS KMS () SSE-KMS
- Dynamische Anforderungen (`PUT`, `POST` oder `DELETE`) an Amazon S3
- Nach Januar 2023 neu AWS-Regionen auf den Markt gebracht

> **Tip**
>
> Stattdessen empfehlen wir, OAC zu verwenden. Informationen zum Einrichten von OAC finden Sie unter Erstellen einer neuen Ursprungszugriffssteuerung. Weitere Informationen zur Migration von OAI zu OAC finden Sie unter the section called “Migration von der Ursprungszugriffsidentität (OAI) zur Ursprungszugriffssteuerung (OAC)”.

Erteilen der Berechtigung zum Lesen von Dateien im Amazon-S3-Bucket für eine Ursprungszugriffsidentität

Wenn Sie mit der CloudFront Konsole eine OAI erstellen oder einer Distribution hinzufügen, können Sie die Amazon S3 S3-Bucket-Richtlinie automatisch aktualisieren, um der OAI Zugriff auf Ihren Bucket zu gewähren. Alternativ können Sie die Bucket-Richtlinie manuell erstellen oder aktualisieren. Unabhängig davon, welche Methode Sie verwenden, sollten Sie die Berechtigungen dennoch überprüfen, um Folgendes sicherzustellen:

- Ihre CloudFront OAI kann im Namen von Zuschauern, die sie anfordern, auf Dateien im Bucket zugreifen. CloudFront
- Zuschauer können Amazon S3 S3-URLs nicht verwenden, um auf Ihre Dateien außerhalb von zuzugreifen CloudFront.

> **Important**
>
> Wenn Sie so konfigurieren CloudFront , dass alle CloudFront unterstützten HTTP-Methoden akzeptiert und weitergeleitet werden, stellen Sie sicher, dass Sie Ihrer CloudFront OAI die gewünschten Berechtigungen erteilen. Wenn Sie beispielsweise so konfigurieren CloudFront , dass Anfragen, die diese `DELETE` Methode verwenden, angenommen und weitergeleitet werden, konfigurieren Sie Ihre Bucket-Richtlinie so, dass `DELETE` Anfragen entsprechend behandelt werden, sodass Zuschauer nur Dateien löschen können, die Sie möchten.

Verwenden von Richtlinien für Amazon-S3-Buckets

Sie können einer CloudFront OAI Zugriff auf Dateien in einem Amazon S3 S3-Bucket gewähren, indem Sie die Bucket-Richtlinie auf folgende Weise erstellen oder aktualisieren:

- Verwenden des Tabs Permissions (Berechtigungen) des Amazon-S3-Buckets in der Amazon-S3-Konsole.
- Verwendung PutBucketPolicyin der Amazon S3 S3-API.
- Verwenden der CloudFront-Konsole Wenn Sie Ihren Origin-Einstellungen in der CloudFront Konsole eine OAI hinzufügen, können Sie Ja, die Bucket-Richtlinie aktualisieren wählen, CloudFront um mitzuteilen, dass die Bucket-Richtlinie in Ihrem Namen aktualisiert werden soll.

Wenn Sie die Bucket-Richtlinie manuell aktualisieren, stellen Sie Folgendes sicher:

- Geben Sie die korrekte OAI als das `Principal` in der Richtlinie an.
- Erteilen Sie der OAI die Berechtigungen, die es für den Zugriff auf Objekte im Auftrag von Viewern benötigt.

Weitere Informationen finden Sie in den folgenden Abschnitten.

Geben Sie in einer Bucket-Richtlinie eine OAI als **`Principal`** an

Um eine OAI als `Principal` in einer Amazon-S3-Bucket-Richtlinie anzugeben, verwenden Sie den Amazon-Ressourcennamen (ARN) der OAI, der die ID der OAI enthält. Beispiel:

```
"Principal": {
    "AWS": "arn:aws:iam::cloudfront:user/CloudFront Origin Access Identity <origin
 access identity ID>"
}
```

Suchen Sie die OAI-ID in der CloudFront Konsole unter Sicherheit, Origin-Zugriff, Identitäten (veraltet). Alternativ können Sie sie ListCloudFrontOriginAccessIdentitiesin der CloudFront API verwenden.

Erteilen von Berechtigungen für eine OAI

Um der OAI die Berechtigungen für den Zugriff auf Objekte in Ihrem Amazon-S3-Bucket zu erteilen, verwenden Sie Aktionen in der Richtlinie, die sich auf bestimmte Amazon-S3-API-Operationen beziehen. Die `s3:GetObject`-Aktion ermöglicht es der OAI beispielsweise, Objekte im Bucket zu lesen. Weitere Informationen finden Sie in den Beispielen im folgenden Abschnitt oder unter Amazon S3-Aktionen im Benutzerhandbuch zu Amazon Simple Storage Service.

Beispiele für Amazon-S3-Bucket-Richtlinien

Die folgenden Beispiele zeigen Amazon S3 S3-Bucket-Richtlinien, die es CloudFront OAI ermöglichen, auf einen S3-Bucket zuzugreifen.

Suchen Sie die OAI-ID in der CloudFront Konsole unter Sicherheit, Origin-Zugriff, Identitäten (veraltet). Alternativ können Sie sie ListCloudFrontOriginAccessIdentitiesin der CloudFront API verwenden.

Example Amazon-S3-Bucket-Richtlinie, die der OAI Lesezugriff gewährt

Im folgenden Beispiel kann die OAI Objekte im angegebenen Bucket (`s3:GetObject`) lesen.

JSON

```
{
    "Version":"2012-10-17",
    "Id": "PolicyForCloudFrontPrivateContent",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::cloudfront:user/CloudFront Origin Access
 Identity <origin access identity ID>"
            },
            "Action": "s3:GetObject",
            "Resource": "arn:aws:s3:::<S3 bucket name>/*"
        }
    ]
}
```

Example Amazon-S3-Bucket-Richtlinie, die der OAI Lese- und Schreibzugriff gewährt

Im folgenden Beispiel kann die OAI Objekte im angegebenen Bucket (`s3:GetObject` und `s3:PutObject`) lesen und schreiben. Auf diese Weise können Zuschauer Dateien in Ihren Amazon S3 S3-Bucket hochladen CloudFront.

JSON

```
{
    "Version":"2012-10-17",
    "Id": "PolicyForCloudFrontPrivateContent",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::cloudfront:user/CloudFront Origin Access
 Identity <origin access identity ID>"
            },
            "Action": [
                "s3:GetObject",
                "s3:PutObject"
            ],
```

```
            "Resource": "arn:aws:s3:::<S3 bucket name>/*"
        }
    ]
}
```

Verwenden von Objekt-ACLs in Amazon S3 (nicht empfohlen)

> **Important**
>
> Es wird empfohlen, Amazon-S3-Bucket-Richtlinien zu verwenden, um einer OAI Zugriff auf einen S3-Bucket zu gewähren. Sie können Access Control Lists (ACLs, Zugriffssteuerungslisten) wie in diesem Abschnitt beschrieben verwenden, dies wird jedoch nicht empfohlen.
>
> Amazon S3 empfiehlt die Einstellung von S3 Object Ownership (S3-Objekteigentümerschaft) auf bucket owner enforced (Bucket-Eigentümer erzwungen), was bedeutet, dass ACLs für den Bucket und die darin enthaltenen Objekte deaktiviert sind. Wenn Sie diese Einstellung für Objekteigentümerschaft anwenden, müssen Sie Bucket-Richtlinien verwenden, um Zugriff auf die OAI zu gewähren (siehe vorheriger Abschnitt).
>
> Der folgende Abschnitt gilt nur für Anwendungsfälle für Legacy-Systeme, für die ACLs erforderlich sind.

Sie können einer CloudFront OAI Zugriff auf Dateien in einem Amazon S3 S3-Bucket gewähren, indem Sie die ACL der Datei auf folgende Weise erstellen oder aktualisieren:

- Verwenden des Tabs Permissions (Berechtigungen) des Amazon-S3-Objekts in der Amazon-S3-Konsole.
- Verwendung PutObjectAclin der Amazon S3 S3-API.

Wenn Sie einer OAI über eine ACL Zugriff gewähren, müssen Sie die OAI mit seiner kanonischen Amazon-S3-Benutzer-ID angeben. In der CloudFront Konsole finden Sie diese ID unter Sicherheit, Origin-Zugriff, Identitäten (veraltet). Wenn Sie die CloudFront API verwenden, verwenden Sie den Wert des `S3CanonicalUserId` Elements, das bei der Erstellung der OAI zurückgegeben wurde, oder rufen Sie die CloudFront API ListCloudFrontOriginAccessIdentitiesauf.

Verwenden einer Ursprungszugriffsidentität in Amazon-S3-Regionen, die nur die Authentifizierung mittels Signature Version 4 unterstützen

In neueren Amazon-S3-Regionen ist es erforderlich, Signature Version 4 für authentifizierte Anforderungen zu verwenden. (Informationen zu den Signaturversionen, die in jeder Amazon-S3-Region unterstützt werden, finden Sie unter Amazon-Simple-Storage-Service-Endpunkte und -Kontingente in der Allgemeine AWS-Referenz.) Wenn Sie eine Ursprungszugriffsidentität verwenden und sich Ihr Bucket in einer der Regionen befindet, die Signature Version 4 erfordert, beachten Sie Folgendes:

- `DELETE`-, `GET`-, `HEAD`-, `OPTIONS`- und `PATCH`-Anfragen werden ohne Einschränkungen unterstützt.
- `POST`-Anfragen werden nicht unterstützt.

Zugriffsbeschränkung mit VPC-Ursprüngen

Sie können CloudFront damit Inhalte aus Anwendungen bereitstellen, die in Ihren privaten VPC-Subnetzen (Virtual Private Cloud) gehostet werden. Sie können Application Load Balancer (ALB), Network Load Balancer (NLB) und EC2-Instances in privaten Subnetzen als VPC-Ursprünge verwenden.

Hier einige Gründe für die Verwendung von VPC-Ursprüngen:

- Sicherheit — VPC Origins wurde entwickelt, um den Sicherheitsstatus Ihrer Anwendung zu verbessern, indem Ihre Load Balancer und EC2-Instances in privaten Subnetzen platziert werden, wodurch CloudFront der zentrale Einstiegspunkt entsteht. Benutzeranfragen werden über eine private, sichere Verbindung CloudFront zu den VPC-Ursprüngen weitergeleitet und bieten so zusätzliche Sicherheit für Ihre Anwendungen.
- Verwaltung — VPC-Ursprünge reduzieren den Betriebsaufwand, der für eine sichere Konnektivität zwischen den CloudFront Ursprüngen erforderlich ist. Sie können Ihre Ursprünge in private Subnetze ohne öffentlichen Zugriff verschieben und müssen keine Zugriffssteuerungslisten (ACLs) oder andere Mechanismen implementieren, um den Zugriff auf Ihre Ursprünge einzuschränken. Auf diese Weise müssen Sie nicht in undifferenzierte Entwicklungsarbeit investieren, um Ihre Webanwendungen zu sichern. CloudFront
- Skalierbarkeit und Leistung — VPC Origins hilft Ihnen dabei, Ihre Webanwendungen zu sichern, sodass Sie Zeit haben, sich auf das Wachstum Ihrer kritischen Geschäftsanwendungen zu konzentrieren und gleichzeitig die Sicherheit zu verbessern und gleichzeitig eine hohe Leistung und globale Skalierbarkeit aufrechtzuerhalten. CloudFront VPC Origins optimiert das

Sicherheitsmanagement und reduziert die betriebliche Komplexität, sodass Sie es CloudFront als zentrale Anlaufstelle für Ihre Anwendungen verwenden können.

> **Tip**
>
> CloudFront unterstützt die gemeinsame Nutzung von VPC-Ursprüngen AWS-Konten, unabhängig davon, ob sie sich in Ihrer Organisation befinden oder nicht. Sie können VPC-Ursprünge von der CloudFront Konsole aus teilen oder AWS Resource Access Manager (AWS RAM) verwenden. Weitere Informationen finden Sie unter Arbeiten mit gemeinsam genutzten Ressourcen in CloudFront.

Voraussetzungen

Bevor Sie einen VPC-Ursprung für Ihre CloudFront Distribution erstellen, müssen Sie die folgenden Schritte ausführen:

VPC-Konfiguration

Erstellen Sie eine virtuelle private Cloud (VPC) auf Amazon VPC in einer der AWS-Regionen , die für VPC-Ursprünge unterstützt werden. Weitere Informationen über das Erstellen einer VPC finden Sie unter Erstellen einer VPC sowie anderer VPC-Ressourcen im Benutzerhandbuch für Amazon VPC. Eine Liste der unterstützten -Regionen finden Sie unter Unterstützt AWS-Regionen für VPC-Ursprünge.

Ihre VPC muss Folgendes beinhalten:

- Internet-Gateway – Sie müssen der VPC, die über Ihre VPC-Ursprungsressourcen verfügt, ein Internet-Gateway hinzufügen. Das Internet-Gateway ist erforderlich, um anzugeben, dass die VPC Datenverkehr aus dem Internet empfangen kann. Das Internet-Gateway wird nicht für die Weiterleitung von Datenverkehr zu Ursprüngen innerhalb des Subnetzes verwendet und Sie müssen die Routing-Richtlinien nicht aktualisieren.
- Privates Subnetz mit mindestens einer verfügbaren IPv4-Adresse (und für Dual-Stack-Unterstützung mindestens einer verfügbaren IPv6-Adresse) — CloudFront leitet zu Ihrem Subnetz über eine serviceverwaltete elastic network interface (ENI) weiter, die CloudFront erstellt wird, nachdem Sie Ihre VPC-Ursprungsressource definiert haben. Sie müssen mindestens eine verfügbare IPv4-Adresse (und für Dual-Stack-Unterstützung mindestens eine verfügbare IPv6-

Adresse) in Ihrem privaten Subnetz haben, damit der ENI-Erstellungsprozess erfolgreich sein kann. Diese Adressen können privat sein und es fallen keine zusätzlichen Kosten für sie an.

Herkunft: Ressourcen

Starten Sie im privaten Subnetz einen Application Load Balancer, einen Network Load Balancer oder eine EC2-Instance zur Verwendung als Ursprung. Die Ressource, die Sie starten, muss vollständig bereitgestellt sein und sich im Status Aktiv befinden, bevor Sie sie als VPC-Ursprung verwenden können.

Einschränkungen bei der Herkunft:

- Gateway Load Balancer können nicht als Ursprünge hinzugefügt werden
- Network Load Balancer mit TLS-Listenern können nicht als Ursprünge hinzugefügt werden
- Um als VPC-Ursprung verwendet zu werden, muss einem Network Load Balancer eine Sicherheitsgruppe zugeordnet sein

Sicherheitsgruppenkonfiguration

Ihren VPC-Ursprungsressourcen (Application Load Balancer, Network Load Balancer oder EC2-Instance) muss eine Sicherheitsgruppe angehängt sein. Wenn Sie einen VPC-Ursprung erstellen, CloudFront wird automatisch eine vom Service verwaltete Sicherheitsgruppe mit dem Benennungsmuster erstellt. `CloudFront-VPCOrigins-Service-SG` Diese Sicherheitsgruppe wird vollständig von AWS verwaltet und sollte nicht bearbeitet werden.

Damit der Datenverkehr Ihren VPC-Ursprung erreicht, aktualisieren Sie die mit Ihrer Ursprungsressource verknüpfte Sicherheitsgruppe (ALB-, NLB- oder EC2-Instance), sodass eingehender Datenverkehr mit einer der folgenden Methoden zugelassen wird: CloudFront

- Option 1: Verkehr aus der Liste der verwalteten Präfixe zulassen. CloudFront Weitere Informationen finden Sie unter Verwenden Sie die Liste der CloudFront verwalteten Präfixe. Dies kann auch geschehen, bevor der VPC-Ursprung erstellt wird. Verwenden Sie für einen IPv4-VPC-Ursprung nur. `com.amazonaws.global.cloudfront.origin-facing` Verwenden Sie für einen Dual-Stack-VPC-Ursprung sowohl als auch`com.amazonaws.global.cloudfront.origin-facing`. `com.amazonaws.global.ipv6.cloudfront.origin-facing`
- Option 2: Datenverkehr aus der vom CloudFront Service verwalteten Sicherheitsgruppe zulassen (). `CloudFront-VPCOrigins-Service-SG` Dies ist erst möglich, nachdem der VPC-Ursprung

und die vom Service verwaltete Sicherheitsgruppe erstellt wurden. Diese Konfiguration ist noch restriktiver, da sie den Datenverkehr nur auf Ihre Distributionen beschränkt. CloudFront

> **Important**
>
> Erstellen Sie keine eigene Sicherheitsgruppe, deren Name mit beginnt. `CloudFront-VPCOrigins-Service-SG` Dies ist ein AWS reserviertes Benennungsmuster für vom Dienst verwaltete Sicherheitsgruppen. Weitere Informationen finden Sie unter Erstellen einer Sicherheitsgruppe.

Protokoll- und Funktionseinschränkungen

VPC-Ursprünge unterstützen Folgendes nicht:

- gRPC-Verkehr
- Auslöser für Origin-Anfrage und Origin-Antwort mit Lambda @Edge
- Network Access Control Lists (NACLs) — Eingehende NACL-Regeln werden nicht für den Datenverkehr von CloudFront zu Ihrem VPC-Ursprung ausgewertet. NACL-Regeln für ausgehende Nachrichten werden jedoch auf dem Rückweg ausgewertet und müssen den Datenverkehr über ephemere TCP-Ports (1024—65535) zu oder zu den IP-Bereichen zulassen, die mit dem Ursprung verbunden sind. `0.0.0.0/0` CloudFront

Erstellen eines VPC-Ursprungs (neue Distribution)

Das folgende Verfahren zeigt Ihnen, wie Sie einen VPC-Ursprung für Ihre neue CloudFront Distribution in der CloudFront Konsole erstellen. Alternativ können Sie die CreateDistributionAPI-Operationen CreateVpcOriginund mit dem SDK AWS CLI oder einem AWS SDK verwenden.

So erstellen Sie einen VPC-Ursprung für eine neue Distribution CloudFront

1. Öffnen Sie die CloudFront Konsole unter. https://console.aws.amazon.com/cloudfront/v4/home
2. Wählen Sie VPC-Ursprünge, VPC-Ursprung erstellen aus.
3. Füllen Sie die erforderlichen Felder aus. Wählen Sie unter Ursprungs-ARN den ARN Ihres Application Load Balancer, Network Load Balancer oder Ihrer EC2-Instance aus. Wenn der ARN nicht angezeigt wird, können Sie Ihren spezifischen Ressourcen-ARN kopieren und hier einfügen.

4. Wählen Sie den Origin-Typ aus. Der Standardwert ist IPv4.
5. Wählen Sie VPC-Ursprung erstellen aus.
6. Warten Sie, bis sich Ihr VPC-Ursprungsstatus in Bereitgestellt ändert. Dieser Vorgang kann bis zu 15 Minuten dauern.
7. Wählen Sie Distributionen, Distribution erstellen aus.
8. Wählen Sie für Ursprungsdomain Ihre VPC-Ursprungsressource aus der Dropdown-Liste aus.

   Wenn Ihr VPC-Ursprung eine EC2-Instance ist, kopieren Sie den privaten IP-DNS-Namen der Instance und fügen Sie ihn in das Feld Ursprungsdomain ein.
9. Schließen Sie die Erstellung der Distribution ab. Weitere Informationen finden Sie unter Erstellen Sie eine CloudFront Distribution in der Konsole.

Erstellen eines VPC-Ursprungs (vorhandene Distribution)

Das folgende Verfahren zeigt Ihnen, wie Sie in der CloudFront Konsole einen VPC-Ursprung für Ihre bestehende CloudFront Distribution erstellen, um die kontinuierliche Verfügbarkeit Ihrer Anwendungen sicherzustellen. Alternativ können Sie die Operationen CreateVpcOriginund die UpdateDistributionWithStagingConfigAPI mit dem AWS CLI oder einem AWS SDK verwenden.

Optional können Sie Ihren VPC-Ursprung zu Ihrer vorhandenen Distribution hinzufügen, ohne eine Staging-Distribution zu erstellen.

Um einen VPC-Ursprung für Ihre bestehende CloudFront Distribution zu erstellen

1. Öffnen Sie die CloudFront Konsole unter. https://console.aws.amazon.com/cloudfront/v4/home
2. Wählen Sie VPC-Ursprünge, VPC-Ursprung erstellen aus.
3. Füllen Sie die erforderlichen Felder aus. Wählen Sie unter Ursprungs-ARN den ARN Ihres Application Load Balancer, Network Load Balancer oder Ihrer EC2-Instance aus. Wenn der ARN nicht angezeigt wird, können Sie Ihren spezifischen Ressourcen-ARN kopieren und hier einfügen.
4. Wählen Sie den Origin-Typ aus. Der Standardwert ist IPv4.
5. Wählen Sie VPC-Ursprung erstellen aus.
6. Warten Sie, bis sich Ihr VPC-Ursprungsstatus in Bereitgestellt ändert. Dieser Vorgang kann bis zu 15 Minuten dauern.
7. Rufen Sie im Navigationsbereich Distributions auf.
8. Wählen Sie die ID Ihrer Distribution aus.

9. Wählen Sie auf der Registerkarte Allgemeines unter Kontinuierliche Bereitstellung die Option Staging-Distribution erstellen aus. Weitere Informationen finden Sie unter Verwenden Sie CloudFront Continuous Deployment, um CDN-Konfigurationsänderungen sicher zu testen.
10. Folgen Sie den Schritten im Assistenten Staging-Distribution erstellen, um eine Staging-Distribution zu erstellen. Führen Sie die folgenden Schritte aus:

    - Wählen Sie unter Ursprünge die Option Ursprung erstellen aus.
    - Wählen Sie für Ursprungsdomain Ihre VPC-Ursprungsressource aus dem Dropdown-Menü aus.

      Wenn Ihr VPC-Ursprung eine EC2-Instance ist, kopieren Sie den privaten IP-DNS-Namen der Instance und fügen Sie ihn in das Feld Ursprungsdomain ein.
    - Wählen Sie Create Origin (Ursprung erstellen) aus.
11. Testen Sie in Ihrer Staging-Distribution den VPC-Ursprung.
12. Stufen Sie die Konfiguration der Staging-Distribution zu Ihrer primären Distribution hoch. Weitere Informationen finden Sie unter Hochstufen der Konfiguration einer Staging-Distribution.
13. Entfernen Sie den öffentlichen Zugriff auf Ihren VPC-Ursprung, indem Sie das Subnetz privat machen. Danach ist der VPC-Ursprung nicht mehr über das Internet auffindbar, hat aber CloudFront weiterhin privaten Zugriff darauf. Weitere Informationen finden Sie unter Zuordnen oder Aufheben der Zuordnung eines Subnetzes zu einer Routing-Tabelle im Benutzerhandbuch für Amazon VPC.

Aktualisieren eines VPC-Ursprungs

Das folgende Verfahren zeigt Ihnen, wie Sie einen VPC-Ursprung für Ihre CloudFront Distribution in der CloudFront Konsole aktualisieren. Alternativ können Sie die UpdateVpcOriginAPI-Operationen UpdateDistributionund mit dem SDK AWS CLI oder einem AWS SDK verwenden.

Um einen vorhandenen VPC-Ursprung für Ihre CloudFront Distribution zu aktualisieren

1. Öffnen Sie die CloudFront Konsole unter. https://console.aws.amazon.com/cloudfront/v4/home
2. Rufen Sie im Navigationsbereich Distributions auf.
3. Wählen Sie die ID Ihrer Distribution aus.
4. Wählen Sie die Registerkarte Behaviors aus.
5. Stellen Sie sicher, dass der VPC-Ursprung nicht der Standardursprung für Ihr Cache-Verhalten ist.

6. Wählen Sie den Tab Ursprünge aus.
7. Wählen Sie den zu aktualisierenden VPC-Ursprung und dann Löschen aus. Dadurch wird der VPC-Ursprung von Ihrer Distribution getrennt. Wiederholen Sie die Schritte 2–7, um den VPC-Ursprung von anderen Distributionen zu trennen.
8. Wählen Sie VPC-Ursprünge aus.
9. Wählen Sie den VPC-Ursprung und dann Bearbeiten aus.
10. Nehmen Sie die Aktualisierungen vor und wählen Sie VPC-Ursprung aktualisieren aus.
11. Warten Sie, bis sich Ihr VPC-Ursprungsstatus in Bereitgestellt ändert. Dieser Vorgang kann bis zu 15 Minuten dauern.
12. Rufen Sie im Navigationsbereich Distributions auf.
13. Wählen Sie die ID Ihrer Distribution aus.
14. Wählen Sie den Tab Ursprünge aus.
15. Wählen Sie Create Origin (Ursprung erstellen) aus.
16. Wählen Sie für Ursprungsdomain Ihre VPC-Ursprungsressource aus dem Dropdown-Menü aus.

    Wenn Ihr VPC-Ursprung eine EC2-Instance ist, kopieren Sie den privaten IP-DNS-Namen der Instance und fügen Sie ihn in das Feld Ursprungsdomain ein.
17. Wählen Sie Create Origin (Ursprung erstellen) aus. Dadurch wird der VPC-Ursprung wieder mit Ihrer Distribution verknüpft. Wiederholen Sie die Schritte 12–17, um den aktualisierten VPC-Ursprung anderen Distributionen zuzuordnen.

Unterstützt AWS-Regionen für VPC-Ursprünge

VPC-Ursprünge werden derzeit in der folgenden Werbung AWS-Regionen unterstützt. Ausgeschlossene Availability Zones (AZ) sind angegeben.

Name der Region	Region
USA Ost (Ohio)	`us-east-2`
USA Ost (Nord-Virginia)	`us-east-1 (except AZ use1-az3)`
USA West (Nordkalifornien)	`us-west-1 (except AZ usw1-az2)`
USA West (Oregon)	`us-west-2`

Name der Region	Region
Africa (Cape Town)	`af-south-1`
Asia Pacific (Hong Kong)	`ap-east-1`
Asien-Pazifik (Mumbai)	`ap-south-1`
Asien-Pazifik (Hyderabad)	`ap-south-2`
Asien-Pazifik (Jakarta)	`ap-southeast-3`
Asien-Pazifik (Melbourne)	`ap-southeast-4`
Asien-Pazifik (Osaka)	`ap-northeast-3`
Asien-Pazifik (Singapur)	`ap-southeast-1`
Asien-Pazifik (Sydney)	`ap-southeast-2`
Asien-Pazifik (Tokio)	`ap-northeast-1 (except AZ apne1-az3)`
Asien-Pazifik (Seoul)	`ap-northeast-2`
Asien-Pazifik (Thailand)	`ap-southeast-7`
Asien-Pazifik (Malaysia)	`ap-southeast-5`
Asien-Pazifik (Neuseeland)	`ap-southeast-6`
Asien-Pazifik (Taipeh)	`ap-east-2`
Kanada (Zentral)	`ca-central-1 (except AZ cac1-az3)`
Kanada West (Calgary)	`ca-west-1`
Europe (Frankfurt)	`eu-central-1`
Europa (Irland)	`eu-west-1`
Europa (London)	`eu-west-2`

Name der Region	Region
Europa (Milan)	`eu-south-1`
Europa (Paris)	`eu-west-3`
Europa (Spain)	`eu-south-2`
Europa (Stockholm)	`eu-north-1`
Europa (Zürich)	`eu-central-2`
Israel (Tel Aviv)	`il-central-1`
Middle East (Bahrain)	`me-south-1`
Naher Osten (VAE)	`me-central-1`
Südamerika (São Paulo)	`sa-east-1`
Mexiko (Zentral)	`mx-central-1`

Beschränken Sie den Zugriff auf einen Amazon S3 Multi-Region Access Point-Ursprung

Sie können Origin Access Control (OAC) verwenden, um den Zugriff auf einen Amazon S3 Multi-Region Access Point-Ursprung einzuschränken. S3 Multi-Region Access Points bieten einen globalen Endpunkt, der Anfragen basierend auf der Netzwerklatenz an den nächstgelegenen S3-Bucket weiterleitet.

> **Note**
>
> OAC mit S3 Multi-Region Access Points wird für MRAP-Endpunkte, die durch Buckets in Opt-in-Regionen unterstützt werden, nicht unterstützt. Anfragen, die an einen Bucket in einer Opt-In-Region weitergeleitet werden, schlagen fehl. Alle mit dem Multi-Region Access Point verknüpften Buckets müssen sich in Regionen befinden, die standardmäßig aktiviert sind.

Informationen zur Verwendung von OAC mit einem standardmäßigen Amazon S3-Bucket-Ursprung finden Sie unter. the section called “Beschränken des Zugriffs auf einen Amazon-S3-Ursprung”

Voraussetzungen

Bevor Sie OAC erstellen und einrichten, benötigen Sie eine CloudFront Distribution mit einem Amazon S3 Multi-Region Access Point-Ursprung. Der Ursprungsdomänenname muss das S3 Multi-Region Access Point-Hostnamenformat verwenden:

```
multi-region-access-point-alias.accesspoint.s3-global.amazonaws.com
```

Weitere Informationen zum Erstellen eines S3 Multi-Region Access Points finden Sie unter Creating Multi-Region Access Points im Amazon Simple Storage Service-Benutzerhandbuch.

Erteilen Sie die CloudFront Erlaubnis, auf den S3 Multi-Region Access Point zuzugreifen

Aktualisieren Sie die Multi-Region Access Point-Richtlinie, um dem CloudFront Service Principal (`cloudfront.amazonaws.com`) den Zugriff auf den Multi-Region Access Point zu ermöglichen. Verwenden Sie ein `Condition` Element in der Richtlinie, um den Zugriff auf den Multi-Region Access Point nur dann CloudFront zu gestatten, wenn die Anfrage im Namen der CloudFront Distribution erfolgt, die den Ursprung enthält.

Informationen zum Hinzufügen oder Ändern einer Multi-Region Access Point-Richtlinie finden Sie unter Beispiele für Multi-Region Access Point-Richtlinien im Amazon Simple Storage Service-Benutzerhandbuch.

Example Multi-Region Access Point-Richtlinie für CloudFront OAC

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowCloudFrontOACAccess",
            "Effect": "Allow",
            "Principal": {
                "Service": "cloudfront.amazonaws.com"
            },
            "Action": "s3:GetObject",
            "Resource": "arn:aws:s3::111122223333:accesspoint/Multi-Region-Access-
Point-Alias.mrap/object/*",
```

```
            "Condition": {
                "StringEquals": {
                    "aws:SourceArn":
 "arn:aws:cloudfront::111122223333:distribution/CloudFront distribution ID"
                }
            }
        }
    ]
}
```

Erteilen Sie die CloudFront Berechtigung für den Zugriff auf die zugrunde liegenden S3-Buckets

Zusätzlich zur Multi-Region Access Point-Richtlinie müssen Sie auch die CloudFront Zugriffsberechtigung für jeden der zugrunde liegenden S3-Buckets gewähren, die dem Multi-Region Access Point zugeordnet sind. Dafür stehen Ihnen zwei Optionen zur Verfügung:

> **Important**
>
> Sie müssen diese Bucket-Richtlinie jedem S3-Bucket hinzufügen, der dem Multi-Region Access Point zugeordnet ist. Wenn in einem Bucket die Richtlinie fehlt, werden CloudFront Anfragen, die an diesen Bucket weitergeleitet werden, abgelehnt.

Option 1: Gewähren Sie nur Zugriff auf CloudFront

Fügen Sie jedem S3-Bucket eine Bucket-Richtlinie hinzu, die dem CloudFront Dienstprinzipal den Zugriff auf den Bucket ermöglicht. Verwenden Sie diese Option, wenn Sie auch den direkten Zugriff auf den Bucket aus anderen Quellen zulassen müssen.

Example S3-Bucket-Richtlinie für einen zugrunde liegenden Bucket

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowCloudFrontOACAccessViaMRAP",
            "Effect": "Allow",
            "Principal": {
                "Service": "cloudfront.amazonaws.com"
            },
```

```
            "Action": "s3:GetObject",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket-us-east-1/*",
            "Condition": {
                "StringEquals": {
                    "aws:SourceArn":
 "arn:aws:cloudfront::111122223333:distribution/CloudFront distribution ID"
                }
            }
        }
    ]
}
```

Option 2: Delegieren Sie den vollen Bucket-Zugriff an den Multi-Region Access Point

Gewähren Sie dem Multi-Region Access Point vollen Zugriff auf jeden zugrunde liegenden Bucket. Bei diesem Ansatz wird der gesamte Zugriff auf den Bucket durch die Multi-Region Access Point-Richtlinie gesteuert, was die Zugriffsverwaltung vereinfacht. Wir empfehlen diese Option für Anwendungsfälle, die keinen direkten Zugriff auf den Bucket erfordern.

Example S3-Bucket-Richtlinie, die den Zugriff auf den Multi-Region Access Point delegiert

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "DelegateAccessToMRAP",
            "Effect": "Allow",
            "Principal": "*",
            "Action": "s3:*",
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket-us-east-1",
                "arn:aws:s3:::amzn-s3-demo-bucket-us-east-1/*"
            ],
            "Condition": {
                "StringEquals": {
                    "s3:DataAccessPointArn":
 "arn:aws:s3::111122223333:accesspoint/Multi-Region-Access-Point-Alias.mrap"
                }
            }
        }
    ]
}
```

Weitere Informationen finden Sie im Beispiel für eine Multi-Region Access Point-Richtlinie im Amazon Simple Storage Service-Benutzerhandbuch.

SSE-KMS

Wenn die Objekte in den zugrunde liegenden S3-Buckets mithilfe der serverseitigen Verschlüsselung mit AWS KMS (SSE-KMS) verschlüsselt werden, müssen Sie sicherstellen, dass die CloudFront Distribution über die Berechtigung zur Verwendung des AWS KMS Schlüssels verfügt. Da S3 Multi-Region Access Points Anfragen an Buckets in mehreren Regionen weiterleiten können, müssen Sie der KMS-Schlüsselrichtlinie in jeder Region, die ein zugrunde liegender Bucket verwendet, eine Anweisung hinzufügen. SSE-KMS Weitere Informationen zum Ändern einer Schlüsselrichtlinie finden Sie unter Ändern einer Schlüsselrichtlinie im AWS Key Management Service -Entwicklerhandbuch.

Example Anweisung für die KMS-Schlüsselrichtlinie

Das folgende Beispiel zeigt eine KMS-Schlüsselrichtlinie, die es der CloudFront Verteilung mit OAC ermöglicht, auf einen KMS-Schlüssel für zuzugreifen. SSE-KMS

```
{
    "Sid": "AllowCloudFrontServicePrincipalSSE-KMS",
    "Effect": "Allow",
    "Principal": {
        "Service": "cloudfront.amazonaws.com"
    },
    "Action": [
        "kms:Decrypt",
        "kms:Encrypt",
        "kms:GenerateDataKey*"
    ],
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "aws:SourceArn": "arn:aws:cloudfront::111122223333:distribution/CloudFront
 distribution ID"
        }
    }
}
```

> **Important**
>
> Sie müssen diese Schlüsselrichtlinienanweisung dem KMS-Schlüssel in jeder Region hinzufügen, in der ein zugrundeliegender S3-Bucket SSE-KMS Verschlüsselung verwendet.

Erstellen der Ursprungszugriffssteuerung

Um eine Origin Access Control (OAC) zu erstellen, können Sie die AWS-Managementkonsole CloudFormation AWS CLI, die oder die CloudFront API verwenden.

> **Note**
>
> Lambda @Edge -Ursprungsauslöser (sowohl die Quellanforderung als auch die Ursprungsantwort) sind nicht kompatibel mit Ursprüngen, die einen SigV4A-OAC verwenden, unabhängig davon, ob auf den Ursprung direkt oder über eine Ursprungsgruppe zugegriffen wird.

Console

So erstellen Sie eine Ursprungszugriffssteuerung

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Konsole unter. CloudFront https://console.aws.amazon.com/cloudfront/v4/home
2. Wählen Sie im Navigationsbereich Origin access (Ursprungszugriff) aus.
3. Wählen Sie Create control setting (Kontrolleinstellung erstellen) aus.
4. Gehen Sie im Formular Create control setting (Kontrolleinstellung erstellen) wie folgt vor:
   a. Geben Sie im Bereich Details einen Namen (Name) und (optional) eine Beschreibung (Description) für die Ursprungszugriffssteuerung ein.
   b. Es empfiehlt sich, im Bereich Settings (Einstellungen) die Standardeinstellung (Sign requests (recommended)) (Anforderungen signieren (empfohlen)) zu belassen. Weitere Informationen finden Sie unter the section called “Erweiterte Einstellungen für die Ursprungszugriffssteuerung”.
5. Wählen Sie S3 aus der Dropdownliste für den Origin-Typ aus und wählen Sie die Option SigV4A-Signaturprotokoll verwenden aus.

6. Wählen Sie Erstellen aus.

   Nachdem die OAC erstellt wurde, notieren Sie sich den Namen. Sie benötigen diesen im folgenden Verfahren.

Um einem S3 Access Point-Ursprung in einer Distribution eine Multi-Region Origin-Zugriffskontrolle hinzuzufügen

1. Öffnen Sie die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home.
2. Wählen Sie eine Distribution mit einem S3 Multi-Region Access Point-Ursprung, zu der Sie den OAC hinzufügen möchten, und wählen Sie dann die Registerkarte Origins.
3. Wählen Sie den S3 Multi-Region Access Point-Ursprung aus, zu dem Sie den OAC hinzufügen möchten, und wählen Sie dann Bearbeiten.
4. Wählen Sie unter Ursprungszugriff die Option Einstellungen der Ursprungszugriffssteuerung (empfohlen) aus.
5. Wählen Sie im Dropdown-Menü Origin access control (Ursprungszugriffssteuerung) die OAC aus, die Sie verwenden möchten.
6. Wählen Sie Änderungen speichern aus.

Die Verteilung beginnt mit der Bereitstellung an allen CloudFront Edge-Standorten. Wenn ein Edge-Standort die neue Konfiguration empfängt, signiert er alle Anfragen, die er an den S3 Multi-Region Access Point-Ursprung sendet.

CLI

Verwenden Sie den create-origin-access-control-Befehl:

```
aws cloudfront create-origin-access-control \
    --origin-access-control-config '{
        "Name": "my-s3-mrap-oac",
        "Description": "OAC for S3 Multi-Region Access Point",
        "SigningProtocol": "sigv4a",
        "SigningBehavior": "always",
        "OriginAccessControlOriginType": "s3"
    }'
```

CloudFormation

Geben Sie die folgenden Werte in das ein`OriginAccessControlConfig`:

- `SigningProtocol`: `sigv4a`
- `SigningBehavior`: `always`, `never`, oder `no-override`
- `OriginAccessControlOriginType`: `s3`

Example CloudFormation Vorlage

```
Type: AWS::CloudFront::OriginAccessControl
Properties:
  OriginAccessControlConfig:
    Description: An optional description for the origin access control
    Name: my-s3-mrap-oac
    OriginAccessControlOriginType: s3
    SigningBehavior: always
    SigningProtocol: sigv4a
```

Verhalten beim Signieren

Die Optionen für das Signierungsverhalten für S3 Multi-Region Access Point-Ursprünge sind dieselben wie für reguläre Amazon S3-Bucket-Ursprünge. Weitere Informationen finden Sie unter the section called “Erweiterte Einstellungen für die Ursprungszugriffssteuerung” Beschränken Sie den Zugriff auf einen Amazon S3-Ursprung.

Beschränken des Zugriffs auf Application Load Balancer

Sie können sowohl interne als auch über das Internet zugängliche Application Load Balancer mit Amazon verwenden. CloudFront Sie können interne Application Load Balancer in privaten Subnetzen verwenden, indem Sie VPC-Ursprünge verwenden. CloudFront CloudFront VPC-Ursprünge ermöglichen es Ihnen, Inhalte von Anwendungen bereitzustellen, die in privaten VPC-Subnetzen gehostet werden, ohne sie dem öffentlichen Internet zugänglich zu machen. Weitere Informationen finden Sie unter Zugriffsbeschränkung mit VPC-Ursprüngen.

Wenn Sie einen mit dem Internet verbundenen Application Load Balancer mit verwenden, können Sie mithilfe der folgenden Sicherheitsmaßnahmen verhindern CloudFront, dass Benutzer direkt auf einen Application Load Balancer zugreifen, und den Zugriff nur über diesen zulassen. CloudFront

1. Konfigurieren Sie CloudFront die Konfiguration so, dass Anforderungen, die an den Application Load Balancer gesendet werden, mit einem benutzerdefinierten HTTP-Header versehen werden.
2. Konfigurieren Sie den Application Load Balancer so, dass nur Anforderungen weitergeleitet werden, die den benutzerdefinierten HTTP-Header enthalten.
3. Fordern Sie HTTPS an, um die Sicherheit dieser Lösung zu verbessern.

CloudFront kann auch dazu beitragen, die Latenz zu reduzieren und sogar einige Distributed-Denial-of-Service (DDoS) -Angriffe abzufangen.

Wenn Ihr Anwendungsfall den doppelten Zugriff auf Webanwendungen sowohl von beiden als auch CloudFront von Application Load Balancer direkt über das Internet erfordert, sollten Sie erwägen, Ihre Webanwendungs-APIs wie folgt aufzuteilen:

- APIs, die durchgehen müssen. CloudFront In diesem Fall sollten Sie erwägen, einen separaten privaten Application Load Balancer als Ursprung zu verwenden.
- APIs, die Zugriff über Application Load Balancer erfordern. In diesem Fall umgehen Sie CloudFront.

Alternativ können Sie für eine Webanwendung oder andere Inhalte, die von einem mit dem Internet verbundenen Application Load Balancer in Elastic Load Balancing bereitgestellt CloudFront werden, Objekte zwischenspeichern und sie Benutzern (Zuschauern) direkt zur Verfügung stellen, wodurch die Last auf Ihrem Application Load Balancer reduziert wird. Ein mit dem Internet verbundener Load Balancer hat einen öffentlich auflösbaren DNS-Namen und leitet Anforderungen von Clients über das Internet an Ziele weiter.

Weitere Informationen finden Sie unter den folgenden Themen. Nachdem Sie diese Schritte abgeschlossen haben, können Benutzer nur noch über diesen auf Ihren Application Load Balancer zugreifen. CloudFront

Themen

- Konfigurieren Sie CloudFront , um Anfragen einen benutzerdefinierten HTTP-Header hinzuzufügen
- Konfigurieren eines Application Load Balancers zum Weiterleiten von Anforderungen, die einen bestimmten Header enthalten
- (Optional) Verbesserung der Sicherheit dieser Lösung
- (Optional) Beschränken Sie den Zugriff auf Origin mithilfe der AWS-verwaltete Präfixliste für CloudFront

Konfigurieren Sie CloudFront , um Anfragen einen benutzerdefinierten HTTP-Header hinzuzufügen

Sie können so konfigurieren CloudFront , dass den Anforderungen, die an Ihren Ursprung gesendet werden, ein benutzerdefinierter HTTP-Header hinzugefügt wird (in diesem Fall ein Application Load Balancer).

> **Important**
>
> Dieser Anwendungsfall beruht darauf, den Namen und den Wert des benutzerdefinierten Headers geheim zu halten. Wenn der Header-Name und der Wert nicht geheim sind, könnten andere HTTP-Clients sie möglicherweise in Anfragen aufnehmen, die sie direkt an den Application Load Balancer senden. Dies kann dazu führen, dass sich der Application Load Balancer so verhält, als kämen die Anfragen von einem Ort, an dem CloudFront sie nicht stammen. Um dies zu verhindern, halten Sie den Namen und den Wert des benutzerdefinierten Headers geheim.

Sie können mithilfe der CloudFront Konsole oder der CloudFront API konfigurieren CloudFront , dass den ursprünglichen Anfragen ein benutzerdefinierter HTTP-Header hinzugefügt wird. CloudFormation

Um einen benutzerdefinierten HTTP-Header hinzuzufügen (CloudFront Konsole)

Verwende in der CloudFront Konsole die Einstellung für benutzerdefinierte Origin-Header in den Origin-Einstellungen. Geben Sie den Header-Namen und seinen Wert ein.

> **Note**
>
> Verwenden Sie in der Produktion zufällig generierte Header-Namen und -Werte. Behandeln Sie Header-Namen und -Werte wie sichere Anmeldeinformationen, wie Benutzernamen und Passwörter.

Du kannst die Origin-Einstellung für benutzerdefinierte Header bearbeiten, wenn du einen Origin für eine bestehende CloudFront Distribution erstellst oder bearbeitest und wenn du eine neue Distribution erstellst. Weitere Informationen erhalten Sie unter Eine Verteilung aktualisieren und Eine Distribution erstellen.

So fügen Sie einen benutzerdefinierten HTTP-Header hinzu (CloudFormation)

Verwenden Sie die `OriginCustomHeaders` Eigenschaft in einer CloudFormation Vorlage, wie im folgenden Beispiel gezeigt.

> **Note**
>
> Der Header-Name und der Wert in diesem Beispiel dienen nur zur Demonstration. Verwenden Sie in der Produktion zufällig generierte Werte. Behandeln Sie den Namen und den Wert des Headers als sichere Berechtigung, ähnlich einem Benutzernamen und einem Passwort.

```
AWSTemplateFormatVersion: '2010-09-09'
Resources:
  TestDistribution:
    Type: 'AWS::CloudFront::Distribution'
    Properties:
      DistributionConfig:
        Origins:
          - DomainName: app-load-balancer.example.com
            Id: Example-ALB
            CustomOriginConfig:
              OriginProtocolPolicy: https-only
              OriginSSLProtocols:
                - TLSv1.2
            OriginCustomHeaders:
               - HeaderName: X-Custom-Header
                 HeaderValue: random-value-1234567890
        Enabled: 'true'
        DefaultCacheBehavior:
          TargetOriginId: Example-ALB
          ViewerProtocolPolicy: allow-all
          CachePolicyId: 658327ea-f89d-4fab-a63d-7e88639e58f6
        PriceClass: PriceClass_All
        ViewerCertificate:
          CloudFrontDefaultCertificate: 'true'
```

Weitere Informationen finden Sie unter Origin und OriginCustomHeader Eigenschaften im AWS CloudFormation Benutzerhandbuch.

Um einen benutzerdefinierten HTTP-Header (CloudFront API) hinzuzufügen

Verwenden Sie in der CloudFront API das darin `CustomHeaders` enthaltene Objekt`Origin`. Weitere Informationen finden Sie unter CreateDistribution und UpdateDistribution in der CloudFront Amazon-API-Referenz sowie in der Dokumentation für Ihr SDK oder einen anderen API-Client.

Es gibt einige Header-Namen, die Sie nicht als benutzerdefinierte Ursprungs-Header angeben können. Weitere Informationen finden Sie unter Benutzerdefinierte Header, die nicht zu CloudFront Quellanfragen hinzugefügt werden können.

Konfigurieren eines Application Load Balancers zum Weiterleiten von Anforderungen, die einen bestimmten Header enthalten

Nachdem Sie den Anforderungen CloudFront , die er an Ihren Application Load Balancer sendet, einen benutzerdefinierten HTTP-Header hinzugefügt haben (siehe vorheriger Abschnitt), können Sie den Load Balancer so konfigurieren, dass er nur Anfragen weiterleitet, die diesen benutzerdefinierten Header enthalten. Dazu fügen Sie eine neue Regel hinzu und ändern die Standardregel im Listener Ihres Load Balancers.

Voraussetzungen

Um die folgenden Verfahren verwenden zu können, benötigen Sie einen Application Load Balancer mit mindestens einem Listener. Wenn Sie noch keinen erstellt haben, lesen Sie Create an Application Load Balancer im Benutzerhandbuch für Application Load Balancer.

Die folgenden Verfahren ändern einen HTTPS-Listener. Sie können den gleichen Prozess verwenden, um einen HTTP-Listener zu ändern.

So aktualisieren Sie die Regeln in einem Application Load Balancer-Listener

1. Fügen Sie eine neue Regel hinzu. Folgen Sie den Anleitungen unter Regel hinzufügen mit den folgenden Änderungen:

   - Fügen Sie die Regel dem Load Balancer hinzu, der der Ursprung Ihrer Distribution ist. CloudFront
   - Wählen Sie für Bedingung hinzufügen die Option HTTP-Header aus. Geben Sie den Namen und den Wert des HTTP-Headers an, den Sie als benutzerdefinierten Quell-Header hinzugefügt haben. CloudFront

- Wählen Sie für Aktion hinzufügen die Option Weiterleiten an aus. Wählen Sie die Zielgruppe aus, in die Sie Anfragen weiterleiten möchten.

2. Bearbeiten Sie die Standardregel im Listener Ihres Load Balancers. Folgen Sie den Anleitungen unter Bearbeiten einer Regel mit den folgenden Änderungen:

   - Bearbeiten Sie die Standardregel des Load Balancers, der der Ursprung Ihrer CloudFront Distribution ist.
   - Löschen Sie die Standardaktion und wählen Sie dann für Aktion hinzufügen die Option Feststehende Antwort zurückgeben aus.
   - Geben Sie für den Antwortcode **`403`** ein.
   - Geben Sie für den Antworttext **`Access denied`** ein.

Nachdem Sie diese Schritte ausgeführt haben, hat Ihr Load Balancer-Listener zwei Regeln. Eine Regel leitet Anfragen weiter, die den HTTP-Header enthalten (Anfragen, die von CloudFront stammen). Die andere Regel sendet eine feste Antwort auf alle anderen Anfragen (Anfragen, von denen nicht stammt CloudFront).

Sie können überprüfen, ob die Lösung funktioniert, indem Sie eine Anfrage an Ihre CloudFront Distribution und eine an Ihren Application Load Balancer senden. Die Anfrage an sendet Ihre Webanwendung oder Ihren Inhalt CloudFront zurück, und die Anfrage, die direkt an Ihren Application Load Balancer gesendet wird, gibt eine `403` Antwort mit der Klartextnachricht zurück. `Access denied`

(Optional) Verbesserung der Sicherheit dieser Lösung

Um die Sicherheit dieser Lösung zu verbessern, können Sie Ihre CloudFront Distribution so konfigurieren, dass beim Senden von Anfragen an Ihren Application Load Balancer immer HTTPS verwendet wird. Denken Sie daran, dass diese Lösung nur funktioniert, wenn Sie den benutzerdefinierten Header-Namen und den Wert geheim halten. Die Verwendung von HTTPS kann dazu beitragen, dass ein Spion den Namen und den Wert des Headers entdeckt. Wir empfehlen außerdem, den Namen und den Wert des Headers regelmäßig zu wechseln.

Verwenden von HTTPS für Origin-Anfragen

Um die Verwendung von HTTPS für Quellanfragen CloudFront zu konfigurieren, setzen Sie die Origin Protocol Policy-Einstellung auf Nur HTTPS. Diese Einstellung ist in der CloudFront Konsole

und in der CloudFront API verfügbar. CloudFormation Weitere Informationen finden Sie unter Protokoll (nur benutzerdefinierte Ursprünge).

Folgendes gilt auch, wenn Sie CloudFront die Verwendung von HTTPS für Quellanfragen konfigurieren:

- Sie CloudFront müssen so konfigurieren, dass der `Host` Header mit der Quellanforderungsrichtlinie an den Ursprung weitergeleitet wird. Sie können die Richtlinie für AllViewer verwaltete Ursprungsanfragen verwenden.
- Stellen Sie sicher, dass Ihr Application Load Balancer über einen HTTPS-Listener verfügt (wie im vorherigen Abschnitt gezeigt). Weitere Informationen finden Sie unter Erstellen eines HTTPS-Listeners im Benutzerhandbuch für Application Load Balancers. Für die Verwendung eines HTTPS-Listeners benötigen Sie ein SSL/TLS Zertifikat, das dem Domainnamen entspricht, der an Ihren Application Load Balancer weitergeleitet wird.
- SSL/TLS Zertifikate für CloudFront können nur im In (ACM) angefordert (oder importiert) werden. `us-east-1` AWS-Region AWS Certificate Manager Da es CloudFront sich um einen globalen Dienst handelt, verteilt er das Zertifikat automatisch aus der `us-east-1` Region an alle Regionen, die Ihrer CloudFront Distribution zugeordnet sind.
  - Wenn Sie beispielsweise einen Application Load Balancer (ALB) in der `ap-southeast-2` Region haben, müssen Sie SSL/TLS Zertifikate sowohl in der `ap-southeast-2` Region (für die Verwendung von HTTPS zwischen CloudFront und dem ALB-Ursprung) als auch in der `us-east-1` Region (für die Verwendung von HTTPS zwischen Zuschauern und) konfigurieren. CloudFront Beide Zertifikate sollten dem Domainnamen entsprechen, der an Ihren Application Load Balancer weitergeleitet wird. Weitere Informationen finden Sie unter AWS-Region for AWS Certificate Manager.
- Wenn die Endbenutzer (auch als Viewer oder Kunden bezeichnet) Ihrer Webanwendung HTTPS verwenden können, können Sie auch so konfigurieren CloudFront , dass HTTPS-Verbindungen von Endbenutzern bevorzugt werden (oder sogar erforderlich sind). Verwenden Sie dazu die Einstellung Betrachter-Protokollrichtlinie. Sie können es so einstellen, dass Endbenutzer von HTTP auf HTTPS umgeleitet oder Anfragen, die HTTP verwenden, abgelehnt werden. Diese Einstellung ist in der CloudFront Konsole und in der CloudFront API verfügbar. CloudFormation Weitere Informationen finden Sie unter Viewer-Protokollrichtlinien.

Wechseln des Header-Namens und des Werts

Zusätzlich zur Verwendung von HTTPS empfehlen wir auch, den Header-Namen und -Wert regelmäßig zu ändern. Befolgen Sie hierfür die folgenden Schritte:

1. Konfigurieren Sie CloudFront die Konfiguration so, dass den Anforderungen, die an den Application Load Balancer gesendet werden, ein zusätzlicher benutzerdefinierter HTTP-Header hinzugefügt wird.
2. Aktualisieren Sie die Application Load Balancer-Listener-Regel, um Anforderungen weiterzuleiten, die diesen zusätzlichen benutzerdefinierten HTTP-Header enthalten.
3. Konfigurieren Sie CloudFront die Konfiguration so, dass der ursprüngliche benutzerdefinierte HTTP-Header nicht mehr zu Anfragen hinzugefügt wird, die an den Application Load Balancer gesendet werden.
4. Aktualisieren Sie die Application Load Balancer-Listener-Regel, um die Weiterleitung von Anforderungen zu beenden, die den ursprünglichen benutzerdefinierten HTTP-Header enthalten.

Weitere Informationen zum Ausführen dieser Schritte finden Sie in den vorherigen Abschnitten.

(Optional) Beschränken Sie den Zugriff auf Origin mithilfe der AWS-verwaltete Präfixliste für CloudFront

Um den Zugriff auf Ihren Application Load Balancer weiter einzuschränken, können Sie die dem Application Load Balancer zugeordnete Sicherheitsgruppe so konfigurieren, dass sie nur Datenverkehr akzeptiert, CloudFront wenn der Dienst eine verwaltete Präfixliste verwendet AWS. Dadurch wird verhindert, dass Datenverkehr, der nicht von CloudFront Ihnen stammt, Ihren Application Load Balancer auf der Netzwerkebene (Ebene 3) oder Transportebene (Schicht 4) erreicht.

Weitere Informationen finden Sie im CloudFront Blogbeitrag Beschränken Sie den Zugriff auf Ihre Ursprünge mithilfe AWS der von Amazon verwalteten Präfixliste.

Einschränken der geografischen Distribution Ihrer Inhalte

Sie können geografische Einschränkungen, manchmal auch als Geoblocking bezeichnet, verwenden, um zu verhindern, dass Benutzer an bestimmten geografischen Standorten auf Inhalte zugreifen, die Sie über eine Amazon-Distribution verteilen. CloudFront Es gibt zwei Möglichkeiten zum Verwenden der geografischen Einschränkung:

- Verwenden Sie die Funktion für CloudFront geografische Einschränkungen. Verwenden Sie diese Option, um den Zugriff auf alle mit einer Distribution verknüpften Dateien zu beschränken und um den Zugriff auf der Länderebene einzuschränken.

- Verwenden Sie den Geolokalisierungsdienst eines Drittanbieters. Verwenden Sie diese Option, um den Zugriff auf einen Teil der mit einer Distribution verknüpften Dateien zu beschränken oder um den Zugriff auf eine feinere Granularität als die Länderebene einzuschränken.

Themen

- Verwenden Sie CloudFront geografische Einschränkungen
- Verwenden des Geolokalisierungsdienstes eines Drittanbieters

Verwenden Sie CloudFront geografische Einschränkungen

Wenn ein Benutzer Ihre Inhalte anfordert, werden die angeforderten Inhalte in der CloudFront Regel unabhängig davon bereitgestellt, wo sich der Benutzer befindet. Wenn Sie verhindern möchten, dass Benutzer in bestimmten Ländern auf Ihre Inhalte zugreifen, können Sie die Funktion für CloudFront geografische Einschränkungen verwenden, um eine der folgenden Aktionen durchzuführen:

- Gewähren Sie Ihren Benutzern den Zugriff auf Ihre Inhalte nur, wenn diese sich in einem der zugelassenen Länder auf Ihrer Zulassungsliste befinden.
- Verhindern Sie, dass Benutzer auf Ihre Inhalte zugreifen, wenn sie sich in einem der gesperrten Länder auf Ihrer Deny-Liste befinden.

Wenn eine Anfrage beispielsweise aus einem Land kommt, in dem Sie nicht zur Verbreitung Ihrer Inhalte berechtigt sind, können Sie die Anfrage mithilfe CloudFront geografischer Einschränkungen blockieren.

> **Note**
>
> CloudFront bestimmt den Standort Ihrer Benutzer mithilfe einer Datenbank eines Drittanbieters. Die Genauigkeit der Zuweisung zwischen IP-Adressen und Ländern variiert je nach Region. Gemäß kürzlich erfolgten Tests beträgt die allgemeine Genauigkeit 99,8 %. Wenn der Standort eines Benutzers nicht ermittelt werden CloudFront kann, CloudFront wird der Inhalt bereitgestellt, den der Benutzer angefordert hat.

So funktioniert die geografische Einschränkung:

1. Nehmen wir an, Sie haben nur Rechte zum Verteilen Ihrer Inhalte in Liechtenstein. Sie aktualisieren Ihre CloudFront Distribution, um eine Zulassungsliste hinzuzufügen, die nur Liechtenstein enthält. (Alternativ können Sie eine Deny-Liste mit allen Ländern außer Liechtenstein hinzufügen.)
2. Ein Benutzer in Monaco fordert Ihre Inhalte an, und DNS leitet die Anfrage an einen CloudFront Edge-Standort in Mailand, Italien, weiter.
3. Der Edge-Standort in Mailand sucht nach Ihrer Distribution und ermittelt, dass der Benutzer in Monaco nicht zum Herunterladen Ihrer Inhalte berechtigt ist.
4. CloudFront gibt einen HTTP-Statuscode `403 (Forbidden)` an den Benutzer zurück.

> **Note**
>
> Geografische Einschränkungen gelten nicht für Anfragen zur Zertifikatsvalidierung für CloudFront verwaltete Zertifikate. Wenn Sie ein CloudFront verwaltetes Zertifikat verwenden, sind Anfragen an den `/.well-known/pki-validation/` Pfad von den Regeln für geografische Beschränkungen ausgeschlossen. Dadurch wird sichergestellt, dass die Zertifizierungsstellen Ihr Zertifikat unabhängig von Ihrer Konfiguration der geografischen Einschränkungen aus den verschiedenen geografisch verteilten Netzwerkperspektiven validieren können, die für die Ausstellung und Verlängerung erforderlich sind.
> Wenn Sie es AWS WAF mit Ihrer Distribution verwenden, stellen Sie sicher, dass Ihre AWS WAF Regeln (einschließlich Bot-Kontrollregeln) Anfragen an den `/.well-known/pki-validation/` Pfad nicht blockieren. Wenn Sie diese Anfragen blockieren, kann dies dazu führen, dass die Zertifikatsvalidierung nicht erfolgreich abgeschlossen wird.

Sie können optional so konfigurieren CloudFront , dass eine benutzerdefinierte Fehlermeldung an den Benutzer zurückgegeben wird, und Sie können angeben, wie lange die Fehlerantwort für die angeforderte Datei zwischengespeichert werden soll CloudFront . Der Standardwert liegt bei 10 Sekunden. Weitere Informationen finden Sie unter Erstellen einer benutzerdefinierten Fehlerseite für bestimmte HTTP-Statuscodes.

Geografische Einschränkungen gelten für eine gesamte Distribution. Wenn Sie eine Einschränkung auf einen Teil Ihrer Inhalte und eine andere Einschränkung (oder keine Beschränkung) auf einen anderen Teil Ihrer Inhalte anwenden müssen, müssen Sie separate CloudFront Verteilungen erstellen oder einen Geolokalisierungsdienst eines Drittanbieters verwenden.

Wenn Sie CloudFront Standardprotokolle (Zugriffsprotokolle) aktivieren, können Sie die CloudFront abgelehnten Anfragen identifizieren, indem Sie nach den Protokolleinträgen suchen, in denen der Wert von `sc-status` (der HTTP-Statuscode) steht. `403` Wenn Sie jedoch nur die Standardprotokolle verwenden, können Sie eine Anfrage, die aufgrund des Standorts des Benutzers CloudFront abgelehnt wurde, nicht von einer Anfrage unterscheiden, die CloudFront abgelehnt wurde, weil der Benutzer aus einem anderen Grund nicht berechtigt war, auf die Datei zuzugreifen. Wenn Sie einen Geolokalisierungsdienst eines Drittanbieters wie Digital Element oder MaxMind verwenden, können Sie den Standort der Anfragen anhand der IP-Adresse in der Spalte `c-ip` (Client-IP) in den Zugriffsprotokollen identifizieren. Weitere Informationen zu CloudFront Standardprotokollen finden Sie unter. Zugriffsprotokolle (Standardprotokolle)

Das folgende Verfahren erklärt, wie Sie mithilfe der CloudFront Konsole geografische Einschränkungen zu einer vorhandenen Distribution hinzufügen. Informationen zur Verwendung der Konsole zum Erstellen von Distributionen finden Sie unter Eine Distribution erstellen.

So fügen Sie Ihrer CloudFront Web-Distribution geografische Einschränkungen hinzu (Konsole)

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home.
2. Wählen Sie im Navigationsbereich die Option Distributionen und dann die Option, die Sie aktualisieren möchten.
3. Wählen Sie die Registerkarte Sicherheit und anschließend Geografische Einschränkungen aus.
4. Wählen Sie Edit (Bearbeiten) aus.
5. Wählen Sie Allow list (Genehmigungsliste) aus, um eine Liste der zulässigen Länder zu erstellen, oder Block list (Blockliste), um eine Liste der gesperrten Länder zu erstellen.
6. Fügen Sie die gewünschten Länder zur Liste hinzu und wählen Sie dann Save changes (Änderungen speichern) aus.

Verwenden des Geolokalisierungsdienstes eines Drittanbieters

Mit der Funktion für CloudFront geografische Einschränkungen steuern Sie die Verteilung Ihrer Inhalte auf Landesebene für alle Dateien, die Sie mit einer bestimmten Webdistribution verteilen. Wenn Sie einen Anwendungsfall für geografische Einschränkungen haben, bei dem die Einschränkungen nicht den Landesgrenzen folgen, oder wenn Sie den Zugriff nur auf einige der Dateien einschränken möchten, die Sie von einer bestimmten Distribution bereitstellen, können Sie dies CloudFront mit einem Geolokalisierungsdienst eines Drittanbieters kombinieren. Dies ermöglicht

Ihnen die Kontrolle des Zugriffs auf Ihre Inhalte nicht nur auf der Grundlage des Landes, sondern auch auf der Grundlage der Stadt, der Postleitzahl oder sogar der Längen- und Breitengrade.

Wenn Sie einen Geolokalisierungsdienst eines Drittanbieters verwenden, empfehlen wir Ihnen, CloudFront signierte URLs zu verwenden, mit denen Sie ein Ablaufdatum und eine Uhrzeit angeben können, nach deren Ablauf die URL nicht mehr gültig ist. Darüber hinaus empfehlen wir, dass Sie einen Amazon S3-Bucket als Ihren Ursprung verwenden, da Sie dann eine Quellzugriffskontrolle verwenden können, um zu verhindern, dass Benutzer direkt vom CloudFront Ursprung aus auf Ihre Inhalte zugreifen. Weitere Informationen zu signierten URLs sowie zur Ursprungszugriffssteuerung finden Sie unter Bereitstellen privater Inhalte mit signierten URLs und signierten Cookies.

In den folgenden Schritten wird erläutert, wie Sie den Zugriff auf Ihre Dateien mithilfe eines Geolokalisierungsservices eines Drittanbieters kontrollieren können.

Um den Zugriff auf Dateien in einer Distribution einzuschränken, verwenden Sie einen Geolocation-Service eines Drittanbieters CloudFront

1. Legen Sie ein Konto bei einem Geolokalisierungsdienst an.
2. Laden Sie Ihre Inhalte in einen Amazon-S3-Bucket hoch.
3. Konfigurieren Sie Amazon CloudFront und Amazon S3 für die Bereitstellung privater Inhalte. Weitere Informationen finden Sie unter Bereitstellen privater Inhalte mit signierten URLs und signierten Cookies.
4. Schreiben Sie Ihre Webanwendung so, dass folgende Aufgaben ausgeführt werden:

   - Senden der IP-Adresse für jede Benutzeranfrage an den Geolokalisierungsdienst.
   - Bewerten Sie den Rückgabewert des Geolocation-Service, um festzustellen, ob sich der Benutzer an einem Ort befindet, an dem Sie Ihre CloudFront Inhalte verteilen möchten.
   - Wenn Sie Ihre Inhalte an den Standort des Benutzers verteilen möchten, generieren Sie eine signierte URL für Ihre CloudFront Inhalte. Wenn Sie keinen Inhalt an diesen Speicherort verteilen möchten, geben Sie dem Benutzer den HTTP-Statuscode `403 (Forbidden)` zurück. Alternativ können Sie so konfigurieren CloudFront , dass eine benutzerdefinierte Fehlermeldung zurückgegeben wird. Weitere Informationen finden Sie unter the section called “Erstellen einer benutzerdefinierten Fehlerseite für bestimmte HTTP-Statuscodes”.

   Weitere Informationen finden Sie in der Dokumentation für den Geolokationsdienst, den Sie verwenden.

Sie können die IP-Adressen der Besucher Ihrer Website mithilfe einer Webserver-Variablen abrufen. Beachten Sie folgende Einschränkungen:

- Wenn Ihr Webserver nicht über einen Load Balancer mit dem Internet verbunden ist, können Sie eine Webserver-Variable verwenden, um die IP-Remote-Adresse abzurufen. Diese IP-Adresse ist jedoch nicht immer die IP-Adresse des Benutzers. Es kann sich dabei auch um die IP-Adresse eines Proxy-Servers handeln, je nachdem, wie der Benutzer mit dem Internet verbunden ist.
- Wenn Ihr Webserver über einen Load Balancer mit dem Internet verbunden ist, enthält eine Webserver-Variable möglicherweise die IP-Adresse des Load Balancers, nicht die IP-Adresse des Benutzers. Bei dieser Konfiguration empfehlen wir, die letzte IP-Adresse im `X-Forwarded-For`-HTTP-Header zu verwenden. Dieser Header enthält in der Regel mehr als eine IP-Adresse, von denen die meisten für Proxys oder Load Balancer gelten. Die letzte IP-Adresse in der Liste wird am wahrscheinlichsten mit dem geografischen Standort des Benutzers verknüpft.

Wenn Ihr Webserver nicht mit einem Load Balancer verbunden ist, empfehlen wir, anstelle des `X-Forwarded-For`-Headers Webserver-Variablen zu verwenden, um Spoofing von IP-Adressen zu vermeiden.

Vertrauliche Daten durch Verschlüsselung auf Feldebene schützen

Mit Amazon können Sie CloudFront mithilfe von HTTPS sichere Ende-zu-Ende-Verbindungen zu den Ursprungsservern durchsetzen. Field-level Die Verschlüsselung bietet eine zusätzliche Sicherheitsebene, mit der Sie bestimmte Daten während der gesamten Systemverarbeitung schützen können, sodass nur bestimmte Anwendungen sie sehen können.

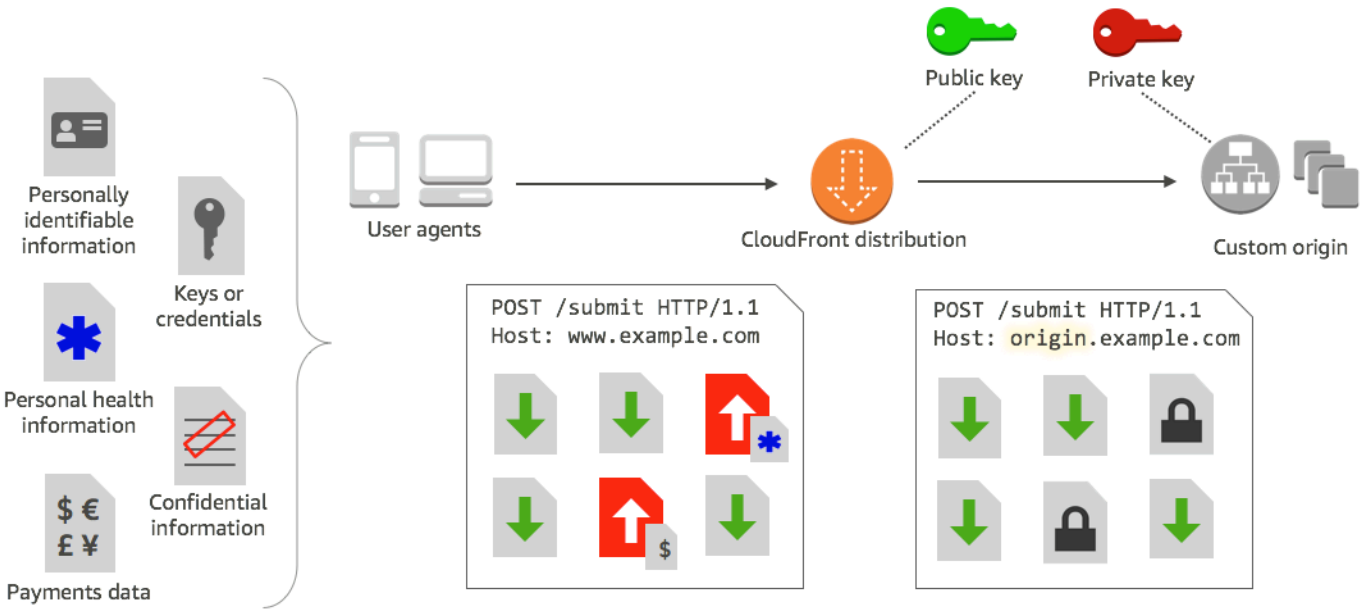
Mit der Verschlüsselung auf Feldebene können Sie es Ihren Benutzern ermöglichen, vertrauliche Informationen sicher auf Ihre Webserver hochzuladen. Die vertraulichen Informationen, die von Ihren Benutzern bereitgestellt werden, werden am Rand, in der Nähe des Benutzers, verschlüsselt und bleiben über den gesamten Anwendungs-Stack hinweg verschlüsselt. Diese Verschlüsselung stellt sicher, dass nur Anwendungen, die die Daten benötigen — und über die Anmeldeinformationen zu ihrer Entschlüsselung verfügen — dazu in der Lage sind.

Um die Verschlüsselung auf Feldebene zu verwenden, geben Sie bei der Konfiguration Ihrer CloudFront Distribution die Gruppe von Feldern in POST-Anforderungen an, die verschlüsselt werden sollen, sowie den öffentlichen Schlüssel, mit dem sie verschlüsselt werden sollen. Sie können bis zu 10 Datenfelder in einer Anfrage verschlüsseln. (Sie können nicht alle Daten in einer

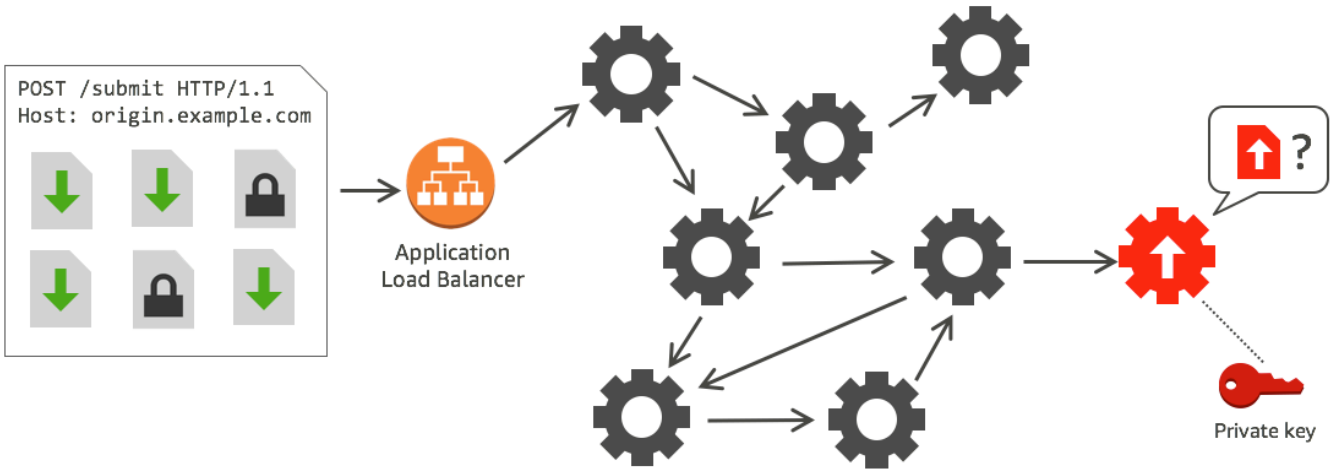
Anfrage mit Verschlüsselung auf Feldebene verschlüsseln; Sie müssen einzelne Felder angeben, die verschlüsselt werden sollen.)

Wenn die HTTPS-Anfrage mit Verschlüsselung auf Feldebene an den Ursprung weitergeleitet und durch Ihre Ursprungsanwendung bzw. Ihr Ursprungssubsystem geleitet wird, sind die vertraulichen Daten immer noch verschlüsselt, wodurch das Risiko einer Datenschutzverletzung oder eines versehentlichen Datenverlustes der vertraulichen Daten verringert wird. Komponenten, die aus geschäftlichen Gründen Zugriff auf die sensiblen Daten benötigen, wie z. B. ein Zahlungssystem für eine Kreditnummer, können mit dem entsprechenden privaten Schlüssel entschlüsseln und auf die Daten zugreifen.

> **Note**
>
> Beachten Sie, dass der Ursprung für die Verschlüsselung auf Feldebene die Chunked-Codierung unterstützen muss.



CloudFront Bei der Verschlüsselung auf Feldebene wird eine asymmetrische Verschlüsselung verwendet, die auch als Verschlüsselung mit öffentlichen Schlüsseln bezeichnet wird. Sie geben einen öffentlichen Schlüssel für an CloudFront, und alle von Ihnen angegebenen vertraulichen Daten werden automatisch verschlüsselt. Der von Ihnen angegebene Schlüssel CloudFront kann nicht zum Entschlüsseln der verschlüsselten Werte verwendet werden. Das kann nur Ihr privater Schlüssel.



Themen

- Überblick über die Verschlüsselung auf Feldebene
- Einrichten der Verschlüsselung auf Feldebene
- Entschlüsseln von Datenfeldern an Ihrem Ursprung

Überblick über die Verschlüsselung auf Feldebene

Die folgenden Schritte geben einen Überblick über die Einrichtung der Verschlüsselung auf Feldebene. Spezifische Schritte finden Sie unter Einrichten der Verschlüsselung auf Feldebene.

1. Holen Sie sich ein Schlüsselpaar aus öffentlichem und privatem Schlüssel. Sie müssen den öffentlichen Schlüssel abrufen und hinzufügen, bevor Sie mit der Einrichtung der Verschlüsselung auf Feldebene in CloudFront beginnen.
2. Erstellen Sie ein Verschlüsselungsprofil auf Feldebene. Field-level Verschlüsselungsprofile, die Sie in erstellen CloudFront, definieren die Felder, die verschlüsselt werden sollen.
3. Erstellen Sie eine Verschlüsselungskonfiguration auf Feldebene. Eine Konfiguration spezifiziert die zu verwendenden Profile, basierend auf dem Inhaltstyp der Anfrage oder einem Abfrageargument, um bestimmte Datenfelder zu verschlüsseln. Sie können auch die Verhaltensoptionen für die Anforderungsweiterleitung auswählen, die Sie für verschiedene Szenarien benötigen. Sie können beispielsweise das Verhalten festlegen, wenn der durch das Abfrageargument in einer Anforderungs-URL angegebene Profilname in nicht existiert CloudFront.

4. Verknüpfen zu einem Cache-Verhalten. Verknüpfen Sie die Konfiguration mit einem Cache-Verhalten für eine Verteilung, um anzugeben, wann CloudFront Daten verschlüsseln soll.

Einrichten der Verschlüsselung auf Feldebene

Führen Sie die folgenden Schritte aus, um mit der Verschlüsselung auf Feldebene zu beginnen. Weitere Informationen zu Kontingenten (früher als Limits bezeichnet) für die Verschlüsselung auf Feldebene finden Sie unter Kontingente.

- Schritt 1: Erstellen eines RSA-Schlüsselpaars
- Schritt 2: Fügen Sie Ihren öffentlichen Schlüssel hinzu CloudFront
- Schritt 3: Erstellen eines Profils für die Verschlüsselung auf Feldebene
- Schritt 4: Erstellen einer Konfiguration
- Schritt 5: Hinzufügen einer Konfiguration zu einem Cache-Verhalten

Schritt 1: Erstellen eines RSA-Schlüsselpaars

Um zu beginnen, müssen Sie ein RSA-Schlüsselpaar erstellen, das einen öffentlichen Schlüssel und einen privaten Schlüssel enthält. Der öffentliche Schlüssel CloudFront ermöglicht das Verschlüsseln von Daten, und der private Schlüssel ermöglicht es Komponenten an Ihrem Ursprung, die verschlüsselten Felder zu entschlüsseln. Sie können OpenSSL oder ein anderes Tool verwenden, um ein Schlüsselpaar zu erstellen. Die Schlüsselgröße muss 2048 Bit betragen.

Wenn Sie beispielsweise OpenSSL verwenden, können Sie mithilfe des folgenden Befehls ein Schlüsselpaar mit einer Länge von 2048 Bits erstellen und in der Datei `private_key.pem` speichern:

```
openssl genrsa -out private_key.pem 2048
```

Die erstellte Datei enthält den öffentlichen und den privaten Schlüssel. Um den öffentlichen Schlüssel aus dieser Datei zu extrahieren, führen Sie den folgenden Befehl aus:

```
openssl rsa -pubout -in private_key.pem -out public_key.pem
```

Die Datei mit dem öffentlichen Schlüssel (`public_key.pem`) enthält den codierten Schlüsselwert, den Sie im folgenden Schritt einfügen.

Schritt 2: Fügen Sie Ihren öffentlichen Schlüssel hinzu CloudFront

Nachdem Sie Ihr RSA-Schlüsselpaar erhalten haben, fügen Sie Ihren öffentlichen Schlüssel zu CloudFront hinzu.

So fügen Sie Ihren öffentlichen Schlüssel zu CloudFront (Konsole) hinzu

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home.
2. Wählen Sie im Navigationsbereich Public key aus.
3. Wählen Sie Add public key (Öffentlichen Schlüssel hinzufügen) aus.
4. Geben Sie unter Key name einen eindeutigen Namen für den Schlüssel ein. Der Name darf keine Leerzeichen enthalten und darf nur alphanumerische Zeichen, Unterstriche (_) und Bindestriche (-) enthalten. Die maximale Anzahl der Zeichen beträgt 128.
5. Fügen Sie unter Key value (Schlüsselwert) den codierten Schlüsselwert für den öffentlichen Schlüssel ein, einschließlich der Zeilen `-----BEGIN PUBLIC KEY-----` und `-----END PUBLIC KEY-----`.
6. In Comment fügen Sie einen optionalen Kommentar hinzu. Beispielsweise können Sie das Ablaufdatum für den öffentlichen Schlüssel angeben.
7. Wählen Sie Hinzufügen aus.

Sie können weitere Schlüssel zur Verwendung hinzufügen, CloudFront indem Sie die Schritte des Verfahrens wiederholen.

Schritt 3: Erstellen eines Profils für die Verschlüsselung auf Feldebene

Nachdem Sie mindestens einen öffentlichen Schlüssel hinzugefügt haben CloudFront, erstellen Sie ein Profil, das angibt, CloudFront welche Felder verschlüsselt werden sollen.

So erstellen Sie ein Profil für die Verschlüsselung auf Feldebene (Konsole)

1. Wählen Sie im Navigationsbereich Field-level Verschlüsselung aus.
2. Wählen Sie Create profile (Profil erstellen) aus.
3. Füllen Sie die folgenden Felder aus:

Profilname

Geben Sie einen eindeutigen Namen für das Profil ein. Der Name darf keine Leerzeichen enthalten und darf nur alphanumerische Zeichen, Unterstriche (_) und Bindestriche (-) enthalten. Die maximale Anzahl der Zeichen beträgt 128.

Public key name

Wählen Sie in der Dropdownliste den Namen eines öffentlichen Schlüssels aus, den Sie CloudFront in Schritt 2 hinzugefügt haben. CloudFront verwendet den Schlüssel, um die Felder zu verschlüsseln, die Sie in diesem Profil angeben.

Name des Anbieters

Geben Sie einen Satz ein, um den Schlüssel zu identifizieren (z. B. den Anbieter, bei dem Sie das Schlüsselpaar erhalten haben). Diese Informationen werden zusammen mit dem privaten Schlüssel benötigt, wenn Anwendungen Datenfelder entschlüsseln. Der Name des Anbieters darf keine Leerzeichen enthalten und darf nur alphanumerische Zeichen, Doppelpunkte (:), Unterstriche (_) und Bindestriche (-) enthalten. Die maximale Anzahl der Zeichen beträgt 128.

Feldnamensmuster für die Zuordnung

Geben Sie die Namen der Datenfelder oder Muster ein, die Datenfeldnamen in der Anfrage identifizieren, die Sie mit CloudFront verschlüsseln möchten. Wählen Sie die Option +, um alle Felder hinzuzufügen, die Sie mit diesem Schlüssel verschlüsseln möchten.

Für das Feldnamenmuster können Sie den gesamten Namen des Datenfeldes eingeben DateOfBirth, z. B. oder nur den ersten Teil des Namens mit einem Platzhalterzeichen (*), z. B. CreditCard *. Das Feldnamensmuster darf neben dem optionalen Platzhalterzeichen (*) nur aus alphanumerischen Zeichen, eckigen Klammern ([und]), Punkten (.), Unterstrichen (_) und Bindestrichen (-) bestehen.

Stellen Sie sicher, dass Sie keine überschneidenden Zeichen für unterschiedliche Feldnamensmuster verwenden. Wenn Sie beispielsweise ein Feldnamensmuster von ABC* haben, können Sie kein weiteres Feldnamensmuster hinzufügen, das AB* ist. Dazu kommt, dass bei Feldnamen zwischen Groß- und Kleinschreibung unterschieden wird und die maximale Anzahl der Zeichen 128 beträgt.

Kommentar

(Optional) Geben Sie einen Kommentar zu diesem Profil ein. Die maximale Anzahl der Zeichen beträgt 128.

4. Nachdem Sie die Felder ausgefüllt haben, wählen Sie Create profile (Profil erstellen) aus.
5. Wenn Sie weitere Profile hinzufügen möchten, wählen Sie Add profile aus.

Schritt 4: Erstellen einer Konfiguration

Nachdem Sie ein oder mehrere Verschlüsselungsprofile auf Feldebene erstellt haben, erstellen Sie eine Konfiguration, die den Inhaltstyp der Anforderung angibt, die die zu verschlüsselnden Daten enthält, das Profil, das für die Verschlüsselung verwendet werden soll, und andere Optionen, die angeben, wie die Verschlüsselung gehandhabt werden CloudFront soll.

Wenn die Daten beispielsweise nicht verschlüsselt CloudFront werden können, können Sie in den folgenden Szenarien angeben, ob eine Anfrage an Ihren Ursprung blockiert oder weitergeleitet werden CloudFront soll:

- Wenn sich der Inhaltstyp einer Anfrage nicht in einer Konfiguration befindet — Möglicherweise haben Sie einer Konfiguration keinen Inhaltstyp hinzugefügt. In diesem Fall können Sie angeben, ob die Anfrage mit diesem Inhaltstyp an den Ursprung CloudFront weitergeleitet wird, ohne die Datenfelder zu verschlüsseln, oder ob die Anfrage blockiert und ein Fehler zurückgegeben wird.

  > **Note**
  >
  > Wenn Sie einer Konfiguration einen Inhaltstyp hinzufügen, aber kein Profil für diesen Typ angegeben haben, werden Anfragen mit diesem Inhaltstyp CloudFront immer an den Ursprung weitergeleitet.

- Wenn der in einem Abfrageargument angegebene Profilname unbekannt ist — Wenn Sie das `fle-profile` Abfrageargument mit einem Profilnamen angeben, der für Ihre Distribution nicht existiert, können Sie angeben, ob die Anforderung ohne Verschlüsselung der Datenfelder an den Ursprung gesendet werden CloudFront soll, oder ob die Anforderung blockiert und ein Fehler zurückgegeben werden soll.

In einer Konfiguration können Sie auch angeben, ob die Bereitstellung eines Profils als Abfrageargument in einer URL ein Profil überschreibt, das Sie dem Inhaltstyp für diese Abfrage

zugeordnet haben. CloudFront Verwendet standardmäßig das Profil, das Sie einem Inhaltstyp zugeordnet haben, sofern Sie einen angeben. Damit können Sie ein Profil verwenden, das standardmäßig verwendet wird, aber für bestimmte Anfragen entscheiden, dass Sie ein anderes Profil erzwingen möchten.

So können Sie z. B. (in Ihrer Konfiguration) **`SampleProfile`** als das zu verwendende Abfrageargumentprofil festlegen. Dann könntest du `https://d1234.cloudfront.net?fle-profile=SampleProfile` statt des Profils`https://d1234.cloudfront.net`, das du **`SampleProfile`** für den Inhaltstyp der Anfrage eingerichtet hast, die URL CloudFront verwenden, um sie für diese Anfrage verwenden zu können.

Sie können bis zu 10 Konfigurationen für ein einzelnes Konto erstellen und dann eine der Konfigurationen dem Cache-Verhalten einer beliebigen Distribution für das Konto zuordnen.

So erstellen Sie eine Konfiguration für die Verschlüsselung auf Feldebene (Konsole)

1. Wählen Sie auf der Field-level Verschlüsselungsseite die Option Konfiguration erstellen aus.

   Hinweis: Wenn Sie nicht mindestens ein Profil erstellt haben, wird die Option zum Erstellen einer Konfiguration nicht angezeigt.
2. Füllen Sie die folgenden Felder aus, um das zu verwendende Profil anzugeben. (Einige Felder können nicht geändert werden.)

   Inhaltstyp (kann nicht geändert werden)

   Der Inhaltstyp ist auf `application/x-www-form-urlencoded` festgelegt und nicht änderbar.

   Default profile ID (optional)

   Wählen Sie in der Dropdown-Liste das Profil aus, das Sie dem Inhaltstyp im Feld Content type zuordnen möchten.

   Content-Format (kann nicht geändert werden)

   Das Content-Format ist auf `URLencoded` festgelegt und nicht änderbar.
3. Wenn Sie das CloudFront Standardverhalten für die folgenden Optionen ändern möchten, aktivieren Sie das entsprechende Kontrollkästchen.

Weiterleiten der Anfrage an den Ursprung, wenn der Inhaltstyp der Anfrage nicht konfiguriert ist.

Aktivieren Sie das Kontrollkästchen, wenn Sie zulassen möchten, dass die Anfrage an Ihren Ursprung geht, wenn Sie kein Profil für den Inhaltstyp der Anfrage angegeben haben.

Überschreiben des Profils für einen Inhaltstyp mit einem bereitgestellten Abfrageargument

Aktivieren Sie das Kontrollkästchen, wenn Sie zulassen möchten, dass ein in einem Abfrageargument angegebenes Profil das Profil überschreibt, das Sie für einen Inhaltstyp angegeben haben.

4. Wenn Sie das Kontrollkästchen aktivieren, damit ein Abfrageargument das Standardprofil überschreiben kann, müssen Sie die folgenden zusätzlichen Felder für die Konfiguration ausfüllen. Sie können bis zu fünf dieser Abfrageargumentzuordnungen für die Verwendung mit Abfragen erstellen.

   Query argument

   Geben Sie den Wert ein, den Sie für URLs für das `fle-profile`-Abfrageargument einbeziehen möchten. Dieser Wert zeigt CloudFront , dass die Profil-ID (die Sie im nächsten Feld angeben), die mit diesem Abfrageargument verknüpft ist, für die Verschlüsselung auf Feldebene für diese Abfrage verwendet werden soll.

   Die maximale Anzahl der Zeichen beträgt 128. Der Wert darf keine Leerzeichen enthalten und darf nur alphanumerische Zeichen oder die folgenden Zeichen enthalten: Bindestrich (-), Punkt (.), Unterstrich (_), Stern (*), Pluszeichen (+), Prozent (%).

   Profile ID

   Wählen Sie in der Dropdown-Liste das Profil aus, das Sie mit dem Wert verknüpfen möchten, den Sie für Query argument eingegeben haben.

   Weiterleiten der Anfrage an den Ursprung, wenn das in einem Abfrageargument angegebene Profil nicht existiert.

   Aktivieren Sie das Kontrollkästchen, wenn Sie zulassen möchten, dass die Anfrage zu Ihrem Ursprung geht, wenn das in einem Abfrageargument angegebene Profil nicht in CloudFront definiert ist.

Schritt 5: Hinzufügen einer Konfiguration zu einem Cache-Verhalten

Um eine Verschlüsselung auf Feldebene zu verwenden, verknüpfen Sie eine Konfiguration mit einem Cache-Verhalten für eine Distribution, indem Sie die Konfigurations-ID als Wert für Ihre Distribution hinzufügen.

> **Important**
>
> Um eine Verschlüsselungskonfiguration auf Feldebene mit einem Cacheverhalten zu verknüpfen, muss die Distribution so konfiguriert sein, dass sie immer HTTPS verwendet und HTTP `POST`- und `PUT`-Anforderungen von Viewern akzeptiert. D.h., eine der folgenden Bedingungen muss erfüllt sein:
>
> - Die Viewer Protocol Policy (Viewer-Protokollrichtlinie) des Cacheverhaltens muss Redirect HTTP to HTTPS (HTTP zu HTTPS umleiten) oder HTTPS only (Nur HTTPS) sein. (In CloudFormation oder der CloudFront API `ViewerProtocolPolicy` muss auf `redirect-to-https` oder gesetzt sein`https-only`.)
> - Die Allowed HTTP Methods (Erlaubte HTTP-Methoden) des Cache-Verhaltens müssen auf GET, HEAD, OPTIONS, PUT, POST, PATCH, DELETE gesetzt sein. (In CloudFormation oder der CloudFront API `AllowedMethods` muss auf,,`GET`,`HEAD`,`OPTIONS`, `PUT` `POST``PATCH`, gesetzt sein`DELETE`. Diese können in beliebiger Reihenfolge angegeben werden.)
> - Die Origin Protocol Policy (Ursprungsprotokollrichtlinie) der Ursprungseinstellungen muss auf Match Viewer (Übereinstimmung mit Viewer) oder HTTPS Only (Nur HTTPS) festgelegt sein. (In CloudFormation oder der CloudFront API `OriginProtocolPolicy` muss auf `match-viewer` oder gesetzt sein`https-only`.)

Weitere Informationen finden Sie unter Referenz für alle Distributionseinstellungen.

Entschlüsseln von Datenfeldern an Ihrem Ursprung

CloudFront verschlüsselt Datenfelder mithilfe von. AWS Encryption SDK Die Daten bleiben während des gesamten Anwendungs-Stacks verschlüsselt und können nur von Anwendungen abgerufen werden, die über die Anmeldeinformationen verfügen, um sie zu entschlüsseln.

Nach der Verschlüsselung wird der Verschlüsselungstext base64-kodiert. Wenn Ihre Anwendungen den Text am Ursprung entschlüsseln, müssen sie zuerst den Verschlüsselungstext entschlüsseln und dann das AWS Encryption SDK verwenden, um die Daten zu entschlüsseln.

Das folgende Codebeispiel veranschaulicht, wie Anwendungen Daten an Ihrem Ursprung entschlüsseln können. Beachten Sie Folgendes:

- Um das Beispiel zu vereinfachen, lädt dieses Beispiel öffentliche und private Schlüssel (im DER-Format) aus Dateien im Arbeitsverzeichnis. In der Praxis würden Sie den privaten Schlüssel an einem sicheren Offline-Platz, wie z. B. einem Offline-Hardware-Sicherheitsmodul, aufbewahren und den öffentlichen Schlüssel an Ihr Entwicklungsteam verteilen.
- CloudFront verwendet beim Verschlüsseln der Daten bestimmte Informationen, und für die Entschlüsselung sollte am Ursprung derselbe Satz von Parametern verwendet werden. Zu den Parametern, CloudFront die bei der Initialisierung verwendet werden, MasterKey gehören die folgenden:
  - PROVIDER_NAME: Sie haben diesen Wert beim Anlegen eines Verschlüsselungsprofils auf Feldebene angegeben. Verwenden Sie hier denselben Wert.
  - KEY_NAME: Sie haben einen Namen für Ihren öffentlichen Schlüssel erstellt, als Sie ihn hochgeladen haben CloudFront, und dann den Schlüsselnamen im Profil angegeben. Verwenden Sie hier denselben Wert.
  - ALGORITHMUS: CloudFront wird `RSA/ECB/OAEPWithSHA-256AndMGF1Padding` als Algorithmus für die Verschlüsselung verwendet, Sie müssen also denselben Algorithmus verwenden, um die Daten zu entschlüsseln.
- Wenn Sie das folgende Beispielprogramm mit dem Verschlüsselungstext als Eingabe ausführen, werden die entschlüsselten Daten in Ihrer Konsole ausgegeben. Weitere Informationen finden Sie im Java-Beispielcode im AWS Verschlüsselungs-SDK.

Beispiel-Code

```
import java.nio.file.Files;
import java.nio.file.Paths;
import java.security.KeyFactory;
import java.security.PrivateKey;
import java.security.PublicKey;
import java.security.spec.PKCS8EncodedKeySpec;
import java.security.spec.X509EncodedKeySpec;
```

```
import org.apache.commons.codec.binary.Base64;

import com.amazonaws.encryptionsdk.AwsCrypto;
import com.amazonaws.encryptionsdk.CryptoResult;
import com.amazonaws.encryptionsdk.jce.JceMasterKey;

/**
 * Sample example of decrypting data that has been encrypted by CloudFront field-level
 encryption.
 */
public class DecryptExample {

    private static final String PRIVATE_KEY_FILENAME = "private_key.der";
    private static final String PUBLIC_KEY_FILENAME = "public_key.der";
    private static PublicKey publicKey;
    private static PrivateKey privateKey;

    // CloudFront uses the following values to encrypt data, and your origin must use
 same values to decrypt it.
    // In your own code, for PROVIDER_NAME, use the provider name that you specified
 when you created your field-level
    // encryption profile. This sample uses 'DEMO' for the value.
    private static final String PROVIDER_NAME = "DEMO";
    // In your own code, use the key name that you specified when you added your public
 key to CloudFront. This sample
    // uses 'DEMOKEY' for the key name.
    private static final String KEY_NAME = "DEMOKEY";
    // CloudFront uses this algorithm when encrypting data.
    private static final String ALGORITHM = "RSA/ECB/OAEPWithSHA-256AndMGF1Padding";

    public static void main(final String[] args) throws Exception {

        final String dataToDecrypt = args[0];

        // This sample uses files to get public and private keys.
        // In practice, you should distribute the public key and save the private key
 in secure storage.
        populateKeyPair();

        System.out.println(decrypt(debase64(dataToDecrypt)));
    }

    private static String decrypt(final byte[] bytesToDecrypt) throws Exception {
```

```
        // You can decrypt the stream only by using the private key.

        // 1. Instantiate the SDK
        final AwsCrypto crypto = new AwsCrypto();

        // 2. Instantiate a JCE master key
        final JceMasterKey masterKey = JceMasterKey.getInstance(
                publicKey,
                privateKey,
                PROVIDER_NAME,
                KEY_NAME,
                ALGORITHM);

        // 3. Decrypt the data
        final CryptoResult <byte[], ? > result = crypto.decryptData(masterKey,
 bytesToDecrypt);
        return new String(result.getResult());
    }

    // Function to decode base64 cipher text.
    private static byte[] debase64(final String value) {
        return Base64.decodeBase64(value.getBytes());
    }

    private static void populateKeyPair() throws Exception {
        final byte[] PublicKeyBytes =
 Files.readAllBytes(Paths.get(PUBLIC_KEY_FILENAME));
        final byte[] privateKeyBytes =
 Files.readAllBytes(Paths.get(PRIVATE_KEY_FILENAME));
        publicKey = KeyFactory.getInstance("RSA").generatePublic(new
 X509EncodedKeySpec(PublicKeyBytes));
        privateKey = KeyFactory.getInstance("RSA").generatePrivate(new
 PKCS8EncodedKeySpec(privateKeyBytes));
    }
}
```

Video auf Abruf und Live-Streaming-Video mit CloudFront

Sie können CloudFront es verwenden, um Video on Demand (VOD) oder Live-Streaming-Video bereitzustellen, indem Sie einen beliebigen HTTP-Ursprung verwenden. Eine Möglichkeit, Video-Workflows in der Cloud einzurichten, besteht darin, sie CloudFront zusammen mit AWS Media Services zu verwenden.

Themen

- Informationen zum Streamen von Videos
- Stellen Sie Video auf Abruf bereit mit CloudFront
- Stellen Sie Videostreaming mit CloudFront und bereit AWS Mediendienste
- Media Quality-Aware Resiliency

Informationen zum Streamen von Videos

Sie müssen einen Encoder verwenden, um Videoinhalte zu verpacken, bevor CloudFront Sie die Inhalte verteilen können. Beim Paketerstellungsprozess werden Segmente erstellt, die Ihre Audio-, Video- und Untertitelinhalte enthalten. Es werden dabei auch Manifestdateien erzeugt, die in einer bestimmten Reihenfolge beschreiben, welche Segmente abgespielt werden sollen und wann. Gängige Paketformate sind MPEG DASH, Apple HLS, Microsoft Smooth Streaming und CMAF.

VOD-Streaming

Beim VOD-Streaming werden Ihre Videoinhalte auf einem Server gespeichert und können Viewern jederzeit angezeigt werden. Zum Erstellen einer Komponente (eines Assets), die von den Betrachtern gestreamt werden kann, verwenden Sie einen Encoder, z. B. AWS Elemental MediaConvert, um Ihre Mediendateien zu formatieren und zu verpacken.

Nachdem Ihr Video in die richtigen Formate gepackt wurde, können Sie es auf einem Server oder in einem Amazon S3-Bucket speichern und dann bereitstellen, wenn die Zuschauer es wünschen. CloudFront

Live-Video-Streaming

Beim Live-Video-Streaming werden Ihre Videoinhalte in Echtzeit gestreamt, wenn Live-Events stattfinden, oder als Rund-um-die-Uhr-Live-Kanal eingerichtet. Um Live-Ausgaben für Übertragung und Streaming zu erstellen, verwenden Sie einen Encoder, z. B. AWS Elemental MediaLive, um das Video zu komprimieren und für die Anzeige auf Geräten zu formatieren.

Nachdem Ihr Video codiert wurde, können Sie es in verschiedenen Übertragungsformaten speichern AWS Elemental MediaStore oder mithilfe von verwenden. AWS Elemental MediaPackage Verwende eine dieser Quellen, um eine CloudFront Distribution für die Bereitstellung der Inhalte einzurichten. Spezielle Schritte und Anleitungen zur Erstellung von Distributionen, die zusammen mit diesen Services verwendet werden können, finden Sie unter Stellen Sie das Video bereit, indem Sie AWS Elemental MediaStore als Ursprung und Servieren Sie Live-Videos, formatiert mit AWS Elemental MediaPackage.

Wowza und Unified Streaming bieten auch Tools, mit denen Sie Videos streamen können. CloudFront Weitere Informationen zur Verwendung von Wowza mit CloudFront findest du unter Bring deine Wowza Streaming Engine-Lizenz zum CloudFront Live-HTTP-Streaming auf der Wowza Dokumentationswebsite. Informationen zur Verwendung von Unified Streaming mit CloudFront für VOD-Streaming findest du CloudFront auf der Unified Streaming-Dokumentationswebsite.

Stellen Sie Video auf Abruf bereit mit CloudFront

Verwenden Sie die folgenden Dienste, um Video-on-Demand (VOD) -Streaming mit CloudFront bereitzustellen:

- Amazon S3, um den Inhalt in seinem Originalformat zu speichern und das transkodierte Video zu speichern,
- Ein Encoder (z. B. AWS Elemental MediaConvert) zum Transcodieren des Videos in Streaming-Formate.
- CloudFront um das transkodierte Video den Zuschauern zur Verfügung zu stellen. Informationen zu Microsoft Smooth Streaming finden Sie unter Konfigurieren von Video-on-Demand für Microsoft Smooth Streaming.

Um eine VOD-Lösung zu erstellen mit CloudFront

1. Laden Sie Ihre Inhalte in einen Amazon-S3-Bucket hoch. Weitere Informationen zum Arbeiten mit Amazon S3 finden Sie im Benutzerhandbuch zu Amazon Simple Storage Service.
2. Transkodieren Sie Ihre Inhalte mithilfe eines MediaConvert Jobs. Über den Auftrag wird Ihr Video in die Formate konvertiert, die von den Abspielgeräten, die Ihre Viewer verwenden, benötigt werden. Sie können den Auftrag auch dazu verwenden, um Komponenten (Assets) zu erstellen, die hinsichtlich der Auflösung und Bitrate variieren. Diese Ressourcen werden für das Streaming mit adaptiver Bitrate (ABR) verwendet, bei dem die Anzeigequalität an die verfügbare Bandbreite

des Betrachters angepasst wird. MediaConvert speichert das transkodierte Video in einem S3-Bucket.

3. Stellen Sie Ihre konvertierten Inhalte mithilfe einer CloudFront Distribution bereit. Viewer können die Inhalte jederzeit auf jedem Gerät ansehen.

Konfigurieren von Video-on-Demand für Microsoft Smooth Streaming

Sie haben die folgenden Optionen, CloudFront um Video-on-Demand-Inhalte (VOD) zu verteilen, die Sie in das Microsoft Smooth Streaming-Format transkodiert haben:

- Geben Sie einen Webserver, auf dem Microsoft IIS ausgeführt wird und der Smooth Streaming unterstützt, als Ursprung für Ihre Distribution an.
- Aktivieren Sie Smooth Streaming im Cache-Verhalten einer CloudFront Distribution. Da Sie in einer Distribution mehrere Cache-Verhaltensweisen verwenden können, können Sie eine Distribution sowohl für Smooth Streaming-Mediendateien als auch für andere Inhalte verwenden.

> **Important**
>
> Wenn Sie einen Webserver, auf dem Microsoft IIS ausgeführt wird, als Ihren Ursprung angeben, aktivieren Sie Smooth Streaming nicht im Cacheverhalten Ihrer CloudFront Distribution. CloudFront kann keinen Microsoft IIS-Server als Ursprung verwenden, wenn Sie Smooth Streaming als Cacheverhalten aktivieren.

Wenn Sie Smooth Streaming in einem Cache-Verhalten aktivieren (d. h. Sie verfügen nicht über einen Server mit Microsoft IIS), beachten Sie Folgendes:

- Sie können weiterhin andere Inhalte mit demselben Cache-Verhalten verteilen, wenn die Inhalte mit dem Wert von Path Pattern für dieses Cache-Verhalten übereinstimmen.
- CloudFront kann entweder einen Amazon S3-Bucket oder einen benutzerdefinierten Ursprung für Smooth Streaming-Mediendateien verwenden. CloudFront kann keinen Microsoft IIS-Server als Ursprung verwenden, wenn Sie Smooth Streaming für das Cache-Verhalten aktivieren.
- Sie können die Gültigkeit von Mediendateien im Smooth Streaming-Format nicht aufheben. Wenn Sie Dateien aktualisieren möchten, bevor sie ablaufen, müssen Sie diese umbenennen. Weitere Informationen finden Sie unter Inhalte hinzufügen, entfernen oder ersetzen, die CloudFront verbreiten.

Informationen zu Smooth-Streaming-Clients finden Sie unter Smooth Streaming auf der Microsoft-Dokumentations-Website.

Zur Verwendung CloudFront zur Verteilung von Smooth Streaming-Dateien, wenn ein Microsoft IIS-Webserver nicht der Ursprung ist

1. Wandeln Sie Ihre Mediendateien in das Smooth Streaming-Fragmented-MP4-Format um.
2. Führen Sie eine der folgenden Aktionen aus:
   - Wenn Sie die CloudFront Konsole verwenden: Wenn Sie eine Distribution erstellen oder aktualisieren, aktivieren Sie Smooth Streaming in einem oder mehreren Cacheverhalten der Distribution.
   - Wenn du die CloudFront API verwendest: Füge das `SmoothStreaming` Element dem `DistributionConfig` komplexen Typ für eines oder mehrere der Cache-Verhaltensweisen der Distribution hinzu.
3. Laden Sie die Smooth Streaming-Dateien auf den Ursprung hoch.
4. Erstellen Sie entweder eine `clientaccesspolicy.xml`- oder eine `crossdomainpolicy.xml`-Datei und fügen Sie diese zu einem Speicherort hinzu, der am Stamm Ihrer Distribution zugänglich ist, z. B. `https://d111111abcdef8.cloudfront.net/clientaccesspolicy.xml`. Folgendes ist eine Beispielrichtlinie:

```
<?xml version="1.0" encoding="utf-8"?>
<access-policy>
<cross-domain-access>
<policy>
<allow-from http-request-headers="*">
<domain uri="*"/>
</allow-from>
<grant-to>
<resource path="/" include-subpaths="true"/>
</grant-to>
</policy>
</cross-domain-access>
</access-policy>
```

   Weitere Informationen finden Sie unter Making a Service Available Across Domain Boundaries auf der Microsoft Developer Network-Website.

5. Geben Sie in Ihrer Anwendung (z.B. einem Media-Player) für Links die URL für die Mediendatei im folgenden Format an:

   ```
   https://d111111abcdef8.cloudfront.net/video/presentation.ism/Manifest
   ```

Stellen Sie Videostreaming mit CloudFront und bereit AWS Mediendienste

Informationen zur Verwendung von AWS Media Services CloudFront zur Bereitstellung von Live-Inhalten für ein globales Publikum finden Sie in den folgenden Anleitungen.

Verwenden Sie AWS Elemental MediaLive, um Live-Video-Streams in Echtzeit zu codieren. Um einen großen Videostream zu kodieren, MediaLive komprimieren Sie ihn in kleinere Versionen (Codierungen), die an Ihre Zuschauer verteilt werden können.

Nachdem Sie einen Live-Videostream komprimiert haben, können Sie eine der folgenden beiden Hauptoptionen verwenden, um den Inhalt vorzubereiten und bereitzustellen:

- Ihre Inhalte in die erforderlichen Formate konvertieren und anschließend bereitstellen – Wenn Sie die Inhalte in mehreren Formaten benötigen, verwenden Sie AWS Elemental MediaPackage, um die Inhalte für verschiedene Gerätetypen zu verpacken. Wenn Sie die Inhalte verpacken, können Sie auch zusätzliche Funktionen implementieren und das Digital Rights Management (digitale Rechteverwaltung, DRM) hinzufügen, um eine unbefugte Verwendung Ihrer Inhalte zu verhindern. Eine schrittweise Anleitung zur Verwendung MediaPackage formatierter Inhalte CloudFront zum Bereitstellen von Inhalten finden Sie unter. Servieren Sie Live-Videos, formatiert mit AWS Elemental MediaPackage
- Speichern und Bereitstellen Ihrer Inhalte mit skalierbarem Ursprung — Wenn Inhalte in den Formaten MediaLive codiert sind, die von allen Geräten, die Ihre Zuschauer verwenden, erforderlich sind, verwenden Sie einen hochgradig skalierbaren Ursprung, z. B. für die AWS Elemental MediaStore Bereitstellung der Inhalte. Eine schrittweise Anleitung zur Verwendung CloudFront zum Bereitstellen von Inhalten, die in einem MediaStore Container gespeichert sind, finden Sie unterStellen Sie das Video bereit, indem Sie AWS Elemental MediaStore als Ursprung.

Nachdem Sie Ihren Ursprung eingerichtet haben, indem Sie eine dieser Optionen auswählen, können Sie Live-Streaming-Videos mithilfe von CloudFront an Betrachter verteilen.

> **Tip**
>
> Sie können sich über eine AWS Lösung informieren, die automatisch Dienste bereitstellt, um ein hochverfügbares Seherlebnis in Echtzeit zu bieten. Die Schritte zur automatischen Bereitstellung dieser Lösung können Sie unter Live-Streaming – automatische Bereitstellung einsehen.

Themen

- Stellen Sie das Video bereit, indem Sie AWS Elemental MediaStore als Ursprung
- Servieren Sie Live-Videos, formatiert mit AWS Elemental MediaPackage
- Bieten Sie Video-on-Demand-Inhalte mit AWS Elemental MediaPackage

Stellen Sie das Video bereit, indem Sie AWS Elemental MediaStore als Ursprung

Wenn Sie ein Video in einem AWS Elemental MediaStore Container gespeichert haben, können Sie eine CloudFront Distribution erstellen, um den Inhalt bereitzustellen.

Zunächst gewähren Sie CloudFront Zugriff auf Ihren MediaStore Container. Dann erstellen Sie eine CloudFront Distribution und konfigurieren sie so, dass sie damit funktioniert MediaStore.

Um Inhalte von einem bereitzustellen AWS Elemental MediaStore Container

1. Folgen Sie den Anweisungen unter Erlauben Sie Amazon den CloudFront Zugriff auf Ihren AWS Elemental MediaStore Container, und kehren Sie dann zu diesen Schritten zurück, um Ihre Distribution zu erstellen.
2. Verwenden Sie die folgenden Einstellungen, um eine Distribution zu erstellen:

    a. Ursprungsdomäne — Der Datenendpunkt, der Ihrem MediaStore Container zugewiesen ist. Wählen Sie aus der Dropdownliste den MediaStore Container für Ihr Live-Video aus.

    b. Quellpfad — Die Ordnerstruktur im MediaStore Container, in dem Ihre Objekte gespeichert sind. Weitere Informationen finden Sie unter the section called “Ursprungspfad”.

    c. Benutzerdefinierten Header hinzufügen — Fügen Sie Header-Namen und -Werte hinzu CloudFront , wenn Sie benutzerdefinierte Header hinzufügen möchten, wenn Anfragen an Ihren Ursprung weitergeleitet werden.

d. Viewer-Protokollrichtlinie – Wählen Sie HTTP zu HTTPS umleiten aus. Weitere Informationen finden Sie unter the section called “Viewer-Protokollrichtlinien”.

e. Cache-Richtlinie und Ursprungsanforderungsrichtlinie

- Wählen Sie für Cache policy (Cache-Richtlinie) die Option Create policy (Richtlinie erstellen) aus und erstellen Sie dann eine Cache-Richtlinie, die Ihren Caching-Anforderungen und der Segmentdauer entspricht. Aktualisieren Sie nach dem Erstellen der Richtlinie die Liste der Cache-Richtlinien und wählen Sie die Richtlinie aus, die Sie gerade erstellt haben.
- Wählen Sie für die Origin-Anforderungsrichtlinie CORS-CustomOrigin aus der Dropdownliste aus.

Für die anderen Einstellungen können Sie bestimmte Werte basierend auf anderen technischen Anforderungen oder den Anforderungen Ihres Unternehmens festlegen. Eine Liste aller Optionen für Verteilungen und Informationen über ihre Einstellungen finden Sie unter the section called “Alle Distributionseinstellungen”.

3. Geben Sie für Links in Ihrer Anwendung (z. B. in einem Media Player) den Namen der Mediendatei in demselben Format an, das Sie für andere Objekte verwenden, die Sie verwenden CloudFront.

Servieren Sie Live-Videos, formatiert mit AWS Elemental MediaPackage

Wenn Sie einen Livestream mithilfe von formatiert haben AWS Elemental MediaPackage, können Sie eine CloudFront Verteilung erstellen und das Cache-Verhalten so konfigurieren, dass der Livestream bereitgestellt wird. Im folgenden Prozess wird davon ausgegangen, dass Sie bereits einen Kanal erstellt und Endpunkte für Ihr Live-Video hinzugefügt haben. MediaPackage

Gehen Sie wie folgt vor, um eine CloudFront Verteilung für MediaPackage manuell zu erstellen:

Schritt 1: Erstellen und konfigurieren Sie eine CloudFront Distribution

Gehen Sie wie folgt vor, um eine CloudFront Verteilung für den Live-Videokanal einzurichten, mit dem Sie erstellt haben MediaPackage.

So erstellen Sie eine Distribution für Ihren Live-Video-Kanal

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home.

2. Wählen Sie Create distribution (Distribution erstellen) aus.
3. Wählen Sie die Einstellungen für die Distribution aus, einschließlich der folgenden:

   Ursprungsdomäne

   Der Ursprung, an dem sich Ihr MediaPackage Live-Videokanal und Ihre Endpunkte befinden. Wählen Sie das Textfeld aus und wählen Sie dann aus der Dropdownliste die MediaPackage Ursprungsdomain für Ihr Live-Video aus. Sie können eine Domäne zu mehreren Ursprungsendpunkten zuordnen.

   Wenn Sie Ihre Ursprungsdomäne mit einem anderen AWS -Konto erstellt haben, geben Sie den Ursprung-URL-Wert in das Feld ein. Der Ursprung muss eine HTTPS-URL sein.

   Beispiel: bei einem HLS-Endpunkt wie `https://3ae97e9482b0d011.mediapackage.us-west-2.amazonaws.com/out/v1/abc123/index.m3u8` ist die Ursprungsdomäne `3ae97e9482b0d011.mediapackage.us-west-2.amazonaws.com`.

   Weitere Informationen finden Sie unter the section called “Ursprungsdomäne”.

   Ursprungspfad

   Der Pfad zum MediaPackage Endpunkt, von dem aus der Inhalt bereitgestellt wird.

   Weitere Informationen über die Funktionsweise eines Ursprungspfads finden Sie unter the section called “Ursprungspfad”.

   > **Important**
   >
   > Der Platzhalterpfad `*` ist für die Weiterleitung an einer beliebigen Stelle in der CloudFront Distribution erforderlich. Um zu verhindern, dass Anfragen, die keinem expliziten Pfad entsprechen, an den echten Ursprung weitergeleitet werden, erstellen Sie einen „Dummy“ -Ursprung für diesen Platzhalterpfad.

Example: Erstellen eines „Dummy“-Ursprungs

Im folgenden Beispiel werden die Endpunkte `abc123` und `def456` zum „echten“ Ursprung weitergeleitet, aber Anfragen nach Videoinhalten eines anderen Endpunkts werden ohne die

richtige Subdomäne an `mediapackage.us-west-2.amazonaws.com` weitergeleitet. Das führt zu einem 404-HTTP-Fehler.

MediaPackage Endpunkte:

```
https://3ae97e9482b0d011.mediapackage.us-west-2.amazonaws.com/out/v1/abc123/
index.m3u8
https://3ae97e9482b0d011.mediapackage.us-west-2.amazonaws.com/out/v1/def456/
index.m3u8
```

CloudFront Ursprung A:

```
Domain: 3ae97e9482b0d011.mediapackage.us-west-2.amazonaws.com
Path: None
```

CloudFront Herkunft B:

```
Domain: mediapackage.us-west-2.amazonaws.com
Path: None
```

CloudFront Verhalten des Caches:

```
1. Path: /out/v1/abc123/* forward to Origin A
2. Path: /out/v1/def456/* forward to Origin A
3. Path: * forward to Origin B
```

Für die anderen Distributionseinstellungen können Sie bestimmte Werte basierend auf anderen technischen Anforderungen oder den Anforderungen Ihres Unternehmens festlegen. Eine Liste aller Optionen für Distributionen und Informationen über ihre Einstellungen finden Sie unter the section called “Alle Distributionseinstellungen”.

Wenn Sie die Auswahl der anderen Distributionseinstellungen abgeschlossen haben, wählen Sie Create Distribution (Distribution erstellen) aus.

4. Wählen Sie die gerade erstellte Distribution aus und klicken Sie dann auf Behaviors (Verhaltensweisen).
5. Wählen Sie das Cache-Standardverhalten und anschließend Edit (Bearbeiten) aus. Geben Sie die korrekten Einstellungen für das Cache-Verhalten für den Kanal an, den Sie für den Ursprung

auswählen. Später fügen Sie einen oder mehrere zusätzliche Ursprünge hinzu und bearbeiten deren Einstellungen für das Cache-Verhalten.

6. Gehen Sie zur Seite mit den CloudFront Distributionen.
7. Warten Sie, bis sich der Wert in der Spalte Letzte Änderung für Ihre Distribution von Bereitgestellt auf ein Datum und eine Uhrzeit geändert CloudFront hat, was darauf hindeutet, dass Ihre Distribution erstellt wurde.

Schritt 2: Fügen Sie Origins für die Domänen Ihrer MediaPackage Endpunkte hinzu

Wiederhole die Schritte hier, um jeden deiner MediaPackage Kanal-Endpunkte zu deiner Distribution hinzuzufügen. Denke dabei daran, dass du einen „Dummy“ -Ursprung erstellen musst.

So fügen Sie andere Endpunkte als Ursprünge hinzu

1. Wähle auf der CloudFront Konsole die Distribution aus, die du für deinen Kanal erstellt hast.
2. Klicken Sie auf Origins (Ursprünge) und wählen Sie Create origin (Ursprung erstellen) aus.
3. Wähle in der Dropdownliste für die Ursprungsdomain einen MediaPackage Endpunkt für deinen Kanal aus.
4. Für die anderen Einstellungen legen Sie die Werte basierend auf anderen technischen Anforderungen oder den Anforderungen Ihres Unternehmens fest. Weitere Informationen finden Sie unter the section called “Ursprungseinstellungen”.
5. Wählen Sie Create Origin (Ursprung erstellen) aus.

Schritt 3: Konfigurieren der Cache-Verhaltensweisen für alle Endpunkte

Für jeden Endpunkt müssen Sie Cache-Verhaltensweisen konfigurieren, um Pfadmuster hinzuzufügen, die Anfragen korrekt weiterleiten. Die Pfadmuster, die Sie angeben, hängen vom bereitgestellten Videoformat ab. Das folgende Verfahren umfasst die Pfadmuster-Informationen, die für Apple HLS-, CMAF-, DASH- und Microsoft Smooth Streaming-Formate zu verwenden sind.

Sie richten in der Regel zwei Cache-Verhaltensweisen für jeden Endpunkt ein:

- Das übergeordnete Manifest, bei dem es sich um den Index für Ihre Dateien handelt.
- Die Segmente, die Dateien der Videoinhalte darstellen.

So erstellen Sie ein Cache-Verhalten für einen Endpunkt

1. Wähle auf der CloudFront Konsole die Distribution aus, die du für deinen Kanal erstellt hast.
2. Wählen Sie Behaviors (Verhaltensweisen) und anschließend die Option Create behavior (Verhalten erstellen) aus.
3. Verwenden Sie für das Pfadmuster eine bestimmte MediaPackage `OriginEndpoint` GUID als Pfadpräfix.

   Pfadmuster

   Erstellen Sie für einen HLS-Endpunkt wie `https://3ae97e9482b0d011.mediapackage.us-west-2.amazonaws.com/out/v1/abc123/index.m3u8` die folgenden zwei Cache-Verhaltensweisen:

   - Verwenden Sie für übergeordnete und untergeordnete Manifeste `/out/v1/abc123/*.m3u8`.
   - Für die Inhaltssegmente verwenden Sie `/out/v1/abc123/*.ts`.

   Erstellen Sie für einen CMAF-Endpunkt wie `https://3ae97e9482b0d011.mediapackage.us-west-2.amazonaws.com/out/v1/abc123/index.m3u8` die folgenden zwei Cache-Verhaltensweisen:

   - Verwenden Sie für übergeordnete und untergeordnete Manifeste `/out/v1/abc123/*.m3u8`.
   - Für die Inhaltssegmente verwenden Sie `/out/v1/abc123/*.mp4`.

   Erstellen Sie für einen DASH-Endpunkt wie `https://3ae97e9482b0d011.mediapackage.us-west-2.amazonaws.com/out/v1/abc123/index.mpd` die folgenden zwei Cache-Verhaltensweisen:

   - Verwenden Sie für das übergeordnete Manifest `/out/v1/abc123/*.mpd`.
   - Für die Inhaltssegmente verwenden Sie `/out/v1/abc123/*.mp4`.

   Für einen Microsoft-Smooth-Streaming-Endpunkt wie `https://3ae97e9482b0d011.mediapackage.us-west-2.amazonaws.com/out/v1/abc123/index.ism` wird nur ein Manifest bereitgestellt, sodass Sie nur ein Cache-Verhalten erstellen: `out/v1/abc123/index.ism/*`.
4. Geben Sie für jedes Cache-Verhalten Werte für die folgenden Einstellungen an:

Viewer-Protokollrichtlinien

Wählen Sie Redirect HTTP to HTTPS (HTTP an HTTPS umleiten) aus.

Cache-Richtlinien und Ursprungsanforderungsrichtlinie

Wählen Sie für Cache policy (Cache-Richtlinie) die Option Create policy (Richtlinie erstellen) aus. Geben Sie für Ihre neue Cache-Richtlinie die folgenden Einstellungen an:

Mindest-TTL

Legen Sie diese Einstellung auf 5 Sekunden oder weniger fest, um zu verhindern, dass veralteter Inhalt bereitgestellt wird.

Abfragezeichenfolgen

Wählen Sie für Query strings (Abfragezeichenfolgen) (in Cache key settings (Cache-Schlüssel-Einstellungen)) die Option Include specified query strings (Angegebene Abfragezeichenfolgen einschließen) aus. Fügen Sie unter Allow (Erlauben) die folgenden Werte hinzu, indem Sie sie eingeben und anschließend Add item (Element hinzufügen) auswählen:

- Fügen Sie `m` als Abfragezeichenfolge einen Parameter hinzu, CloudFront den Sie als Grundlage für das Caching verwenden möchten. Die MediaPackage Antwort enthält immer das Tag`?m=###`, mit dem die geänderte Uhrzeit des Endpunkts erfasst wird. Wenn Inhalt bereits mit einem anderen Wert für dieses Tag zwischengespeichert ist, CloudFront wird ein neues Manifest angefordert, anstatt die zwischengespeicherte Version bereitzustellen.
- Wenn Sie die Funktion zur zeitversetzten Anzeige in verwenden MediaPackage, geben Sie `start` und `end` als zusätzliche Abfragezeichenfolgenparameter für das Cacheverhalten für Manifestanforderungen an (`*.m3u8*.mpd`, und). `index.ism/*` Auf diese Weise werden Inhalte bereitgestellt, die spezifisch für den angeforderten Zeitraum in der Manifest-Anfrage sind. Weitere Informationen zum zeitversetzten Anzeigen und Formatieren von Start- und Endanforderungsparametern für Inhalte finden Sie im AWS Elemental MediaPackage Benutzerhandbuch unter Time-shifted Anzeigen.
- Wenn Sie die Funktion zum Filtern von Manifesten in verwenden MediaPackage, geben Sie `aws.manifestfilter` als zusätzlichen Abfragezeichenfolge-Parameter die Cache-Richtlinie an, die Sie zusammen mit dem Cacheverhalten für Manifestanforderungen verwenden (`*.m3u8*.mpd`, und`index.ism/*`). Dadurch wird

Ihre Distribution so konfiguriert, dass die `aws.manifestfilter` Abfragezeichenfolge an Ihren MediaPackage Ursprung weitergeleitet wird. Dies ist erforderlich, damit die Manifestfilterfunktion funktioniert. Weitere Informationen finden Sie unter Manifestfilterung im AWS Elemental MediaPackage Benutzerhandbuch.

- Wenn Sie HLS (LL-HLS) mit niedriger Latenz verwenden, geben Sie `_HLS_msn` und `_HLS_part` als zusätzliche Abfragezeichenfolgenparameter für die Cache-Richtlinie an, die Sie zusammen mit dem Cache-Verhalten für Manifestanforderungen () verwenden. `*.m3u8` Dadurch wird Ihre Distribution so konfiguriert, dass sie die Zeichenketten `_HLS_msn` und `_HLS_part` Abfragen an Ihren MediaPackage Ursprung weiterleitet. Dies ist erforderlich, damit die Funktion zum LL-HLS Blockieren von Playlist-Anfragen funktioniert.

5. Wählen Sie Erstellen aus.
6. Nachdem Sie die Cache-Richtlinie erstellt haben, kehren Sie zum Workflow zur Erstellung des Cacheverhaltens zurück. Aktualisieren Sie die Liste der Cache-Richtlinien und wählen Sie die Richtlinie aus, die Sie gerade erstellt haben.
7. Wählen Sie Create behavior (Verhalten erstellen) aus.
8. Wenn es sich bei Ihrem Endpunkt nicht um einen Microsoft-Smooth-Streaming–Endpunkt handelt, wiederholen Sie diese Schritte, um ein zweites Cache-Verhalten zu erstellen.

Schritt 4: Aktiviere die MediaPackage Header-basierte CDN-Autorisierung

Wir empfehlen, die headerbasierte MediaPackage CDN-Autorisierung zwischen Endpunkten und der Distribution zu aktivieren. MediaPackage CloudFront Weitere Informationen finden Sie im MediaPackage Benutzerhandbuch unter CDN-Autorisierung aktivieren. AWS Elemental MediaPackage

Schritt 5: CloudFront Zur Bereitstellung des Livestream-Kanals verwenden

Nachdem Sie die Distribution erstellt, die Ursprünge hinzugefügt, das Cache-Verhalten erstellt und die Header-basierte CDN-Autorisierung aktiviert haben, können Sie den Livestream-Kanal mit bereitstellen. CloudFront CloudFront leitet Anfragen von Zuschauern auf der Grundlage der Einstellungen, die Sie für das Cache-Verhalten konfiguriert haben, an die richtigen MediaPackage Endpunkte weiter.

Geben Sie für Links in Ihrer Anwendung (z. B. einem Media Player) die URL für die Mediendatei im Standardformat für CloudFront URLs an. Weitere Informationen finden Sie unter the section called “Anpassen von Datei-URLs”.

Bieten Sie Video-on-Demand-Inhalte mit AWS Elemental MediaPackage

Wenn Sie Ihre Video-on-Demand-Inhalte (VOD) von einer Quelle beziehen, AWS Elemental MediaPackage können Sie eine CloudFront Verteilung erstellen und optimiertes Cacheverhalten konfigurieren, um die VOD-Inhalte den Zuschauern bereitzustellen. Im folgenden Prozess wird davon ausgegangen, dass Sie bereits eine Verpackungsgruppe mit einer Verpackungskonfiguration erstellt und ein Asset mit aufgenommen haben. MediaPackage

Gehen Sie folgendermaßen vor, um eine CloudFront Verteilung für MediaPackage manuell zu erstellen:

Schritt 1: Erstellen und konfigurieren Sie eine CloudFront Distribution

Gehen Sie wie folgt vor, um eine CloudFront Verteilung für die Verpackungsgruppe einzurichten, mit der Sie erstellt haben MediaPackage.

So erstellen Sie eine Distribution für Ihre VOD-Inhalte

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home.
2. Wählen Sie Create distribution (Distribution erstellen) aus.
3. Wählen Sie die Einstellungen für die Distribution aus, einschließlich der folgenden:

   Ursprungsdomäne

   Der Ursprung Ihrer MediaPackage Verpackungsgruppe. Geben Sie den Wert der Ursprungs-URL in das Textfeld ein. Der Ursprung muss eine HTTPS-URL sein.

   Beispiel: bei einem HLS-Endpunkt wie `https://3ae97e9482b0d011.egress.mediapackage-vod.us-west-2.amazonaws.com/out/v1/abc123/def456/ghi789/index.m3u8` ist die Ursprungsdomäne `3ae97e9482b0d011.egress.mediapackage-vod.us-west-2.amazonaws.com`.

   Weitere Informationen finden Sie unter the section called “Ursprungsdomäne”.

   Ursprungspfad

   Der Pfad, von dem aus der Inhalt bereitgestellt wird.

Weitere Informationen über die Funktionsweise eines Ursprungspfads finden Sie unter the section called “Ursprungspfad”.

> **Important**
>
> Der Platzhalterpfad * ist erforderlich, um irgendwo in der CloudFront Distribution weiterzuleiten. Um zu verhindern, dass Anfragen, die keinem expliziten Pfad entsprechen, an den echten Ursprung weitergeleitet werden, erstellen Sie einen „Dummy“ -Ursprung für diesen Platzhalterpfad.

Example: Erstellen eines „Dummy“-Ursprungs

Im folgenden Beispiel werden die Verpackungskonfigurationen `def456` und `321xyz` zum „echten“ Ursprung weitergeleitet, aber Anforderungen nach anderen Videoinhalten werden ohne die richtige Subdomain an `mediapackage-vod.us-west-2.amazonaws.com` weitergeleitet. Das führt zu einem HTTP-`404`-Fehler.

MediaPackage Inhalts-URLs für ein einzelnes Asset für eine Verpackungsgruppe mit zwei Verpackungskonfigurationen:

```
https://3ae97e9482b0d011.egress.mediapackage-vod.us-west-2.amazonaws.com/out/v1/
abc123/def456/ghi789/index.m3u8
https://3ae97e9482b0d011.egress.mediapackage-vod.us-west-2.amazonaws.com/out/v1/
abc123/321xyz/654uvw/index.m3u8
```

CloudFront Herkunft A:

```
Domain: 3ae97e9482b0d011.egress.mediapackage-vod.us-west-2.amazonaws.com
Path: None
```

CloudFront Herkunft B:

```
Domain: mediapackage-vod.us-west-2.amazonaws.com
Path: None
```

CloudFront Cache-Verhalten:

```
1. Path: /out/v1/*/def456/* forward to Origin A
2. Path: /out/v1/*/321xyz/* forward to Origin A
3. Path: * forward to Origin B
```

Für die anderen Distributionseinstellungen können Sie bestimmte Werte basierend auf anderen technischen Anforderungen oder den Anforderungen Ihres Unternehmens festlegen. Eine Liste aller Optionen für Distributionen und Informationen über ihre Einstellungen finden Sie unter the section called “Alle Distributionseinstellungen”.

Wenn Sie die Auswahl der anderen Distributionseinstellungen abgeschlossen haben, wählen Sie Create Distribution (Distribution erstellen) aus.

4. Wählen Sie die gerade erstellte Distribution aus und klicken Sie dann auf Behaviors (Verhaltensweisen).
5. Wählen Sie das Cache-Standardverhalten und anschließend Edit (Bearbeiten) aus. Geben Sie die korrekten Einstellungen für das Cache-Verhalten für die Verpackungskonfiguration an, die Sie für den Ursprung auswählen. Später fügen Sie einen oder mehrere zusätzliche Ursprünge hinzu und bearbeiten deren Einstellungen für das Cache-Verhalten.
6. Gehen Sie zur Seite mit den CloudFront Distributionen.
7. Warten Sie, bis sich der Wert in der Spalte Letzte Änderung für Ihre Distribution von Bereitgestellt auf ein Datum und eine Uhrzeit geändert CloudFront hat, was darauf hindeutet, dass Ihre Distribution erstellt wurde.

Schritt 2: Fügen Sie Origins für die Domains Ihrer MediaPackage Paketgruppen hinzu

Wiederholen Sie die hier aufgeführten Schritte, um jede Ihrer MediaPackage Verpackungsgruppen zu Ihrer Distribution hinzuzufügen. Beachten Sie dabei, dass Sie einen „Dummy“ -Ursprung erstellen müssen.

So fügen Sie andere Verpackungsgruppen als Ursprünge hinzu

1. Wähle auf der CloudFront Konsole die Distribution aus, die du für deinen Kanal erstellt hast.
2. Klicken Sie auf Origins (Ursprünge) und wählen Sie Create origin (Ursprung erstellen) aus.
3. Geben Sie für die Origin-Domain die URL für die MediaPackage Paketgruppe ein.
4. Für die anderen Einstellungen legen Sie die Werte basierend auf anderen technischen Anforderungen oder den Anforderungen Ihres Unternehmens fest. Weitere Informationen finden Sie unter the section called “Ursprungseinstellungen”.

5. Wählen Sie Create Origin (Ursprung erstellen) aus.

Schritt 3: Konfigurieren des Cache-Verhaltens für alle Verpackungskonfigurationen

Für jede Verpackungskonfiguration müssen Sie Cache-Verhaltensweisen konfigurieren, um Pfadmuster hinzuzufügen, die Anforderungen korrekt weiterleiten. Die Pfadmuster, die Sie angeben, hängen vom bereitgestellten Videoformat ab. Das folgende Verfahren umfasst die Pfadmuster-Informationen, die für Apple HLS-, CMAF-, DASH- und Microsoft Smooth Streaming-Formate zu verwenden sind.

Sie richten in der Regel mehrere Cache-Verhaltensweisen für jede Verpackungskonfiguration ein:

- Das übergeordnete Manifest, bei dem es sich um den Index für Ihre Dateien handelt.
- Die Segmente, die Dateien der Videoinhalte darstellen. Ein Format kann je nach Konfiguration mehr als eine Erweiterung für Inhalte verwenden. Für jede Erweiterung ist ein Cache-Verhalten erforderlich.

So erstellen Sie ein Cache-Verhalten für eine Verpackungskonfiguration

1. Wähle auf der CloudFront Konsole die Distribution aus, die du für deinen Kanal erstellt hast.
2. Wählen Sie Behaviors (Verhaltensweisen) und anschließend die Option Create behavior (Verhalten erstellen) aus.
3. Verwenden Sie für das Pfadmuster eine bestimmte MediaPackage VOD-Paketkonfigurations-GUID als Pfadpräfix. Dies ist die zweite GUID in einem MediaPackage VOD-Pfad.

   Pfadmuster

   Erstellen Sie für einen HLS-Inhalt wie `https://3ae97e9482b0d011.egress.mediapackage-vod.us-west-2.amazonaws.com/out/v1/abc123/def456/ghi789/index.m3u8` die folgenden zwei Cache-Verhaltensweisen:

   - Verwenden Sie für übergeordnete und untergeordnete Manifeste `/out/v1/*/def456/*.m3u8`.
   - Für die Inhaltssegmente verwenden Sie `/out/v1/*/def456/*.ts` und wiederholen den Vorgang für alle benötigten Segmenterweiterungen.

Erstellen Sie für einen CMAF-Inhalt wie `https://3ae97e9482b0d011.egress.mediapackage-vod.us-west-2.amazonaws.com/out/v1/abc123/def456/ghi789/index.m3u8` die folgenden zwei Cache-Verhaltensweisen:

- Verwenden Sie für übergeordnete und untergeordnete Manifeste `/out/v1/*/def456/*.m3u8`.
- Für die Inhaltssegmente verwenden Sie `/out/v1/*/def456/*.mp4` und wiederholen den Vorgang für alle benötigten Segmenterweiterungen.

Erstellen Sie für einen DASH-Inhalt wie `https://3ae97e9482b0d011.egress.mediapackage-vod.us-west-2.amazonaws.com/out/v1/abc123/def456/ghi789/index.mpd` die folgenden zwei Cache-Verhaltensweisen:

- Verwenden Sie für das übergeordnete Manifest `/out/v1/*/def456/*.mpd`.
- Für die Inhaltssegmente verwenden Sie `/out/v1/*/def456/*.mp4`.

Für einen Microsoft-Smooth-Streaming-Endpunkt wie `https://3ae97e9482b0d011.egress.mediapackage-vod.us-west-2.amazonaws.com/out/v1/abc123/def456/ghi789/index.ism/Manifest` wird nur ein Manifest bereitgestellt, sodass Sie nur ein Cache-Verhalten erstellen: `out/v1/*/def456/*/index.ism/*`.

4. Geben Sie für jedes Cache-Verhalten Werte für die folgenden Einstellungen an:

   Viewer-Protokollrichtlinien

   Wählen Sie Redirect HTTP to HTTPS (HTTP an HTTPS umleiten) aus.

   Cache-Richtlinien und Ursprungsanforderungsrichtlinie

   Wählen Sie für Cache policy (Cache-Richtlinie) die Option Create policy (Richtlinie erstellen) aus. Geben Sie für Ihre neue Cache-Richtlinie die folgenden Einstellungen an:

   Mindest-TTL

   Legen Sie diese Einstellung auf 5 Sekunden oder weniger fest, um zu verhindern, dass veralteter Inhalt bereitgestellt wird.

Abfragezeichenfolgen

Wählen Sie für Query strings (Abfragezeichenfolgen) (in Cache key settings (Cache-Schlüssel-Einstellungen)) die Option Include specified query strings (Angegebene Abfragezeichenfolgen einschließen) aus. Fügen Sie unter Allow (Erlauben) die folgenden Werte hinzu, indem Sie sie eingeben und anschließend Add item (Element hinzufügen) auswählen:

- Wenn Sie die Funktion zur Manifestfilterung in verwenden MediaPackage, geben Sie `aws.manifestfilter` als zusätzlichen Abfragezeichenfolgenparameter die Cache-Richtlinie an, die Sie zusammen mit dem Cacheverhalten für Manifestanforderungen verwenden (`*.m3u8*.mpd`, und`index.ism/*`). Dadurch wird Ihre Distribution so konfiguriert, dass die `aws.manifestfilter` Abfragezeichenfolge an Ihren MediaPackage Ursprung weitergeleitet wird. Dies ist erforderlich, damit die Manifestfilterfunktion funktioniert. Weitere Informationen finden Sie unter Manifestfilterung im AWS Elemental MediaPackage Benutzerhandbuch.

5. Wählen Sie Erstellen aus.
6. Nachdem Sie die Cache-Richtlinie erstellt haben, kehren Sie zum Workflow zur Erstellung des Cacheverhaltens zurück. Aktualisieren Sie die Liste der Cache-Richtlinien und wählen Sie die Richtlinie aus, die Sie gerade erstellt haben.
7. Wählen Sie Create behavior (Verhalten erstellen) aus.
8. Wenn es sich bei Ihrem Endpunkt nicht um einen Microsoft-Smooth-Streaming–Endpunkt handelt, wiederholen Sie diese Schritte, um ein zweites Cache-Verhalten zu erstellen.

Schritt 4: Aktivieren Sie die MediaPackage Header-basierte CDN-Autorisierung

Wir empfehlen, die headerbasierte MediaPackage CDN-Autorisierung zwischen MediaPackage VOD-Inhalten und der Distribution zu aktivieren. CloudFront Weitere Informationen finden Sie im MediaPackage Benutzerhandbuch unter CDN-Autorisierung aktivieren. AWS Elemental MediaPackage

Schritt 5: CloudFront Zur Bereitstellung des VOD-Inhalts verwenden

Nachdem Sie die Distribution erstellt, die Ursprünge hinzugefügt, das Cache-Verhalten erstellt und die Header-basierte CDN-Autorisierung aktiviert haben, können Sie den VOD-Inhalt mithilfe von bereitstellen. CloudFront CloudFront leitet Anfragen von Viewern auf Grundlage der Einstellungen, die Sie für das Cache-Verhalten konfiguriert haben, an den richtigen MediaPackage VOD-Inhalt weiter.

Geben Sie für Links in Ihrer Anwendung (z. B. einem Media Player) die URL für die Mediendatei im Standardformat für CloudFront URLs an. Weitere Informationen finden Sie unter the section called “Anpassen von Datei-URLs”.

Media Quality-Aware Resiliency

Media Quality-Aware Resiliency (MQAR) ist eine integrierte Funktion zwischen Amazon und Media Services. CloudFront AWS MQAR bietet eine automatisierte regionsübergreifende Ursprungsauswahl auf der Grundlage des Media Quality Confidence Score (MQCS). Der MQCS wird von AWS Elemental MediaLive auf der Grundlage von Parametern synthetisiert, die sich auf die Medienqualität auswirken, die von Viewern wahrgenommen wird. Sie können AWS Media Services so konfigurieren CloudFront , dass Ihr Live-Event-Streaming mit hoher Ausfallsicherheit bereitgestellt wird, indem Sie mehrere Optionen verwenden, die Sie in den Failover-Kriterien für die CloudFront Ursprungsgruppe angeben können.

Wenn Sie die MQAR-Funktion für Ihre Distribution aktivieren, autorisieren CloudFront Sie, automatisch die Quelle auszuwählen, von der angenommen wird, dass sie den höchsten Qualitätsfaktor aufweist.

Der Qualitätsfaktor steht für wahrgenommene Qualitätsprobleme beim Medienstreaming aufgrund Ihres Ursprungs, wie z. B. schwarze Frames, eingefrorene oder verworfene Frames oder wiederholte Frames. Wenn Ihre AWS Elemental MediaPackage v2-Quellen beispielsweise in zwei verschiedenen AWS-Regionen Quellen bereitgestellt werden und einer einen höheren Medienqualitätsfaktor als der andere meldet, CloudFront wird automatisch zu der Quelle gewechselt, die den höheren Wert meldet.

Gehen Sie wie folgt vor, CloudFront um dies zu erreichen:

1. CloudFront leitet eine `GET` Anfrage an den primären MediaPackage Ursprung weiter und initiiert gleichzeitig eine `HEAD` Anfrage an den sekundären MediaPackage Ursprung. CloudFront empfängt den Medienqualitätswert in den Antwort-Headern von jedem Ursprung.
2. Verfolgt als CloudFront Nächstes die Bewertung für jeden Ursprung und verwendet diese Informationen, um den Ursprung mit dem höheren Punktewert zu ermitteln, wenn eine neue Anfrage eingeht.

Die Bewertung der Medienqualität für Ihre Herkunft kann sich in Echtzeit ändern. CloudFront bestimmt dies anhand der MQCS-Änderungen und wechselt zwischen den Quellen, um sicherzustellen, dass die Zuschauer die Inhalte mit höherer Medienqualität sehen. Weitere

Informationen finden Sie im AWS Elemental MediaPackage V2-Benutzerhandbuch unter Nutzung der MediaPackage Medienqualitätsauswertungen mit.

MQAR hilft dabei, so früh wie möglich CloudFront festzustellen, ob ein Problem vorliegt, das sich potenziell auf Kunden auswirken könnte. Probleme mit der Netzwerkverbindung, Videoverarbeitung, Audioverlust oder -ausfälle sowie Probleme mit der Encodergeschwindigkeit können sich beispielsweise auf den Medienqualitätswert für Ihre Viewer auswirken.

MQAR ermöglicht einen nahtlosen Wechsel zwischen den Quellen, sodass Sie einen stabilen, regionsübergreifenden und durchgängigen Workflow für die Medienbereitstellung bereitstellen und AWS Ihren Zuschauern qualitativ hochwertige Inhalte bereitstellen können.

> **Note**
>
> Derzeit unterstützt diese Funktion nur MediaPackage v2-Ursprünge.

Um dieses Feature für Ihre Distribution zu aktivieren, führen Sie die folgenden Schritte aus:

1. Falls Sie dies noch nicht getan haben, erstellen Sie Ihre MediaPackage v2-Ursprünge und aktivieren Sie diese Funktion in Ihrer Endpunktkonfiguration. Für eine regionsübergreifende Bereitstellung erstellen Sie einen sekundären Channel in einem anderen Kanal AWS-Region mit denselben Einstellungen. Weitere Informationen finden Sie in den folgenden Themen im Benutzerhandbuch für AWS Elemental MediaPackage V2:
   - Erstellen eines Kanals und eines Endpunkts
   - Aktivieren des Medienqualitätswerts
2. Um Ihre MediaPackage v2-Ursprünge für eine CloudFront Distribution zu verwenden CloudFront, erstellen oder aktualisieren Sie sie. Siehe Eine Distribution erstellen und Eine Verteilung aktualisieren.
3. Erstellen Sie eine Ursprungsgruppe und wählen Sie Ihre beiden Ursprünge als primären und sekundären Ursprung aus. Aktivieren Sie in Ihrer Ursprungsgruppe die Option Medienqualitätswert. Weitere Informationen finden Sie unter Erstellen einer Ursprungsgruppe.
4. Wählen Sie in dem Cache-Verhalten für Ihre Distribution die von Ihnen erstellte Ursprungsgruppe aus. Wir empfehlen, dass das Cache-Verhalten dem Kanalpfadmuster entspricht.

Wenn CloudFront festgestellt wird, dass beide MediaPackage v2-Ursprünge dieselbe Punktzahl haben, leitet es die Anfrage an den primären Ursprung weiter, der in der Ursprungsgruppe aufgeführt

ist. Wenn der ursprünglich ausgewählte Ursprung mit einem Fehlercode antwortet, der den Failover-Kriterien entspricht, die Sie in Ihrer Ursprungsgruppe angegeben haben, wird CloudFront die Anfrage unabhängig von der Medienqualitätsbewertung erneut an den alternativen Ursprung in Ihrer Ursprungsgruppe gesendet.

Hinweise

- CloudFront verfolgt den Qualitätsfaktor für jedes Cacheverhalten, das eine für die Medienqualitätsbewertung aktivierte Ursprungsgruppe verwendet. Wenn dieselbe Ursprungsgruppe für mehrere Kanäle verwendet wird, die einen Medienqualitätswert ausgeben, erstellen Sie ein separates Cache-Verhalten für das Pfadmuster jedes Kanals, um eine Vermischung der Werte zu vermeiden. Weitere Informationen zu Ursprungsgruppenkontingenten finden Sie unter Allgemeine Kontingente für Verteilungen.
- Derzeit ist MQAR nicht verfügbar, wenn Sie eine Lambda@Edge-Funktion in Auslösern auf Ursprungsseite (Ursprungsanforderung und Ursprungsantwort) verwenden, die mit dem Cache-Verhalten Ihrer Distribution verknüpft sind. Weitere Informationen finden Sie unter Einstellungen für das Cache-Verhalten.
- Wenn Sie die MQAR-Funktion und die Ursprungszugriffssteuerung (OAC) aktiviert haben, fügen Sie die Aktion `mediapackagev2:GetHeadObject` der IAM-Richtlinie hinzu. MQAR benötigt diese Berechtigung, um `HEAD` Anfragen an den MediaPackage v2-Ursprung zu senden. Weitere Informationen zu OAC finden Sie unter Beschränken Sie den Zugriff auf einen AWS Elemental MediaPackage v2-Ursprung.

MQAR-Protokollfelder

CloudFront stellt die folgenden Felder in Echtzeit-Zugriffsprotokollen zur Verfügung, um den Qualitätsfaktor und den ausgewählten Ursprung wiederzugeben. Sie können diese Felder in Ihren Protokollen für CloudFront Echtzeit-Zugriffsprotokolle aktivieren:

- `r-host`
- `sr-reason`
- `x-edge-mqcs`

Weitere Informationen finden Sie unter Felder 65–67.

Anpassen am Edge mit Funktionen

Bei Amazon können Sie Ihren eigenen Code schreiben CloudFront, um anzupassen, wie Ihre CloudFront Distributionen HTTP-Anfragen und -Antworten verarbeiten. Der Code wird in der Nähe Ihrer Zuschauer (Benutzer) ausgeführt, um die Latenz zu minimieren, und Sie müssen keine Server oder andere Infrastruktur verwalten. Sie können Code schreiben, um die durchgehenden Anfragen und Antworten zu manipulieren CloudFront, grundlegende Authentifizierungen und Autorisierungen durchzuführen, HTTP-Antworten am Edge zu generieren und vieles mehr.

Der Code, den Sie schreiben und an Ihre CloudFront Distribution anhängen, wird als Edge-Funktion bezeichnet. CloudFront bietet zwei Möglichkeiten, Edge-Funktionen zu schreiben und zu verwalten:

CloudFront Funktionen

Sie können einfache Funktionen JavaScript für umfangreiche, latenzempfindliche CDN-Anpassungen einschreiben. Die CloudFront Functions-Laufzeitumgebung bietet Startzeiten im Submillisekundenbereich, skaliert sofort, um Millionen von Anfragen pro Sekunde zu verarbeiten, und ist äußerst sicher. CloudFront Functions ist eine systemeigene Funktion von CloudFront, was bedeutet, dass Sie Ihren Code vollständig darin erstellen, testen und bereitstellen können. CloudFront

Lambda@Edge

Lambda@Edge ist eine Erweiterung von AWS Lambda, die leistungsstarke und flexible Datenverarbeitung für komplexe Funktionen und eine vollständige Anwendungslogik bietet, die besser auf die Ihre Viewer eingeht und sehr sicher ist. Lambda @Edge -Funktionen werden in einer Node.js oder einer Python-Laufzeitumgebung ausgeführt. Sie veröffentlichen sie in einer einzigen Version AWS-Region, aber wenn Sie die Funktion einer CloudFront Distribution zuordnen, repliziert Lambda @Edge Ihren Code automatisch auf der ganzen Welt.

CloudFront Funktionen der Verbindung

Sie können einfache JavaScript Funktionen schreiben, die während des mTLS-Verbindungshandshakes ausgeführt werden, um Client-Zertifikate zu validieren, eine benutzerdefinierte Authentifizierungslogik zu implementieren und den Zertifikatssperrvorgang abzuwickeln. Wie CloudFront Funktionen sind auch Verbindungsfunktionen ein systemeigenes Feature CloudFront , das Sie vollständig in ihm erstellen, testen und bereitstellen können. CloudFront Weitere Informationen finden Sie unter Mit CloudFront Verbindungsfunktionen anpassen.

Wenn Sie AWS WAF weitermachen CloudFront, können Sie AWS WAF eingefügte Header sowohl für CloudFront Functions als auch für Lambda @Edge verwenden. Dies funktioniert für Anforderungen und Antworten von Viewern und Ursprüngen.

Themen

- Unterschiede zwischen CloudFront Functions und Lambda @Edge
- Mit CloudFront Funktionen am Rand anpassen
- Mit CloudFront Verbindungsfunktionen anpassen
- Anpassen am Edge mit Lambda@Edge
- Einschränkungen für Edge-Funktionen

Unterschiede zwischen CloudFront Functions und Lambda @Edge

CloudFront Functions und Lambda @Edge bieten beide eine Möglichkeit, Code als Reaktion auf Ereignisse auszuführen. CloudFront

CloudFront Functions ist ideal für einfache, kurz laufende Funktionen für die folgenden Anwendungsfälle:

- Cache-Schlüssel-Normalisierung – Wandeln Sie HTTP-Anforderungseigenschaften (Header, Abfragezeichenfolgen, Cookies und sogar den URL-Pfad) um und verbessern Sie die Cache-Trefferquote mit dem optimalen Cache-Schlüssel.
- Header-Manipulation – HTTP-Header in der Anforderung oder Antwort können eingefügt, verändert oder gelöscht werden. Beispielsweise können Sie jeder Anfrage einen `True-Client-IP`-Header hinzufügen.
- URL-Weiterleitungen oder -Umschreibungen – Leiten Sie Viewer basierend auf den Informationen in der Anforderung auf andere Seiten weiter oder schreiben Sie alle Anforderungen von einem Pfad in einen anderen um.
- Anforderungsberechtigung – Validieren Sie Hash-Autorisierungstoken wie JSON-Web-Token (JWT), indem Sie Autorisierungsheader oder andere Anforderungsmetadaten prüfen.

Informationen zu den ersten Schritten mit CloudFront Funktionen finden Sie unterMit CloudFront Funktionen am Rand anpassen.

Lambda@Edge eignet sich ideal für folgende Anwendungsfälle:

- Funktionen, die mehrere Millisekunden oder mehr benötigen
- Funktionen, die eine einstellbare CPU oder einen einstellbaren Speicher benötigen
- Funktionen, die von Bibliotheken von Drittanbietern abhängen (einschließlich des AWS SDK für die Integration mit anderen AWS-Services)
- Funktionen, die Netzwerkzugriff benötigen, um externe Services für die Verarbeitung zu nutzen
- Funktionen, die Dateisystemzugriff oder Zugriff auf den Text von HTTP-Anforderungen erfordern

Die ersten Schritte mit Lambda@Edge finden Sie unter Anpassen am Edge mit Lambda@Edge.

Um Ihnen bei der Auswahl der Option für Ihren Anwendungsfall zu helfen, verwenden Sie die folgende Tabelle, um die Unterschiede zwischen CloudFront Functions und Lambda @Edge zu verstehen. Informationen zu den Unterschieden, die für die Hilfsmethoden zur Änderung des Ursprungs gelten, finden Sie unter Wählen Sie zwischen CloudFront Functions und Lambda @Edge.

	CloudFront Funktionen	Lambda@Edge
Programmiersprachen	JavaScript (ECMAScript 5.1-konform)	Node.js und Python
Ereignisquellen	• Betrachteranfrage • Betrachterantwort	• Betrachteranfrage • Betrachterantwort • Ursprungsanfrage • Ursprungsantwort
Unterstützt Amazon CloudFront KeyValueStore	Ja CloudFront KeyValueStore unterstützt nur JavaScript Runtime 2.0	Nein
Skalieren	Bis zu Millionen von Anforderungen pro Sekunde	Bis zu 10.000 Anforderungen pro Sekunde pro Region
Dauer der Funktion	Unterhalb einer Millisekunde	Bis zu 30 Sekunden (Zuschaueranfrage und Zuschauerantwort)

	CloudFront Funktionen	Lambda@Edge
		Bis zu 30 Sekunden (Ursprungsanfrage und Ursprungsantwort)
maximale Funktionsspeichergröße	2 MB	128 MB (Zuschauer anfrage und Zuschauer antwort) 10 240 MB (10 GB) (Ursprungsanforderung und Ursprungsantwort) Weitere Informationen finden Sie unter Kontingente für Lambda@Edge.
Maximale Größe des Funktionscodes und der enthaltenen Bibliotheken	10 KB	50 MB (Zuschauer anfrage und Zuschauer antwort) 50 MB (Ursprungsanfrage und Ursprungsantwort)
Netzwerkzugriff	Nein	Ja
Zugriff auf das Dateisystem	Nein	Ja
Zugang zum Anfragetext	Nein	Ja
Zugriff auf Geolokations- und Gerätedaten	Ja	Nein (Viewer-Anforderung und Viewer-Antwort) Ja (Ursprungsanforderung und Ursprungsantwort)

	CloudFront Funktionen	Lambda@Edge
Kann vollständig darin bauen und testen CloudFront	Ja	Nein
Funktionsprotokollierung und Metriken	Ja	Ja

Mit CloudFront Funktionen am Rand anpassen

Mit CloudFront Functions können Sie einfache Funktionen JavaScript für umfangreiche, latenzempfindliche CDN-Anpassungen einschreiben. Ihre Funktionen können die durchgehenden Anfragen und Antworten manipulieren, grundlegende Authentifizierungen und Autorisierungen durchführen CloudFront, HTTP-Antworten am Edge generieren und vieles mehr. Die CloudFront Functions-Laufzeitumgebung bietet Startzeiten im Submillisekundenbereich, skaliert sofort, um Millionen von Anfragen pro Sekunde zu verarbeiten, und ist äußerst sicher. CloudFront Functions ist eine systemeigene Funktion von CloudFront, was bedeutet, dass Sie Ihren Code vollständig darin erstellen, testen und bereitstellen können. CloudFront

Wenn Sie eine CloudFront Funktion mit einer CloudFront Verteilung verknüpfen, CloudFront fängt es Anfragen und Antworten an CloudFront Edge-Standorten ab und leitet sie an Ihre Funktion weiter. Sie können CloudFront Functions aufrufen, wenn die folgenden Ereignisse eintreten:

- Wenn eine Anfrage von einem Zuschauer CloudFront empfangen wird (Viewer-Anfrage)
- Before CloudFront gibt die Antwort an den Zuschauer zurück (Zuschauerantwort)
- Während des TLS-Verbindungsaufbaus (Verbindungsanfrage) — derzeit für gegenseitige TLS-Verbindungen (mTLS) verfügbar

Weitere Informationen zu CloudFront Funktionen finden Sie in den folgenden Themen. Eine Anleitung zum Speichern von Daten in einem Schlüsselwertspeicher finden Sie unterTutorial: Erstellen Sie eine CloudFront Funktion, die Schlüsselwerte enthält. Nachdem Sie eine Funktion geschrieben, getestet und veröffentlicht haben, verknüpfen Sie sie mit einem Cache-Verhalten in Ihrer Distribution — sieheVerknüpfen von Funktionen mit Distributionen.

Themen

- Tutorial: Erstellen Sie eine einfache Funktion mit CloudFront Funktionen
- Tutorial: Erstellen Sie eine CloudFront Funktion, die Schlüsselwerte enthält

- Schreiben von Funktionscode
- Erstellen von Funktionen
- Testfunktionen
- Aktualisieren von Funktionen
- Veröffentlichen von Funktionen
- Verknüpfen von Funktionen mit Distributionen
- Amazon CloudFront KeyValueStore

Tutorial: Erstellen Sie eine einfache Funktion mit CloudFront Funktionen

Dieses Tutorial zeigt Ihnen, wie Sie mit CloudFront Functions beginnen können. Sie können eine einfache Funktion erstellen, die den Viewer auf eine andere URL umleitet und einen benutzerdefinierten Antwortheader zurückgibt.

Inhalt

- Voraussetzungen
- Erstellen der Funktion
- Verifizieren der Funktion

Voraussetzungen

Um CloudFront Functions verwenden zu können, benötigen Sie eine CloudFront Distribution. Falls Sie keines haben, finden Sie weitere Informationen unter Beginnen Sie mit einer CloudFront Standarddistribution.

Erstellen der Funktion

Sie können die CloudFront Konsole verwenden, um eine einfache Funktion zu erstellen, die den Betrachter zu einer anderen URL weiterleitet und außerdem einen benutzerdefinierten Antwortheader zurückgibt.

Um eine CloudFront Funktion zu erstellen

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home.

2. Wählen Sie im linken Navigationsbereich Funktionen und dann Funktion erstellen.
3. Geben Sie auf der Seite Funktion erstellen in das Feld Name einen Funktionsnamen ein, z. *MyFunctionName* B.
4. (Optional) Sie können im Feld Beschreibung eine Beschreibung für die Funktion eingeben, wie z. B. **Simple test function**.
5. Behalten Sie für Runtime die ausgewählte JavaScript Standardversion bei.
6. Wählen Sie Funktion erstellen.
7. Kopieren Sie den folgenden Funktionscode. Dieser Funktionscode leitet den Betrachter auf eine andere URL um und gibt auch einen benutzerdefinierten Antwortheader zurück.

```
function handler(event) {
    // NOTE: This example function is for a viewer request event trigger.
    // Choose viewer request for event trigger when you associate this function
 with a distribution.
    var response = {
        statusCode: 302,
        statusDescription: 'Found',
        headers: {
            'cloudfront-functions': { value: 'generated-by-CloudFront-Functions' },
            'location': { value: 'https://aws.amazon.com/cloudfront/' }
        }
    };
    return response;
}
```

8. Fügen Sie für Funktionscode den Code in den Code-Editor ein, um den Standardcode zu ersetzen.
9. Wählen Sie Änderungen speichern aus.
10. (Optional) Sie können die Funktion testen, bevor Sie sie veröffentlichen. In diesem Tutorial wird nicht beschrieben, wie eine Funktion getestet wird. Weitere Informationen finden Sie unter Testfunktionen.
11. Wählen Sie die Registerkarte Veröffentlichen und dann Funktion veröffentlichen aus. Sie müssen die Funktion veröffentlichen, bevor Sie sie Ihrer CloudFront Distribution zuordnen können.
12. Als Nächstes können Sie die Funktion einer Distribution oder einem Cacheverhalten zuordnen. Wählen Sie auf der *MyFunctionName* Seite die Registerkarte Veröffentlichen aus.

> **Warning**
>
> In den folgenden Schritten wählen Sie eine Distribution oder ein Cacheverhalten zum Testen. Verknüpfen Sie diese Testfunktion nicht mit einer Distribution oder einem Cacheverhalten, die bzw. das in der Produktion verwendet wird.

13. Wählen Sie Add association.
14. Wählen Sie im Dialogfeld Zuordnen eine Distribution und/oder ein Cacheverhalten aus. Behalten Sie im Feld Ereignistyp den Standardwert bei.
15. Wählen Sie Add association.

    In der Tabelle Zugeordnete Verteilungen wird die zugeordnete Verteilung angezeigt.
16. Warten Sie ein paar Minuten, bis die zugehörige Verteilung die Bereitstellung abgeschlossen hat. Um den Status der Distribution zu überprüfen, wählen Sie die Distribution in der Tabelle Zugeordnete Distributionen und dann Distribution anzeigen aus.

    Wenn der Status der Verteilung Bereitgestellt lautet, können Sie überprüfen, ob die Funktion funktioniert.

Verifizieren der Funktion

Nachdem Sie die Funktion bereitgestellt haben, können Sie überprüfen, ob sie für Ihre Distribution funktioniert.

So verifizieren Sie die Funktion

1. Navigieren Sie im Webbrowser zum Domainnamen Ihrer Distribution (z. B. `https://d111111abcdef8.cloudfront.net`).

   Die Funktion gibt eine Umleitung an den Browser zurück, sodass der Browser automatisch an weitergeleitet wir `https://aws.amazon.com/cloudfront/`.
2. In einem Befehlszeilenfenster können Sie ein Tool wie curl verwenden, um eine Anforderung an den Domainnamen Ihrer Distribution zu senden.

   ```
   curl -v https://d111111abcdef8.cloudfront.net/
   ```

In der Antwort werden die Umleitungsantwort (`302 Found`) und die benutzerdefinierten Antwortheader, die die Funktion hinzugefügt hat, angezeigt. Das folgende Beispiel zeigt, wie die Anforderung aussehen kann.

Example

```
curl -v https://d111111abcdef8.cloudfront.net/
> GET / HTTP/1.1
> Host: d111111abcdef8.cloudfront.net
> User-Agent: curl/7.64.1
> Accept: */*
>
< HTTP/1.1 302 Found
< Server: CloudFront
< Date: Tue, 16 Mar 2021 18:50:48 GMT
< Content-Length: 0
< Connection: keep-alive
< Location: https://aws.amazon.com/cloudfront/
< Cloudfront-Functions: generated-by-CloudFront-Functions
< X-Cache: FunctionGeneratedResponse from cloudfront
< Via: 1.1 3035b31bddaf14eded329f8d22cf188c.cloudfront.net (CloudFront)
< X-Amz-Cf-Pop: PHX50-C2
< X-Amz-Cf-Id: ULZdIz6j43uGBlXyob_JctF9x7CCbwpNniiMlmNbmwzH1YWP9FsEHg==
```

Tutorial: Erstellen Sie eine CloudFront Funktion, die Schlüsselwerte enthält

Dieses Tutorial zeigt Ihnen, wie Sie Schlüsselwerte in CloudFront Funktionen einbeziehen. Schlüsselwerte sind Teil eines Schlüssel-Wert-Paars. Sie nehmen den Namen (aus dem Schlüssel-Wert-Paar) in den Funktionscode auf. Wenn die Funktion ausgeführt wird, wird der Name durch den Wert CloudFront ersetzt.

Schlüssel-Wert-Paare sind Variablen, die in einem Schlüsselwertspeicher gespeichert werden. Wenn Sie in Ihrer Funktion einen Schlüssel verwenden (anstelle von hartkodierten Werten), ist Ihre Funktion flexibler. Sie können den Wert des Schlüssels ändern, ohne Codeänderungen vornehmen zu müssen. Schlüssel-Wert-Paare können auch die Größe Ihrer Funktion reduzieren. Weitere Informationen finden Sie unter ???.

Inhalt

- Voraussetzungen

- Erstellen des Schlüsselwertspeichers
- Hinzufügen von Schlüssel-Wert-Paaren zum Schlüsselwertspeicher
- Zuordnen des Schlüsselwertspeichers zur Funktion
- Testen und Veröffentlichen des Funktionscodes

Voraussetzungen

Wenn Sie mit CloudFront Funktionen und dem Schlüsselwertspeicher noch nicht vertraut sind, empfehlen wir Ihnen, das Tutorial unter zu befolgenthe section called “Tutorial: Erstelle eine einfache CloudFront Funktion”.

Nachdem Sie das Tutorial abgeschlossen haben, können Sie diesem Tutorial folgen, um die von Ihnen erstellte Funktion zu erweitern. Für dieses Tutorial empfehlen wir, dass Sie zuerst den Schlüsselwertspeicher erstellen.

Erstellen des Schlüsselwertspeichers

Erstellen Sie zunächst den Schlüsselwertspeicher, der für Ihre Funktion verwendet werden soll.

So erstellen Sie den Schlüsselwertspeicher

1. Planen Sie die Schlüssel-Wert-Paare, die Sie in die Funktion aufnehmen möchten. Notieren Sie sich die Schlüsselnamen. Die Schlüssel-Wert-Paare, die Sie in einer Funktion verwenden möchten, müssen sich in einem einzigen Schlüsselwertspeicher befinden.
2. Entscheiden Sie sich, in welcher Reihenfolge Sie vorgehen möchten. Es gibt zwei Vorgehensweisen:
   - Erstellen Sie einen Schlüsselwertspeicher und fügen Sie dem Speicher Schlüssel-Wert-Paare hinzu. Dann die Funktion erstellen (oder ändern) und die Schlüsselnamen integrieren.
   - Oder: Die Funktion erstellen (oder ändern) und die zu verwendenden Schlüsselnamen integrieren. Erstellen Sie dann einen Schlüsselwertspeicher und fügen Sie die Schlüssel-Wert-Paare hinzu.
3. Melden Sie sich bei an AWS-Managementkonsole und öffnen Sie die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home.
4. Wählen Sie im Navigationsbereich Funktionen und dann die KeyValueStoresRegisterkarte aus.
5. Wählen Sie Erstellen KeyValueStore und geben Sie die folgenden Felder ein:

- Geben Sie für den Speicher einen Namen und (optional) eine Beschreibung ein.
- Lassen Sie den S3-URI leer. In diesem Tutorial geben Sie die Schlüssel-Wert-Paare manuell ein.

6. Wählen Sie Erstellen aus. Die Seite mit den Details für den neuen Schlüsselwertspeicher wird angezeigt. Diese Seite enthält den Abschnitt Schlüssel-Wert-Paare, der derzeit leer ist.

Hinzufügen von Schlüssel-Wert-Paaren zum Schlüsselwertspeicher

Als Nächstes fügen Sie dem zuvor erstellten Schlüssel-Wert-Speicher manuell eine Liste von Schlüssel-Wert-Paaren hinzu.

So fügen Sie dem Schlüsselwertspeicher Schlüssel-Wert-Paare hinzu

1. Wählen Sie im Abschnitt Schlüssel-Wert-Paare die Option Schlüssel-Wert-Paare hinzufügen aus.
2. Wählen Sie Paar hinzufügen und geben Sie dann einen Namen und einen Wert ein. Wählen Sie das Häkchen aus, um Ihre Änderungen zu bestätigen, und wiederholen Sie diesen Schritt, um weitere Paare hinzuzufügen.
3. Wenn Sie fertig sind, wählen Sie Änderungen speichern, um die Schlüssel-Wert-Paare im Schlüsselwertspeicher zu speichern. Wählen Sie im Bestätigungsdialogfeld die Option Fertig aus.

Sie haben jetzt einen Schlüsselwertspeicher, der eine Gruppe von Schlüssel-Wert-Paaren enthält.

Zuordnen des Schlüsselwertspeichers zur Funktion

Sie haben jetzt den Schlüsselwertspeicher erstellt. Weiterhin haben Sie eine Funktion erstellt oder geändert, die die Schlüsselnamen aus dem Schlüsselwertspeicher enthält. Sie können jetzt den Schlüsselwertspeicher und die Funktion verknüpfen. Sie erstellen diese Zuordnung innerhalb der Funktion.

So ordnen Sie den Schlüsselwertspeicher der Funktion zu

1. Wählen Sie im Navigationsbereich Funktionen aus. Die Registerkarte Funktionen wird standardmäßig oben angezeigt.
2. Wählen Sie den Funktionsnamen und wählen Sie im KeyValueStore Bereich Zugeordnet die Option Vorhandenes zuordnen aus KeyValueStore.

3. Wählen Sie den Schlüsselwertspeicher aus und klicken Sie auf Zuordnen KeyValueStore.

> **Note**
>
> Sie können jeder Funktion nur einen Schlüsselwertspeicher zuordnen.

Testen und Veröffentlichen des Funktionscodes

Nachdem Sie den Schlüsselwertspeicher mit Ihrer Funktion verknüpft haben, können Sie den Funktionscode testen und veröffentlichen. Sie sollten den Funktionscode jedes Mal testen, wenn Sie ihn ändern, auch wenn Sie Folgendes tun:

- Der Funktion einen Schlüsselwertspeicher zuordnen.
- die Funktion und ihren Schlüsselwertspeicher so ändern, dass sie ein neues Schlüssel-Wert-Paar enthalten
- den Wert eines Schlüssel-Wert-Paares ändern

So testen und veröffentlichen Sie den Funktionscode

1. Weitere Informationen zum Testen einer Funktion finden Sie unter the section called “Testfunktionen”. Stellen Sie sicher, dass Sie die Funktion in der `DEVELOPMENT`-Phase testen.
2. Veröffentlichen Sie die Funktion, wenn Sie bereit sind, die Funktion (mit den neuen oder überarbeiteten Schlüssel-Wert-Paaren) in einer `LIVE`-Umgebung zu verwenden.

   Beim Veröffentlichen wird die Version der Funktion von der `DEVELOPMENT` Bühne in die Live-Phase CloudFront kopiert. Die Funktion hat den neuen Code und ist dem Schlüsselwertspeicher zugeordnet. (Die Zuordnung muss in der Live-Phase nicht erneut ausgeführt werden.)

   Weitere Informationen zum Veröffentlichen einer Funktion finden Sie unter the section called “Veröffentlichen von Funktionen”.

Schreiben von Funktionscode

Sie können CloudFront Functions verwenden, um einfache Funktionen JavaScript für umfangreiche, latenzempfindliche CDN-Anpassungen zu schreiben. Ihr Funktionscode kann die Anfragen und

Antworten bearbeiten, die durchfließen CloudFront, grundlegende Authentifizierung und Autorisierung durchführen, HTTP-Antworten am Edge generieren und vieles mehr.

Informationen zum Schreiben von Funktionscode für CloudFront Funktionen finden Sie in den folgenden Themen. Codebeispiele finden Sie unter CloudFront Funktionen, Beispiele für CloudFront und im Amazon-Cloudfront-Functions-Repository auf. GitHub

Informationen zur Verwendung von CBOR Web Tokens (CWT) für die tokenbasierte Authentifizierung und Autorisierung an Edge-Standorten finden Sie unter. CloudFront CWT-Unterstützung für Funktionen CloudFront

Themen

- Ermitteln des Zwecks Ihrer Funktion
- CloudFront Funktionen, Eventstruktur
- JavaScript Laufzeitfunktionen für CloudFront Funktionen
- Hilfsmethoden für Schlüsselwertspeicher
- Hilfsmethoden für die Änderung des Ursprungs
- Hilfsmethoden für CloudFront SaaS Manager-Eigenschaften
- Verwenden Sie async und await
- CWT-Unterstützung für Funktionen CloudFront
- Allgemeine Hilfsmethoden
- Hilfsmethoden für gegenseitiges TLS

Ermitteln des Zwecks Ihrer Funktion

Bevor Sie Ihren Funktionscode schreiben, bestimmen Sie den Zweck Ihrer Funktion. Die meisten CloudFront Funktionen in Functions haben einen der folgenden Zwecke.

Themen

- Ändern der HTTP-Anforderung in einem Viewer-Anforderungsereignistyp
- Generieren einer HTTP-Antwort in einem Viewer-Anforderungsereignistyp
- Ändern der HTTP-Antwort in einem Viewer-Antwortereignistyp
- Überprüfen Sie mTLS-Verbindungen in einem Ereignistyp für Verbindungsanfragen
- Ähnliche Informationen

Unabhängig vom Zweck Ihrer Funktion ist `handler` der Einstiegspunkt für jede Funktion. Es benötigt ein einzelnes aufgerufenes Argument`event`, das an die Funktion von übergeben wird CloudFront. `event` ist ein JSON-Objekt, das eine Darstellung der HTTP-Anfrage enthält (und der Antwort, wenn Ihre Funktion die HTTP-Antwort ändert).

Ändern der HTTP-Anforderung in einem Viewer-Anforderungsereignistyp

Ihre Funktion kann die HTTP-Anfrage ändern, die vom Viewer (Client) CloudFront empfangen wird, und die geänderte Anfrage CloudFront zur weiteren Verarbeitung an sie zurücksenden. Beispielsweise könnte Ihr Funktionscode den Cache-Schlüssel normalisieren oder Anforderungs-Header ändern.

Nachdem Sie eine Funktion erstellt und veröffentlicht haben, die die HTTP-Anforderung ändert, achten Sie darauf, eine Zuordnung zum Ereignistyp der Viewer-Anforderung hinzuzufügen. Weitere Informationen finden Sie unter Erstellen der Funktion. Dadurch wird die Funktion jedes Mal ausgeführt, wenn CloudFront eine Anfrage von einem Viewer eingeht, bevor überprüft wird, ob sich das angeforderte Objekt im CloudFront Cache befindet.

Example Beispiel

Der folgende Pseudocode zeigt die Struktur einer Funktion, die die HTTP-Anfrage ändert.

```
function handler(event) {
    var request = event.request;

    // Modify the request object here.

    return request;
}
```

Die Funktion gibt das geänderte `request` Objekt an zurück CloudFront. CloudFrontsetzt die Verarbeitung der zurückgegebenen Anfrage fort, indem sie den CloudFront Cache auf einen Cache-Treffer überprüft und die Anfrage gegebenenfalls an den Ursprung sendet.

Generieren einer HTTP-Antwort in einem Viewer-Anforderungsereignistyp

Ihre Funktion kann am Edge eine HTTP-Antwort generieren und sie direkt an den Viewer (Client) zurückgeben, ohne nach einer zwischengespeicherten Antwort zu suchen oder sie weiter zu verarbeiten. CloudFront Beispielsweise könnte Ihr Funktionscode die Anfrage an eine neue URL umleiten oder nach Autorisierung suchen und eine `401`- oder `403`-Antwort auf nicht autorisierte Anfragen zurückgeben.

Wenn Sie eine Funktion erstellen, die eine HTTP-Antwort generiert, achten Sie darauf, den Ereignistyp der Betrachteranfrage auszuwählen. Das bedeutet, dass die Funktion jedes Mal ausgeführt wird, wenn sie eine Anfrage von einem Viewer CloudFront erhält, bevor CloudFront die Anfrage weiter verarbeitet wird.

Example Beispiel

Der folgende Pseudocode zeigt die Struktur einer Funktion, die eine HTTP-Antwort generiert.

```
function handler(event) {
    var request = event.request;

    var response = ...; // Create the response object here,
                        // using the request properties if needed.

    return response;
}
```

Die Funktion gibt ein `response` Objekt an zurück CloudFront, das CloudFront sofort zum Viewer zurückkehrt, ohne den CloudFront Cache zu überprüfen oder eine Anfrage an den Ursprung zu senden.

Ändern der HTTP-Antwort in einem Viewer-Antwortereignistyp

Ihre Funktion kann die HTTP-Antwort ändern, bevor sie CloudFront an den Viewer (Client) gesendet wird, unabhängig davon, ob die Antwort aus dem CloudFront Cache oder vom Ursprung stammt. Beispielsweise könnte Ihr Funktionscode Antwortheader, Statuscodes oder Textinhalte hinzufügen oder ändern.

Wenn Sie eine Funktion erstellen, die die HTTP-Antwort ändert, achten Sie darauf, den Ereignistyp Betrachterantwort zu wählen. Das bedeutet, dass die Funktion ausgeführt wird, bevor sie eine Antwort an den Viewer CloudFront zurückgibt, unabhängig davon, ob die Antwort aus dem CloudFront Cache oder vom Ursprung stammt.

Example Beispiel

Der folgende Pseudocode zeigt die Struktur einer Funktion, die die HTTP-Antwort ändert.

```
function handler(event) {
    var request = event.request;
```

```
    var response = event.response;

    // Modify the response object here,
    // using the request properties if needed.

    return response;
}
```

Die Funktion gibt das geänderte `response` Objekt an zurück CloudFront, das CloudFront sofort zum Viewer zurückkehrt.

Überprüfen Sie mTLS-Verbindungen in einem Ereignistyp für Verbindungsanfragen

Verbindungsfunktionen sind Funktionen, die während TLS-Verbindungen ausgeführt werden, um eine benutzerdefinierte Validierungs- und Authentifizierungslogik bereitzustellen. CloudFront Verbindungsfunktionen sind derzeit für Mutual TLS (mTLS) -Verbindungen verfügbar, mit denen Sie Client-Zertifikate validieren und eine benutzerdefinierte Authentifizierungslogik implementieren können, die über die standardmäßige Zertifikatsvalidierung hinausgeht. Verbindungsfunktionen werden während des TLS-Handshake-Prozesses ausgeführt und können Verbindungen auf der Grundlage von Zertifikatseigenschaften, Client-IP-Adressen oder anderen Kriterien zulassen oder verweigern.

Nachdem Sie eine Verbindungsfunktion erstellt und veröffentlicht haben, stellen Sie sicher, dass Sie eine Zuordnung für den Ereignistyp der Verbindungsanforderung mit einer TLS-enabled M-Verteilung hinzufügen. Dadurch wird die Funktion jedes Mal ausgeführt, wenn ein Client versucht, eine mTLS-Verbindung mit CloudFront herzustellen.

Example

Der folgende Pseudocode zeigt die Struktur einer Verbindungsfunktion:

```
function connectionHandler(connection) {
    // Validate certificate and connection properties here.

    if (/* validation passes */) {
        connection.allow();
    } else {
        connection.deny();
    }
}
```

Die Funktion verwendet Hilfsmethoden, um zu bestimmen, ob die Verbindung zugelassen oder verweigert werden soll. Im Gegensatz zu den Funktionen Viewer-Request und Viewer-Response können Verbindungsfunktionen HTTP-Anfragen oder -Antworten nicht ändern.

Ähnliche Informationen

Weitere Informationen zum Arbeiten mit CloudFront Funktionen finden Sie in den folgenden Themen:

- Ereignisstruktur
- JavaScript Laufzeitfunktionen
- CloudFront Funktionen, Beispiele
- Einschränkungen für Edge-Funktionen

CloudFront Funktionen, Eventstruktur

CloudFront Functions übergibt ein `event` Objekt als Eingabe an Ihren Funktionscode, wenn die Funktion ausgeführt wird. Wenn Sie eine Funktion testen, erstellen Sie das `event`-Objekt und übergeben es an Ihre Funktion. Wenn Sie ein `event`-Objekt zum Testen einer Funktion erstellen, können Sie die Felder `distributionDomainName`, `distributionId` und `requestId` im `context`-Objekt weglassen. Stellen Sie sicher, dass die Namen der Header in Kleinbuchstaben geschrieben sind, was bei dem `event` Objekt, das CloudFront Functions in der Produktion an Ihre Funktion übergibt, immer der Fall ist.

Im Folgenden wird ein Überblick über die Struktur dieses Ereignisobjekts gegeben.

```
{
    "version": "1.0",
    "context": {
        <context object>
    },
    "viewer": {
        <viewer object>
    },
    "request": {
        <request object>
    },
    "response": {
        <response object>
    }
```

```
}
```

Weitere Informationen finden Sie unter den folgenden Themen:

Themen

- Feld Version
- Context-Objekt
- Struktur des Verbindungsereignisses
- Betrachterobjekt
- Objekt anfordern
- Antwortobjekt
- Statuscode und Text
- Struktur von Abfragezeichenfolge, Header und Cookie
- Beispiel für Antwortobjekt
- Beispiel für Ereignisobjekt

Feld Version

Das `version` Feld enthält eine Zeichenfolge, die die Version des CloudFront Functions-Ereignisobjekts angibt. Die aktuelle Version ist `1.0`.

Context-Objekt

Das `context`-Objekt enthält kontextbezogene Informationen über das Ereignis. Er enthält folgende Felder:

`distributionDomainName`

Der CloudFront Domänenname (z. B. d111111abcdef8.cloudfront.net) der Standarddistribution, die dem Ereignis zugeordnet ist.

Das Feld `distributionDomainName` wird nur angezeigt, wenn Ihre Funktion für Standarddistributionen aufgerufen wird.

`endpoint`

Der CloudFront Domänenname (z. B. d111111abcdef8.cloudfront.net) der Verbindungsgruppe, die dem Ereignis zugeordnet ist.

Das Feld `endpoint` wird nur angezeigt, wenn Ihre Funktion für Multi-Tenant-Distributionen aufgerufen wird.

`distributionId`

Die ID der Verteilung (zum Beispiel EDFDVBD6EXAMPLE), die mit dem Ereignis verknüpft ist.

`eventType`

Der Ereignistyp, entweder `viewer-request` oder `viewer-response`.

`requestId`

Eine Zeichenfolge, die eine Anfrage (und die zugehörige Antwort) eindeutig identifiziert. CloudFront

Struktur des Verbindungsereignisses

Verbindungsfunktionen erhalten eine andere Ereignisstruktur als Viewer-Funktionen. Detaillierte Informationen zur Verbindungsereignisstruktur und zum Antwortformat finden Sie unterOrdnen Sie eine CloudFront Verbindungsfunktion zu.

Betrachterobjekt

Das `viewer`-Objekt enthält ein `ip`-Feld, dessen Wert die IP-Adresse des Betrachters (Clients) ist, der die Anfrage gesendet hat. Wenn der Betrachter einen HTTP-Proxy oder einen Load Balancer verwendet hat, um die Anfrage zu senden, entspricht der Wert der IP-Adresse des Proxys bzw. des Load Balancers.

Objekt anfordern

Das `request` Objekt enthält eine Repräsentation einer CloudFront Viewer-to-HTTP-Anfrage. In dem `event` Objekt, das an Ihre Funktion übergeben wird, stellt das `request` Objekt die eigentliche Anfrage dar, die vom Viewer CloudFront empfangen wurde.

Wenn Ihr Funktionscode ein `request` Objekt an zurückgibt CloudFront, muss es dieselbe Struktur verwenden.

Das `request`-Objekt enthält die folgenden Felder:

`method`

Die HTTP-Methode der Anforderung. Wenn Ihr Funktionscode eine `request` zurückgibt, kann er dieses Feld nicht ändern. Dies ist das einzige schreibgeschützte Feld im `request`-Objekt.

`uri`

Der relative Pfad des angeforderten Objekts.

> **Note**
>
> Wenn Ihre Funktion den `uri`-Wert ändert, gilt Folgendes:
>
> - Der neue `uri`-Wert muss mit einem Schrägstrich (/) beginnen (`/`).
> - Wenn eine Funktion den `uri`-Wert ändert, ändert sie auch das Objekt, das die Betrachter anfordert.
> - Wenn eine Funktion den `uri`-Wert ändert, wird weder das Cache-Verhalten für die Anforderung noch der Ursprung geändert, an den die Ursprungsanfrage weitergeleitet wird.

`querystring`

Ein Objekt, das die Abfragezeichenkette in der Anfrage darstellt. Wenn die Anfrage keine Abfragezeichenfolge enthält, enthält das `request`-Objekt immer noch ein leeres `querystring`-Objekt.

Das `querystring`-Objekt enthält ein Feld für jeden Abfragezeichenfolgenparameter in der Anforderung.

`headers`

Ein Objekt, das den HTTP-Header in der Anforderung darstellt. Wenn die Anfrage `Cookie`-Header enthält, sind diese Header nicht Teil des `headers`-Objekts. Cookies werden im `cookies`-Objekt separat dargestellt.

Das `headers`-Objekt enthält ein Feld für jeden Header in der Anfrage. Header-Namen werden ASCII-lowercase in das Event-Objekt umgewandelt, und Header-Namen müssen so sein, wie ASCII-lowercase sie durch Ihren Funktionscode hinzugefügt werden. Wenn CloudFront Functions das Ereignisobjekt wieder in eine HTTP-Anfrage konvertiert, wird der erste Buchstabe jedes Worts in Header-Namen groß geschrieben, falls es sich um eine ASCII-letter handelt. CloudFront Functions wendet keine Änderungen an Nicht-ASCII-Symbolen in Header-Namen an. Beispielsweise wird `TÈst-header` innerhalb der Funktion zu `tÈst-header`. Das Nicht-ASCII-Symbol È bleibt unverändert.

Wörter werden durch einen Bindestrich (-) getrennt. Wenn Ihr Funktionscode beispielsweise einen Header namens`example-header-name`, CloudFront konvertiert diesen `Example-Header-Name` in die HTTP-Anfrage.

`cookies`

Ein Objekt, das die Cookies in der Anfrage darstellt (`Cookie`-Header).

Das `cookies`-Objekt enthält ein Feld für jeden Cookie in der Anfrage.

Weitere Hinweise zur Struktur von Abfragezeichenfolgen, Headern und Cookies finden Sie unter Struktur von Abfragezeichenfolge, Header und Cookie.

Ein `event`-Beispielobjekt finden Sie unter Beispiel für Ereignisobjekt.

Antwortobjekt

Das `response` Objekt enthält eine Darstellung einer CloudFront-to-viewer HTTP-Antwort. In dem `event` Objekt, das an Ihre Funktion übergeben wird, stellt das `response` Objekt die CloudFront tatsächliche Antwort auf eine Viewer-Anfrage dar.

Wenn Ihr Funktionscode ein `response`-Objekt zurückgibt, muss er dieselbe Struktur verwenden.

Das `response`-Objekt enthält die folgenden Felder:

`statusCode`

Den HTTP-Statuscode der Antwort. Dieser Wert ist eine Ganzzahl, keine Zeichenfolge.

Ihre Funktion kann den `statusCode` generieren oder ändern.

`statusDescription`

Die HTTP-Statusbeschreibung der Antwort. Wenn Ihr Funktionscode eine Antwort generiert, ist dieses Feld optional.

`headers`

Ein Objekt, das die HTTP-Header in der Antwort darstellt. Wenn die Antwort `Set-Cookie` Header enthält, sind diese Header nicht Teil des `headers`-Objekts. Cookies werden im `cookies`-Objekt separat dargestellt.

Das `headers`-Objekt enthält ein Feld für jeden Header in der Antwort. Headernamen werden im Ereignisobjekt in Kleinbuchstaben konvertiert und Headernamen müssen in Kleinbuchstaben

stehen, wenn sie vom Funktionscode hinzugefügt werden. Wenn CloudFront Functions das Ereignisobjekt wieder in eine HTTP-Antwort konvertiert, wird der erste Buchstabe jedes Worts in Header-Namen groß geschrieben. Wörter werden durch einen Bindestrich (`-`) getrennt. Wenn Ihr Funktionscode beispielsweise einen Header namens`example-header-name`, CloudFront konvertiert diesen `Example-Header-Name` in die HTTP-Antwort.

`cookies`

Ein Objekt, das die Cookies in der Antwort darstellt (`Set-Cookie`-Header).

Das `cookies`-Objekt enthält ein Feld für jedes Cookie in der Antwort.

`body`

Das Hinzufügen des `body`-Felds ist optional. Es wird im `response`-Objekt nur dann vorhanden sein, wenn Sie es in Ihrer Funktion angeben. Ihre Funktion hat keinen Zugriff auf den ursprünglichen Text, der vom CloudFront Cache oder vom Ursprung zurückgegeben wird. Wenn Sie das `body` Feld in Ihrer Viewer-Antwortfunktion nicht angeben, wird der vom CloudFront Cache oder Ursprung zurückgegebene Originaltext an den Viewer zurückgegeben.

Wenn Sie einen benutzerdefinierten Text CloudFront an den Viewer zurückgeben möchten, geben Sie den Inhalt des Textkörpers im `data` Feld und die Textkodierung im `encoding` Feld an. Sie können die Kodierung als Klartext (`"encoding": "text"`) oder als Base64-encoded Inhalt (`"encoding": "base64"`) angeben.

Als Abkürzung können Sie den Textinhalt auch direkt im Feld `body` angeben (`"body": "<specify the body content here>"`). Lassen Sie dabei die `encoding` Felder `data` und weg. CloudFront behandelt den Text in diesem Fall als Klartext.

`encoding`

Die Codierung für den `body`-Inhalt (Feld `data`). Die einzigen gültigen Codierungen sind `text` und `base64`.

Wenn Sie `encoding` als angeben, der `base64` Text aber kein gültiges Base64-Format ist, wird ein Fehler CloudFront zurückgegeben.

`data`

Der `body`-Inhalt.

Weitere Informationen zu geänderten Statuscodes und Textinhalten finden Sie unter Statuscode und Text.

Weitere Informationen zur Struktur von Headern und Cookies finden Sie unter Struktur von Abfragezeichenfolge, Header und Cookie.

Ein `response`-Beispielobjekt finden Sie unter Beispiel für Antwortobjekt.

Statuscode und Text

Mit CloudFront Functions können Sie den Statuscode der Zuschauerantwort aktualisieren, den gesamten Antworttext durch einen neuen ersetzen oder den Antworttext entfernen. Zu den häufigsten Szenarien für die Aktualisierung der Zuschauerantwort, nachdem Aspekte der Antwort aus dem CloudFront Cache oder der Quelle bewertet wurden, gehören die folgenden:

- Ändern des Status, um einen HTTP-200-Statuscode festzulegen, und Erstellen statischer Textinhalte für die Rückgabe an den Viewer.
- Ändern des Status, um einen HTTP-301- oder -302-Statuscode festzulegen, der den Benutzer auf eine andere Website umleitet.
- Entscheiden, ob der Text der Viewer-Antwort weitergeleitet oder verworfen werden soll.

> **Note**
>
> Wenn der Ursprung einen HTTP-Fehler von 400 und höher zurückgibt, wird die CloudFront Funktion nicht ausgeführt. Weitere Informationen finden Sie unter Einschränkungen für alle Edge-Funktionen.

Wenn Sie mit der HTTP-Antwort arbeiten, hat CloudFront Functions keinen Zugriff auf den Antworttext. Sie können den Textinhalt ersetzen, indem Sie ihn auf den gewünschten Wert setzen, oder den Text entfernen, indem Sie den Wert auf leer setzen. Wenn Sie das Textfeld in Ihrer Funktion nicht aktualisieren, wird der ursprüngliche Text, der vom CloudFront Cache oder Ursprung zurückgegeben wurde, an den Viewer zurückgegeben.

> **Tip**
>
> Wenn Sie CloudFront Functions verwenden, um einen Hauptteil zu ersetzen, achten Sie darauf, die entsprechenden Header, z. B.`content-encoding`, oder `content-typecontent-length`, am neuen Textinhalt auszurichten.
> Wenn beispielsweise der CloudFront Ursprung oder der Cache zurückkehrt, `content-encoding: gzip` aber die Antwortfunktion des Viewers einen Textkörper festlegt, der

aus reinem Text besteht, muss die Funktion auch die `content-type` Header `content-encoding` und entsprechend ändern.

Wenn Ihre CloudFront Funktion so konfiguriert ist, dass sie einen HTTP-Fehler von 400 oder höher zurückgibt, wird Ihrem Betrachter keine benutzerdefinierte Fehlerseite angezeigt, die Sie für denselben Statuscode angegeben haben.

Struktur von Abfragezeichenfolge, Header und Cookie

Abfragezeichenfolgen, Header und Cookies haben dieselbe Struktur. Abfragezeichenfolgen können in Anforderungen vorkommen. Header erscheinen in Anforderungen und Antworten. Cookies erscheinen in Anforderungen und Antworten.

Jede Abfragezeichenfolge, jeder Header oder jedes Cookie ist ein eindeutiges Feld innerhalb des übergeordneten `querystring`-, `headers`- oder `cookies`-Objekts. Der Feldname ist der Name der Abfragezeichenfolge, des Headers oder des Cookies. Jedes Feld enthält eine `value`-Eigenschaft mit dem Wert der Abfragezeichenfolge, des Headers oder des Cookies.

Inhalt

- Werte oder Objekte von Abfragezeichenfolgen
- Besondere Überlegungen für Header
- Doppelte Abfragezeichenfolgen, Header und Cookies (MultiValue-Array)
- Cookie-Attribute

Werte oder Objekte von Abfragezeichenfolgen

Eine Funktion kann zusätzlich zum Objekt den Wert einer Abfragezeichenfolge zurückgeben. Der Wert der Abfragezeichenfolge kann verwendet werden, um die Parameter der Abfragezeichenfolge in beliebiger benutzerdefinierter Reihenfolge anzuordnen.

Example Beispiel

Um eine Abfragezeichenfolge in Ihrem Funktionscode zu ändern, verwenden Sie einen Code wie den folgenden:

```
var request = event.request;
```

```
request.querystring =
 'ID=42&Exp=1619740800&TTL=1440&NoValue=&querymv=val1&querymv=val2,val3';
```

Besondere Überlegungen für Header

Nur für Header werden Headernamen im Ereignisobjekt in Kleinbuchstaben konvertiert und Headernamen müssen Kleinbuchstaben aufweisen, wenn sie vom Funktionscode hinzugefügt werden. Wenn CloudFront Functions das Event-Objekt wieder in eine HTTP-Anfrage oder -Antwort konvertiert, wird der erste Buchstabe jedes Worts in Header-Namen groß geschrieben. Wörter werden durch einen Bindestrich (`-`) getrennt. Wenn Ihr Funktionscode beispielsweise einen Header namens`example-header-name`, CloudFront konvertiert diesen `Example-Header-Name` in die HTTP-Anfrage oder -Antwort.

Example Beispiel

Betrachten Sie den folgenden `Host`-Header in einer HTTP-Anforderung.

```
Host: video.example.com
```

Dieser Header wird im `request`-Objekt wie folgt dargestellt:

```
"headers": {
    "host": {
        "value": "video.example.com"
    }
}
```

Um auf den `Host`-Header in Ihrem Funktionscode zuzugreifen, verwenden Sie Code wie den folgenden:

```
var request = event.request;
var host = request.headers.host.value;
```

Um einen Header in Ihrem Funktionscode hinzuzufügen oder zu ändern, verwenden Sie Code wie den folgenden (dieser Code fügt einen Header mit Namen `X-Custom-Header` und Wert `example value` hinzu):

```
var request = event.request;
request.headers['x-custom-header'] = {value: 'example value'};
```

Doppelte Abfragezeichenfolgen, Header und Cookies (MultiValue-Array)

Eine HTTP-Anfrage oder Antwort kann mehr als eine Abfragezeichenfolge, einen Header oder ein Cookie mit demselben Namen enthalten. In diesem Fall sind die doppelten Abfragezeichenfolgen, -Header oder -Cookies in einem Feld im `request`- oder `response`-Objekt zusammengefasst, aber dieses Feld enthält eine zusätzliche Eigenschaft namens `multiValue`. Die `multiValue`-Eigenschaft enthält ein Array mit den Werten der doppelten Abfragezeichenfolgen, Header oder Cookies.

Example Beispiel

Betrachten Sie eine HTTP-Anforderung mit den folgenden `Accept`-Headern.

```
Accept: application/json
Accept: application/xml
Accept: text/html
```

Diese Header werden im `request`-Objekt wie folgt dargestellt.

```
"headers": {
    "accept": {
        "value": "application/json",
        "multiValue": [
            {
                "value": "application/json"
            },
            {
                "value": "application/xml"
            },
            {
                "value": "text/html"
            }
        ]
    }
}
```

> **Note**
>
> Der erste Headerwert (in diesem Fall `application/json`) wird sowohl in der Eigenschaft `value` als auch in der Eigenschaft `multiValue` wiederholt. Auf diese Weise können Sie auf alle Werte zugreifen, indem Sie das `multiValue`-Array durchlaufen.

Wenn Ihr Funktionscode eine Abfragezeichenfolge, einen Header oder ein Cookie ändert, das ein `multiValue` Array enthält, verwendet CloudFront Functions die folgenden Regeln, um die Änderungen anzuwenden:

1. Wenn das `multiValue`-Array existiert und Änderungen hat, wird diese Änderung angewendet. Das erste Element in der `value`-Eigenschaft wird ignoriert.
2. Andernfalls wird jede Änderung der `value`-Eigenschaft angewendet, und nachfolgende Werte (falls vorhanden) bleiben unverändert.

Die `multiValue`-Eigenschaft wird nur verwendet, wenn die HTTP-Anfrage oder Antwort doppelte Abfragezeichenfolgen, Header oder Cookies mit demselben Namen enthält, wie im vorherigen Beispiel gezeigt. Wenn jedoch mehrere Werte in einer einzelnen Abfragezeichenfolge, einem Header oder einem Cookie vorhanden sind, wird die `multiValue`-Eigenschaft nicht verwendet.

Example Beispiel

Stellen Sie sich eine Anforderung mit einem `Accept`-Header vor, der drei Werte enthält.

```
Accept: application/json, application/xml, text/html
```

Dieser Header wird im `request`-Objekt wie folgt dargestellt.

```
"headers": {
    "accept": {
        "value": "application/json, application/xml, text/html"
    }
}
```

Cookie-Attribute

In einem `Set-Cookie`-Header in einer HTTP-Antwort enthält der Header das Name-Wert-Paar für das Cookie und optional eine Reihe von durch Semikolons getrennten Attributen.

Example Beispiel

```
Set-Cookie: cookie1=val1; Secure; Path=/; Domain=example.com; Expires=Wed, 05 Apr 2021
 07:28:00 GMT
```

In dem `response`-Objekt werden diese Attribute in der `attributes`-Eigenschaft des Cookie-Feldes dargestellt. Der vorangehende `Set-Cookie`-Header wird beispielsweise wie folgt dargestellt:

```
"cookie1": {
    "value": "val1",
    "attributes": "Secure; Path=/; Domain=example.com; Expires=Wed, 05 Apr 2021
 07:28:00 GMT"
}
```

Beispiel für Antwortobjekt

Das folgende Beispiel zeigt ein `response`-Objekt – die Ausgabe einer Viewer-Antwortfunktion –, in dem der Text durch eine Viewer-Antwortfunktion ersetzt wurde.

```
{
  "response": {
    "statusCode": 200,
    "statusDescription": "OK",
    "headers": {
      "date": {
        "value": "Mon, 04 Apr 2021 18:57:56 GMT"
      },
      "server": {
        "value": "gunicorn/19.9.0"
      },
      "access-control-allow-origin": {
        "value": "*"
      },
      "access-control-allow-credentials": {
        "value": "true"
      },
      "content-type": {
        "value": "text/html"
      },
      "content-length": {
        "value": "86"
      }
    },
    "cookies": {
      "ID": {
        "value": "id1234",
        "attributes": "Expires=Wed, 05 Apr 2021 07:28:00 GMT"
      },
```

```
      "Cookie1": {
        "value": "val1",
        "attributes": "Secure; Path=/; Domain=example.com; Expires=Wed, 05 Apr 2021
 07:28:00 GMT",
        "multiValue": [
          {
            "value": "val1",
            "attributes": "Secure; Path=/; Domain=example.com; Expires=Wed, 05 Apr 2021
 07:28:00 GMT"
          },
          {
            "value": "val2",
            "attributes": "Path=/cat; Domain=example.com; Expires=Wed, 10 Jan 2021
 07:28:00 GMT"
          }
        ]
      }
    },

    // Adding the body field is optional and it will not be present in the response
 object
    // unless you specify it in your function.
    // Your function does not have access to the original body returned by the
 CloudFront
    // cache or origin.
    // If you don't specify the body field in your viewer response function, the
 original
    // body returned by the CloudFront cache or origin is returned to viewer.

     "body": {
      "encoding": "text",
      "data": "<!DOCTYPE html><html><body><p>Here is your custom content.</p></body></
html>"
    }
  }
}
```

Beispiel für Ereignisobjekt

Das folgende Beispiel zeigt ein vollständiges `event`-Objekt: Dies ist ein Beispielaufruf für eine Standarddistribution und nicht für eine Multi-Tenant-Distribution. Bei Distributionen mit mehreren Mandanten wird das `endpoint` Feld anstelle von verwendet. `distributionDomainName` Der

Wert von `endpoint` ist der CloudFront Domänenname (z. B. d111111abcdef8.cloudfront.net) der Verbindungsgruppe, die mit dem Ereignis verknüpft ist.

> **Note**
>
> Das `event`-Objekt ist die Eingabe für Ihre Funktion. Ihre Funktion gibt nur das `request`- oder `response`-Objekt zurück, nicht das vollständige `event`-Objekt.

```
{
    "version": "1.0",
    "context": {
        "distributionDomainName": "d111111abcdef8.cloudfront.net",
        "distributionId": "EDFDVBD6EXAMPLE",
        "eventType": "viewer-response",
        "requestId": "EXAMPLEntjQpEXAMPLE_SG5Z-EXAMPLEPmPfEXAMPLEu3EqEXAMPLE=="
    },
    "viewer": {"ip": "198.51.100.11"},
    "request": {
        "method": "GET",
        "uri": "/media/index.mpd",
        "querystring": {
            "ID": {"value": "42"},
            "Exp": {"value": "1619740800"},
            "TTL": {"value": "1440"},
            "NoValue": {"value": ""},
            "querymv": {
                "value": "val1",
                "multiValue": [
                    {"value": "val1"},
                    {"value": "val2,val3"}
                ]
            }
        },
        "headers": {
            "host": {"value": "video.example.com"},
            "user-agent": {"value": "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:83.0)
 Gecko/20100101 Firefox/83.0"},
            "accept": {
                "value": "application/json",
                "multiValue": [
                    {"value": "application/json"},
```

```
                    {"value": "application/xml"},
                    {"value": "text/html"}
                ]
            },
            "accept-language": {"value": "en-GB,en;q=0.5"},
            "accept-encoding": {"value": "gzip, deflate, br"},
            "origin": {"value": "https://website.example.com"},
            "referer": {"value": "https://website.example.com/videos/12345678?
action=play"},
            "cloudfront-viewer-country": {"value": "GB"}
        },
        "cookies": {
            "Cookie1": {"value": "value1"},
            "Cookie2": {"value": "value2"},
            "cookie_consent": {"value": "true"},
            "cookiemv": {
                "value": "value3",
                "multiValue": [
                    {"value": "value3"},
                    {"value": "value4"}
                ]
            }
        }
    },
    "response": {
        "statusCode": 200,
        "statusDescription": "OK",
        "headers": {
            "date": {"value": "Mon, 04 Apr 2021 18:57:56 GMT"},
            "server": {"value": "gunicorn/19.9.0"},
            "access-control-allow-origin": {"value": "*"},
            "access-control-allow-credentials": {"value": "true"},
            "content-type": {"value": "application/json"},
            "content-length": {"value": "701"}
        },
        "cookies": {
            "ID": {
                "value": "id1234",
                "attributes": "Expires=Wed, 05 Apr 2021 07:28:00 GMT"
            },
            "Cookie1": {
                "value": "val1",
                "attributes": "Secure; Path=/; Domain=example.com; Expires=Wed, 05 Apr
 2021 07:28:00 GMT",
```

```
                "multiValue": [
                    {
                        "value": "val1",
                        "attributes": "Secure; Path=/; Domain=example.com; Expires=Wed,
 05 Apr 2021 07:28:00 GMT"
                    },
                    {
                        "value": "val2",
                        "attributes": "Path=/cat; Domain=example.com; Expires=Wed, 10
 Jan 2021 07:28:00 GMT"
                    }
                ]
            }
        }
    }
}
```

JavaScript Laufzeitfunktionen für CloudFront Funktionen

Die JavaScript Runtime-Umgebung von CloudFront Functions ist kompatibel mit ECMAScript (ES) Version 5.1 und unterstützt auch einige Funktionen der ES-Versionen 6 bis 12.

Für die meisten up-to-date Funktionen empfehlen wir die Verwendung von JavaScript Runtime 2.0.

Die JavaScript Runtime 2.0-Features weisen im Vergleich zu 1.0 die folgenden Änderungen auf:

- Puffermodul-Methoden sind verfügbar.
- Die folgenden nicht standardmäßigen Prototyp-Methoden für Zeichenfolgen sind nicht verfügbar:
  - `String.prototype.bytesFrom()`
  - `String.prototype.fromBytes()`
  - `String.prototype.fromUTF8()`
  - `String.prototype.toBytes()`
  - `String.prototype.toUTF8()`
- Das kryptografische Modul weist die folgenden Änderungen auf:
  - `hash.digest()` – Der Rückgabetyp wird in `Buffer` geändert, wenn keine Kodierung angegeben ist.
  - `hmac.digest()` – Der Rückgabetyp wird in `Buffer` geändert, wenn keine Kodierung angegeben ist.

- Weitere Informationen zu den zusätzlichen neuen Features finden Sie unter JavaScript Runtime 2.0-Funktionen für CloudFront Funktionen.

Themen

- JavaScript Runtime 1.0-Funktionen für CloudFront Funktionen
- JavaScript Runtime 2.0-Funktionen für CloudFront Funktionen

JavaScript Runtime 1.0-Funktionen für CloudFront Funktionen

Die JavaScript Runtime-Umgebung von CloudFront Functions ist kompatibel mit ECMAScript (ES) Version 5.1 und unterstützt auch einige Funktionen der ES-Versionen 6 bis 9. Sie enthält auch einige nicht standardmäßige Methoden, die nicht Teil der ES-Spezifikationen sind.

In den folgenden Themen werden alle unterstützten Sprachfunktionen aufgeführt.

Themen

- Kernfunktionen
- Primitive Objekte
- Integrierte Objekte
- Arten von Fehlern
- Globale
- Integrierten Module
- Eingeschränkte Funktionen

Kernfunktionen

Die folgenden Kern-Features von ES werden unterstützt.

Typen

Alle ES 5.1-Typen werden unterstützt. Dies umfasst boolesche Werte, Zahlen, Zeichenfolgen, Objekte, Arrays, Funktionen, Funktionskonstruktoren und reguläre Ausdrücke.

Operatoren

Alle ES 5.1-Operatoren werden unterstützt.

Der Potenzierungsoperator (**) wird unterstützt.

Anweisungen

> **Note**
>
> `const`- und `let`-Anweisungen werden nicht unterstützt.

Die folgenden ES 5.1-Anweisungen werden unterstützt:

- `break`
- `catch`
- `continue`
- `do-while`
- `else`
- `finally`
- `for`
- `for-in`
- `if`
- `return`
- `switch`
- `throw`
- `try`
- `var`
- `while`
- Bezeichnete Anweisungen

Literale

ES 6-Vorlagenliterale werden unterstützt: mehrzeilige Zeichenfolgen, Interpolation von Ausdrücken und Verschachtelungsvorlagen.

Funktionen

Alle Features von ES 5.1 werden unterstützt.

ES 6-Pfeilfunktionen werden unterstützt, und die ES 6 Rest-Parametersyntax wird unterstützt.

Unicode

Quelltext und Zeichenfolgenliterale können Unicode-codierte Zeichen enthalten. Unicode-Code-Punkt-Escape-Sequenzen von sechs Zeichen (z. B. \uXXXX) werden ebenfalls unterstützt.

Strikter Modus

Funktionen arbeiten standardmäßig im strikten Modus, sodass Sie Ihrem Funktionscode keine `use strict`-Anweisung hinzufügen müssen. Dies können nicht geändert werden.

Primitive Objekte

Die folgenden primitiven Objekte von ES werden unterstützt.

Objekt

Die folgenden ES 5.1-Methoden für Objekte werden unterstützt:

- `create` (ohne Eigenschaftenliste)
- `defineProperties`
- `defineProperty`
- `freeze`
- `getOwnPropertyDescriptor`
- `getOwnPropertyNames`
- `getPrototypeOf`
- `hasOwnProperty`
- `isExtensible`
- `isFrozen`
- `prototype.isPrototypeOf`
- `isSealed`
- `keys`
- `preventExtensions`
- `prototype.propertyIsEnumerable`
- `seal`
- `prototype.toString`

- `prototype.valueOf`

Die folgenden ES 6-Methoden für Objekte werden unterstützt:

- `assign`
- `is`
- `prototype.setPrototypeOf`

Die folgenden ES 8-Methoden für Objekte werden unterstützt:

- `entries`
- `values`

Zeichenfolge

Die folgenden ES 5.1-Methoden für Zeichenfolgen werden unterstützt:

- `fromCharCode`
- `prototype.charAt`
- `prototype.concat`
- `prototype.indexOf`
- `prototype.lastIndexOf`
- `prototype.match`
- `prototype.replace`
- `prototype.search`
- `prototype.slice`
- `prototype.split`
- `prototype.substr`
- `prototype.substring`
- `prototype.toLowerCase`
- `prototype.trim`
- `prototype.toUpperCase`

Die folgenden ES 6-Methoden für Zeichenfolgen werden unterstützt:

- `fromCodePoint`
- `prototype.codePointAt`

- `prototype.endsWith`
- `prototype.includes`
- `prototype.repeat`
- `prototype.startsWith`

Die folgenden ES 8-Methoden für Zeichenfolgen werden unterstützt:

- `prototype.padStart`
- `prototype.padEnd`

Die folgenden ES 9-Methoden für Zeichenfolgen werden unterstützt:

- `prototype.trimStart`
- `prototype.trimEnd`

Die folgenden nicht standardmäßigen Methoden für Zeichenfolgen werden unterstützt:

- `prototype.bytesFrom(`*`array | string, encoding`*`)`

  Erstellt eine Bytezeichenfolge aus einem Array von Oktetten oder einer codierten Zeichenfolge. Die Optionen für die Zeichenfolgencodierung sind `hex`, `base64` und `base64url`.

- `prototype.fromBytes(`*`start`*`[, `*`end`*`])`

  Erstellt eine Unicode-Zeichenfolge aus einer Bytezeichenfolge, in der jedes Byte durch den entsprechenden Unicode-Codepunkt ersetzt wird.

- `prototype.fromUTF8(`*`start`*`[, `*`end`*`])`

  Erstellt eine Unicode-Zeichenfolge aus einer UTF-8-codierten Bytezeichenfolge. Wenn die Codierung falsch ist, wird `null` zurückgegeben.

- `prototype.toBytes(start[, end])`

  Erstellt eine Bytezeichenfolge aus einer Unicode-Zeichenfolge. Alle Zeichen müssen im Bereich von [0,255] liegen. Wenn nicht, wird `null` zurückgegeben.

- `prototype.toUTF8(start[, `*`end`*`])`

  Erstellt eine UTF-8-codierte Bytezeichenfolge aus einer Unicode-Zeichenfolge.

Nummer

Alle ES 5.1-Methoden für Zahlen werden unterstützt.

Die folgenden ES 6-Methoden für Zahlen werden unterstützt:

- `isFinite`
- `isInteger`
- `isNaN`
- `isSafeInteger`
- `parseFloat`
- `parseInt`
- `prototype.toExponential`
- `prototype.toFixed`
- `prototype.toPrecision`
- `EPSILON`
- `MAX_SAFE_INTEGER`
- `MAX_VALUE`
- `MIN_SAFE_INTEGER`
- `MIN_VALUE`
- `NEGATIVE_INFINITY`
- `NaN`
- `POSITIVE_INFINITY`

Integrierte Objekte

Die folgenden integrierten Objekte von ES werden unterstützt.

Math-Knoten

Alle Mathematikmethoden von ES 5.1 werden unterstützt.

> **Note**
>
> In der Runtime-Umgebung von CloudFront Functions verwendet die `Math.random()` Implementierung OpenBSD`arc4random`, das mit dem Zeitstempel versehen ist, wann die Funktion ausgeführt wird.

Die folgenden ES 6-Mathematikmethoden werden unterstützt:

- `acosh`
- `asinh`
- `atanh`
- `cbrt`
- `clz32`
- `cosh`
- `expm1`
- `fround`
- `hypot`
- `imul`
- `log10`
- `log1p`
- `log2`
- `sign`
- `sinh`
- `tanh`
- `trunc`
- `E`
- `LN10`
- `LN2`
- `LOG10E`
- `LOG2E`
- `PI`
- `SQRT1_2`
- `SQRT2`

Datum

Alle `Date`-Features von ES 5.1 werden unterstützt.

> **Note**
>
> Aus Sicherheitsgründen gibt `Date` immer den gleichen Wert – die Startzeit der Funktion – während der Lebensdauer einer einzelnen Funktionsausführung zurück. Weitere Informationen finden Sie unter Eingeschränkte Funktionen.

Funktion

Die Methoden `apply`, `bind` und `call` werden unterstützt.

Funktionskonstruktoren werden nicht unterstützt.

Reguläre Ausdrücke

Alle Features für reguläre Ausdrücke von ES 5.1 werden unterstützt. Die Sprache für reguläre Ausdrücke ist Perl-kompatibel. ES 9 benannte Aufnahmegruppen werden unterstützt.

JSON

Alle Funktionen von ES 5.1 JSON werden unterstützt, einschließlich `parse` und `stringify`.

Array

Die folgenden ES 5.1-Methoden für Arrays werden unterstützt:

- `isArray`
- `prototype.concat`
- `prototype.every`
- `prototype.filter`
- `prototype.forEach`
- `prototype.indexOf`
- `prototype.join`
- `prototype.lastIndexOf`
- `prototype.map`
- `prototype.pop`
- `prototype.push`
- `prototype.reduce`

- `prototype.reduceRight`
- `prototype.reverse`
- `prototype.shift`
- `prototype.slice`
- `prototype.some`
- `prototype.sort`
- `prototype.splice`
- `prototype.unshift`

Die folgenden ES 6-Methoden für Arrays werden unterstützt:

- `of`
- `prototype.copyWithin`
- `prototype.fill`
- `prototype.find`
- `prototype.findIndex`

Die folgenden ES 7-Methoden für Arrays werden unterstützt:

- `prototype.includes`

Eingegebene Arrays

Die folgenden von ES 6 eingegebenen Arrays werden unterstützt:

- `Int8Array`
- `Uint8Array`
- `Uint8ClampedArray`
- `Int16Array`
- `Uint16Array`
- `Int32Array`
- `Uint32Array`
- `Float32Array`
- `Float64Array`
- `prototype.copyWithin`

- `prototype.fill`
- `prototype.join`
- `prototype.set`
- `prototype.slice`
- `prototype.subarray`
- `prototype.toString`

ArrayBuffer

Die folgenden Methoden für `ArrayBuffer` werden unterstützt:

- `prototype.isView`
- `prototype.slice`

Promise

Die folgenden Methoden für Versprechen werden unterstützt:

- `reject`
- `resolve`
- `prototype.catch`
- `prototype.finally`
- `prototype.then`

Crypto

Das kryptografische Modul bietet standardmäßige Hashing- und Hash-basierte HMAC-Helfer (Message Authentication Code). Sie können das Modul mit `require('crypto')` laden. Das Modul stellt die folgenden Methoden bereit, die sich genau wie ihre Gegenstücke von Node.js verhalten:

- `createHash(`*`algorithm`*`)`
- `hash.update(`*`data`*`)`
- `hash.digest([`*`encoding`*`])`
- `createHmac(`*`algorithm, secret key`*`)`
- `hmac.update(`*`data`*`)`
- `hmac.digest([`*`encoding`*`])`

Weitere Informationen finden Sie unter Krypto (Hash und HMAC) im Abschnitt über integrierte Module.

Konsole

Dies ist ein Hilfsobjekt zum Debuggen. Es unterstützt nur die `log()`-Methode, um Protokollnachrichten aufzuzeichnen.

> **Note**
>
> CloudFront Functions unterstützt keine Kommasyntax, wie z. `console.log('a', 'b')` Verwenden Sie stattdessen das Format `console.log('a' + ' ' + 'b')`.

Arten von Fehlern

Die folgenden Fehlerobjekte werden unterstützt:

- `Error`
- `EvalError`
- `InternalError`
- `MemoryError`
- `RangeError`
- `ReferenceError`
- `SyntaxError`
- `TypeError`
- `URIError`

Globale

Das `globalThis`-Objekt wird unterstützt.

Die folgenden globalen Funktionen von ES 5.1 werden unterstützt:

- `decodeURI`
- `decodeURIComponent`
- `encodeURI`

- `encodeURIComponent`
- `isFinite`
- `isNaN`
- `parseFloat`
- `parseInt`

Die folgenden globalen Konstanten werden unterstützt:

- `NaN`
- `Infinity`
- `undefined`

Integrierten Module

Die folgenden integrierten Module werden unterstützt.

Module

- Krypto (Hash und HMAC)
- Abfragezeichenfolge

Krypto (Hash und HMAC)

Das kryptographische Modul (`crypto`) bietet standardmäßige Hashing- und HMAC-Helfer (Hash-basierter Nachrichtenauthentifizierungscode). Sie können das Modul mit `require('crypto')` laden. Das Modul bietet die folgenden Methoden, die sich genau wie ihre Gegenstücke von Node.js verhalten.

Hashing-Methoden

`crypto.createHash(`*`algorithm`*`)`

Erstellt und gibt ein Hash-Objekt zurück, mit dem Sie Hash-Digests mit dem angegebenen Algorithmus generieren können: `md5`, `sha1` oder `sha256`.

`hash.update(`*`data`*`)`

Aktualisiert den Hash-Inhalt mit dem angegebenen `data`.

`hash.digest([encoding])`

Berechnet den Digest aller mit `hash.update()` übergebenen Daten. Die Codierung kann `hex`, `base64` oder `base64url` sein.

HMAC-Methoden

`crypto.createHmac(algorithm, secret key)`

Erstellt und gibt ein HMAC-Objekt zurück, das das angegebene `algorithm` und `secret key` verwendet. Der Algorithmus kann `md5`, `sha1` oder `sha256` sein.

`hmac.update(data)`

Aktualisiert den HMAC-Inhalt mit den angegebenen `data`.

`hmac.digest([encoding])`

Berechnet den Digest aller mit `hmac.update()` übergebenen Daten. Die Codierung kann `hex`, `base64` oder `base64url` sein.

Abfragezeichenfolge

> **Note**
>
> Das CloudFront Functions-Ereignisobjekt analysiert automatisch URL-Abfragezeichenfolgen für Sie. Das bedeutet, dass Sie dieses Modul in den meisten Fällen nicht verwenden müssen.

Das Modul für Abfragezeichenfolgen (`querystring`) bietet Methoden zum Analysieren und Formatieren von URL-Abfragezeichenfolgen. Sie können das Modul mit `require('querystring')` laden. Das Modul bietet die folgenden Methoden.

`querystring.escape(string)`

URL-kodiert die angegebene `string` und gibt eine entflohene Abfragezeichenfolge zurück. Die Methode wird von `querystring.stringify()` verwendet und sollte nicht direkt verwendet werden.

`querystring.parse(string[, separator[, equal[, options]]])`

Analysiert eine Abfragezeichenfolge (`string`) und gibt ein Objekt zurück.

Der `separator`-Parameter ist eine Teilzeichenfolge zum Abgrenzen von Schlüssel- und Wertepaaren in der Abfragezeichenfolge. Standardmäßig ist dies &.

Der `equal`-Parameter ist eine Teilzeichenfolge zum Abgrenzen von Schlüsseln und Werten in der Abfragezeichenfolge. Standardmäßig ist dies =.

Der `options`-Parameter ist ein Objekt mit den folgenden Schlüsseln:

`decodeURIComponent` *`function`*

Eine Funktion zum Entschlüsseln von prozentkodierten Zeichen in der Abfragezeichenfolge. Standardmäßig ist dies `querystring.unescape()`.

`maxKeys` *`number`*

Die maximale Anzahl der Schlüssel zum Parsen. Standardmäßig ist dies `1000`. Verwenden Sie den Wert `0`, um die Beschränkungen für das Zählen von Schlüsseln aufzuheben.

Standardmäßig wird davon ausgegangen, dass prozentcodierte Zeichen innerhalb der Abfragezeichenfolge die UTF-8-Codierung verwenden. Ungültige UTF-8-Sequenzen werden durch das Ersatzzeichen `U+FFFD` ersetzt.

Zum Beispiel für die folgende Abfragezeichenfolge:

```
'name=value&abc=xyz&abc=123'
```

Der Rückgabewert von `querystring.parse()` ist:

```
{
name: 'value',
abc: ['xyz', '123']
}
```

`querystring.decode()` ist ein Alias für `querystring.parse()`.

`querystring.stringify(`*`object`*`[, `*`separator`*`[, `*`equal`*`[, `*`options`*`]]])`

Serialisiert ein `object` und gibt eine Abfragezeichenfolge zurück.

Der `separator`-Parameter ist eine Teilzeichenfolge zum Abgrenzen von Schlüssel- und Wertepaaren in der Abfragezeichenfolge. Standardmäßig ist dies &.

Der `equal`-Parameter ist eine Teilzeichenfolge zum Abgrenzen von Schlüsseln und Werten in der Abfragezeichenfolge. Standardmäßig ist dies =.

Der `options`-Parameter ist ein Objekt mit den folgenden Schlüsseln:

`encodeURIComponent` *`function`*

Die Funktion, die zum Konvertieren von URL-unsicheren Zeichen in die prozentuale Kodierung in der Abfragezeichenfolge verwendet wird. Standardmäßig ist dies `querystring.escape()`.

Standardmäßig werden Zeichen, die eine prozentuale Kodierung innerhalb der Abfragezeichenfolge erfordern, als UTF-8 codiert. Um eine andere Codierung zu verwenden, geben Sie die Option `encodeURIComponent` an.

Zum Beispiel für den folgenden Code:

```
querystring.stringify({ name: 'value', abc: ['xyz', '123'], anotherName: '' });
```

Der Rückgabewert ist:

```
'name=value&abc=xyz&abc=123&anotherName='
```

`querystring.encode()` ist ein Alias für `querystring.stringify()`.

`querystring.unescape(`*`string`*`)`

Dekodiert die prozentualen Zeichen der URL in der angegebenen `string` und gibt eine nicht entdeckene Abfragezeichenfolge zurück. Diese Methode wird von `querystring.parse()` verwendet und sollte nicht direkt verwendet werden.

Eingeschränkte Funktionen

Die folgenden JavaScript Sprachfunktionen werden entweder nicht unterstützt oder sind aus Sicherheitsgründen eingeschränkt.

Dynamische Codeauswertung

Die dynamische Codeauswertung wird nicht unterstützt. Sowohl `eval()`- als auch `Function`-Konstruktoren geben einen Fehler aus, wenn sie versucht werden. Zum Beispiel gibt `const sum = new Function('a', 'b', 'return a + b')` einen Fehler aus.

Timer

Die Funktionen `setTimeout()`, `setImmediate()` und `clearTimeout()` werden nicht unterstützt. Es gibt keine Bestimmung, innerhalb einer Funktionsausführung zu verschieben oder zu ergeben. Ihre Funktion muss synchron bis zum Abschluss ausgeführt werden.

Datum und Zeitstempel

Aus Sicherheitsgründen besteht kein Zugang zu hochauflösenden Timern. Alle `Date`-Methoden zum Abfragen der aktuellen Uhrzeit geben während der Lebensdauer einer einzelnen Funktionsausführung immer den gleichen Wert zurück. Der zurückgegebene Zeitstempel ist die Zeit, zu der die Funktion gestartet wurde. Folglich können Sie die verstrichene Zeit in Ihrer Funktion nicht messen.

Zugriff auf das Dateisystem

Es gibt keinen Zugriff auf das Dateisystem. Zum Beispiel gibt es kein `fs`-Modul für den Dateisystemzugriff wie in Node.js.

Prozesszugriff

Es gibt keinen Prozesszugriff. Zum Beispiel gibt es kein globales `process`-Objekt für die Verarbeitung von Informationszugriffen wie in Node.js.

Umgebungsvariablen

Es gibt keinen Zugriff auf Umgebungsvariablen.

Stattdessen können Sie verwenden, CloudFront KeyValueStore um einen zentralen Datenspeicher mit Schlüssel-Wert-Paaren für Ihre Funktionen zu erstellen. CloudFront CloudFront KeyValueStore ermöglicht dynamische Aktualisierungen Ihrer Konfigurationsdaten, ohne dass Codeänderungen vorgenommen werden müssen. Sie müssen JavaScript Runtime 2.0 verwenden, um es zu verwenden CloudFront KeyValueStore. Weitere Informationen finden Sie unter Amazon CloudFront KeyValueStore.

Netzwerkzugriff

Es gibt keine Unterstützung für Netzwerkaufrufe. Zum Beispiel werden XHR, HTTP(S) und Socket nicht unterstützt.

JavaScript Runtime 2.0-Funktionen für CloudFront Funktionen

Die JavaScript Runtime-Umgebung von CloudFront Functions ist kompatibel mit ECMAScript (ES) Version 5.1 und unterstützt auch einige Funktionen der ES-Versionen 6 bis 12. Sie enthält auch

einige nicht standardmäßige Methoden, die nicht Teil der ES-Spezifikationen sind. In den folgenden Themen werden alle unterstützten Features dieser Laufzeit aufgeführt.

Themen

- Kern-Features
- Primitive Objekte
- Integrierte Objekte
- Fehlertypen
- Globale
- Integrierten Module
- Eingeschränkte Features

Kern-Features

Die folgenden Kern-Features von ES werden unterstützt.

Typen

Alle ES 5.1-Typen werden unterstützt. Dies umfasst boolesche Werte, Zahlen, Zeichenfolgen, Objekte, Arrays, Funktionen und reguläre Ausdrücke.

Betreiber

Alle ES 5.1-Operatoren werden unterstützt.

Der Potenzierungsoperator (`**`) wird unterstützt.

Anweisungen

Die folgenden ES 5.1-Anweisungen werden unterstützt:

- `break`
- `catch`
- `continue`
- `do-while`
- `else`
- `finally`

- `for`
- `for-in`
- `if`
- `label`
- `return`
- `switch`
- `throw`
- `try`
- `var`
- `while`

Die folgenden ES 6-Anweisungen werden unterstützt:

- `const`
- `let`

Die folgenden ES 8-Anweisungen werden unterstützt:

- `async`
- `await`

> **Note**
>
> `async`, `awaitconst`, und `let` werden in JavaScript Runtime 2.0 unterstützt. `await` kann nur innerhalb von `async`-Funktionen verwendet werden. `async`-Argumente und -Schließungen werden nicht unterstützt.

Literale

ES 6-Vorlagenliterale werden unterstützt: mehrzeilige Zeichenfolgen, Interpolation von Ausdrücken und Verschachtelungsvorlagen.

Funktionen

Alle Features von ES 5.1 werden unterstützt.

ES 6-Pfeilfunktionen werden unterstützt, und die ES 6 Rest-Parametersyntax wird unterstützt.

Unicode

Quelltext und Zeichenfolgenliterale können Unicode-codierte Zeichen enthalten. Unicode-Code-Punkt-Escape-Sequenzen von sechs Zeichen (z. B. \uXXXX) werden ebenfalls unterstützt.

Strikter Modus

Funktionen arbeiten standardmäßig im strikten Modus, sodass Sie Ihrem Funktionscode keine `use strict`-Anweisung hinzufügen müssen. Dies können nicht geändert werden.

Primitive Objekte

Die folgenden primitiven Objekte von ES werden unterstützt.

Objekt

Die folgenden ES 5.1-Methoden für Objekte werden unterstützt:

- `Object.create()` (ohne Eigenschaftenliste)
- `Object.defineProperties()`
- `Object.defineProperty()`
- `Object.freeze()`
- `Object.getOwnPropertyDescriptor()`
- `Object.getOwnPropertyDescriptors()`
- `Object.getOwnPropertyNames()`
- `Object.getPrototypeOf()`
- `Object.isExtensible()`
- `Object.isFrozen()`
- `Object.isSealed()`
- `Object.keys()`
- `Object.preventExtensions()`
- `Object.seal()`

Die folgenden ES 6-Methoden für Objekte werden unterstützt:

- `Object.assign()`

Die folgenden ES 8-Methoden für Objekte werden unterstützt:

- `Object.entries()`
- `Object.values()`

Die folgenden ES-5.1-Prototyp-Methoden für Objekte werden unterstützt:

- `Object.prototype.hasOwnProperty()`
- `Object.prototype.isPrototypeOf()`
- `Object.prototype.propertyIsEnumerable()`
- `Object.prototype.toString()`
- `Object.prototype.valueOf()`

Die folgenden ES 6-Prototyp-Methoden für Objekte werden unterstützt:

- `Object.prototype.is()`
- `Object.prototype.setPrototypeOf()`

Zeichenfolge

Die folgenden ES 5.1-Methoden für Zeichenfolgen werden unterstützt:

- `String.fromCharCode()`

Die folgenden ES 6-Methoden für Zeichenfolgen werden unterstützt:

- `String.fromCodePoint()`

Die folgenden ES-5.1-Prototyp-Methoden für Zeichenfolgen werden unterstützt:

- `String.prototype.charAt()`
- `String.prototype.concat()`
- `String.prototype.indexOf()`
- `String.prototype.lastIndexOf()`
- `String.prototype.match()`
- `String.prototype.replace()`
- `String.prototype.search()`
- `String.prototype.slice()`
- `String.prototype.split()`
- `String.prototype.substr()`
- `String.prototype.substring()`
- `String.prototype.toLowerCase()`

- `String.prototype.trim()`
- `String.prototype.toUpperCase()`

Die folgenden ES 6-Prototyp-Methoden für Zeichenfolgen werden unterstützt:

- `String.prototype.codePointAt()`
- `String.prototype.endsWith()`
- `String.prototype.includes()`
- `String.prototype.repeat()`
- `String.prototype.startsWith()`

Die folgenden ES 8-Prototyp-Methoden für Zeichenfolgen werden unterstützt:

- `String.prototype.padStart()`
- `String.prototype.padEnd()`

Die folgenden ES 9-Prototyp-Methoden für Zeichenfolgen werden unterstützt:

- `String.prototype.trimStart()`
- `String.prototype.trimEnd()`

Die folgenden ES 12-Prototyp-Methoden für Zeichenfolgen werden unterstützt:

- `String.prototype.replaceAll()`

> **Note**
>
> `String.prototype.replaceAll()`ist neu in JavaScript Runtime 2.0.

Zahl

ALLE ES 5-Zahlen werden unterstützt.

Die folgenden ES 6-Eigenschaften für Zahlen werden unterstützt:

- `Number.EPSILON`
- `Number.MAX_SAFE_INTEGER`
- `Number.MIN_SAFE_INTEGER`
- `Number.MAX_VALUE`
- `Number.MIN_VALUE`
- `Number.NaN`

- `Number.NEGATIVE_INFINITY`
- `Number.POSITIVE_INFINITY`

Die folgenden ES 6-Methoden für Zahlen werden unterstützt:

- `Number.isFinite()`
- `Number.isInteger()`
- `Number.isNaN()`
- `Number.isSafeInteger()`
- `Number.parseInt()`
- `Number.parseFloat()`

Die folgenden ES 5.1-Prototyp-Methoden für Zahlen werden unterstützt:

- `Number.prototype.toExponential()`
- `Number.prototype.toFixed()`
- `Number.prototype.toPrecision()`

Numerische ES 12-Trennzeichen werden unterstützt.

> **Note**
>
> Numerische ES12-Trennzeichen sind neu in JavaScript Runtime 2.0.

Integrierte Objekte

Die folgenden integrierten Objekte von ES werden unterstützt.

Math-Knoten

Alle Mathematikmethoden von ES 5.1 werden unterstützt.

> **Note**
>
> In der Runtime-Umgebung von CloudFront Functions verwendet die `Math.random()` Implementierung OpenBSD`arc4random`, das mit dem Zeitstempel versehen ist, wann die Funktion ausgeführt wird.

Die folgenden ES 6-Mathematikeigenschaften werden unterstützt:

- `Math.E`
- `Math.LN10`
- `Math.LN2`
- `Math.LOG10E`
- `Math.LOG2E`
- `Math.PI`
- `Math.SQRT1_2`
- `Math.SQRT2`

Die folgenden ES 6-Mathematikmethoden werden unterstützt:

- `Math.abs()`
- `Math.acos()`
- `Math.acosh()`
- `Math.asin()`
- `Math.asinh()`
- `Math.atan()`
- `Math.atan2()`
- `Math.atanh()`
- `Math.cbrt()`
- `Math.ceil()`
- `Math.clz32()`
- `Math.cos()`
- `Math.cosh()`
- `Math.exp()`
- `Math.expm1()`
- `Math.floor()`
- `Math.fround()`
- `Math.hypot()`
- `Math.imul()`

- `Math.log()`
- `Math.log1p()`
- `Math.log2()`
- `Math.log10()`
- `Math.max()`
- `Math.min()`
- `Math.pow()`
- `Math.random()`
- `Math.round()`
- `Math.sign()`
- `Math.sinh()`
- `Math.sin()`
- `Math.sqrt()`
- `Math.tan()`
- `Math.tanh()`
- `Math.trunc()`

Datum

Alle `Date`-Features von ES 5.1 werden unterstützt.

> **Note**
>
> Aus Sicherheitsgründen gibt `Date` immer den gleichen Wert – die Startzeit der Funktion – während der Lebensdauer einer einzelnen Funktionsausführung zurück. Weitere Informationen finden Sie unter Eingeschränkte Funktionen.

Funktion

Die folgenden ES-5.1-Prototyp-Methoden werden unterstützt:

- `Function.prototype.apply()`
- `Function.prototype.bind()`
- `Function.prototype.call()`

Funktionskonstruktoren werden nicht unterstützt.

Reguläre Ausdrücke

Alle Features für reguläre Ausdrücke von ES 5.1 werden unterstützt. Die Sprache für reguläre Ausdrücke ist Perl-kompatibel.

Die folgenden ES-5.1-Prototyp-Zugriffseigenschaften werden unterstützt:

- `RegExp.prototype.global`
- `RegExp.prototype.ignoreCase`
- `RegExp.protoype.multiline`
- `RegExp.protoype.source`
- `RegExp.prototype.sticky`
- `RegExp.prototype.flags`

> **Note**
>
> `RegExp.prototype.sticky`und `RegExp.prototype.flags` sind neu in JavaScript Runtime 2.0.

Die folgenden ES-5.1-Prototyp-Methoden werden unterstützt:

- `RegExp.prototype.exec()`
- `RegExp.prototype.test()`
- `RegExp.prototype.toString()`
- `RegExp.prototype[@@replace]()`
- `RegExp.prototype[@@split]()`

> **Note**
>
> `RegExp.prototype[@@split]()`ist neu in JavaScript Runtime 2.0.

Die folgenden ES-5.1-Instance-Eigenschaften werden unterstützt:

- `lastIndex`

ES 9 benannte Aufnahmegruppen werden unterstützt.

JSON

Die folgenden ES 5.1-Mathematikmethoden werden unterstützt:

- `JSON.parse()`
- `JSON.stringify()`

Array

Die folgenden ES 5.1-Methoden für Arrays werden unterstützt:

- `Array.isArray()`

Die folgenden ES 6-Methoden für Arrays werden unterstützt:

- `Array.of()`

Die folgenden ES-5.1-Prototyp-Methoden werden unterstützt:

- `Array.prototype.concat()`
- `Array.prototype.every()`
- `Array.prototype.filter()`
- `Array.prototype.forEach()`
- `Array.prototype.indexOf()`
- `Array.prototype.join()`
- `Array.prototype.lastIndexOf()`
- `Array.prototype.map()`
- `Array.prototype.pop()`
- `Array.prototype.push()`
- `Array.prototype.reduce()`
- `Array.prototype.reduceRight()`
- `Array.prototype.reverse()`
- `Array.prototype.shift()`
- `Array.prototype.slice()`
- `Array.prototype.some()`
- `Array.prototype.sort()`
- `Array.prototype.splice()`

- `Array.prototype.unshift()`

Die folgenden ES 6-Prototyp-Methoden werden unterstützt:

- `Array.prototype.copyWithin()`
- `Array.prototype.fill()`
- `Array.prototype.find()`
- `Array.prototype.findIndex()`

Die folgenden ES 7-Prototyp-Methoden werden unterstützt:

- `Array.prototype.includes()`

Eingegebene Arrays

Die folgenden von ES 6 eingegebenen Array-Konstruktoren werden unterstützt:

- `Float32Array`
- `Float64Array`
- `Int8Array`
- `Int16Array`
- `Int32Array`
- `Uint8Array`
- `Uint8ClampedArray`
- `Uint16Array`
- `Uint32Array`

Die folgenden ES-6-Methoden werden unterstützt:

- `TypedArray.from()`
- `TypedArray.of()`

> **Note**
>
> `TypedArray.from()`und `TypedArray.of()` sind neu in JavaScript Runtime 2.0.

Die folgenden ES 6-Prototyp-Methoden werden unterstützt:

- `TypedArray.prototype.copyWithin()`
- `TypedArray.prototype.every()`

- `TypedArray.prototype.fill()`
- `TypedArray.prototype.filter()`
- `TypedArray.prototype.find()`
- `TypedArray.prototype.findIndex()`
- `TypedArray.prototype.forEach()`
- `TypedArray.prototype.includes()`
- `TypedArray.prototype.indexOf()`
- `TypedArray.prototype.join()`
- `TypedArray.prototype.lastIndexOf()`
- `TypedArray.prototype.map()`
- `TypedArray.prototype.reduce()`
- `TypedArray.prototype.reduceRight()`
- `TypedArray.prototype.reverse()`
- `TypedArray.prototype.some()`
- `TypedArray.prototype.set()`
- `TypedArray.prototype.slice()`
- `TypedArray.prototype.sort()`
- `TypedArray.prototype.subarray()`
- `TypedArray.prototype.toString()`

> **Note**
>
> `TypedArray.prototype.every()`,`TypedArray.prototype.fill()`,`TypedArray.proto`
> `TypedArray.prototype.reduceRight()``TypedArray.prototype.reverse()`,
> und `TypedArray.prototype.some()` sind neu in JavaScript Runtime 2.0.

ArrayBuffer

Die folgenden ES 6-Methoden ArrayBuffer werden unterstützt:

- `isView()`

Die folgenden ES 6-Prototypmethoden ArrayBuffer werden unterstützt:

- `ArrayBuffer.prototype.slice()`

Promise

Die folgenden ES 6-Methoden für Versprechen werden unterstützt:

- `Promise.all()`
- `Promise.allSettled()`
- `Promise.any()`
- `Promise.reject()`
- `Promise.resolve()`
- `Promise.race()`

> **Note**
>
> `Promise.all()`, `Promise.allSettled()``Promise.any()`, und `Promise.race()` sind neu in JavaScript Runtime 2.0.

Die folgenden ES 6-Prototyp-Methoden für Versprechen werden unterstützt:

- `Promise.prototype.catch()`
- `Promise.prototype.finally()`
- `Promise.prototype.then()`

DataView

Die folgenden ES 6-Prototyp-Methoden werden unterstützt:

- `DataView.prototype.getFloat32()`
- `DataView.prototype.getFloat64()`
- `DataView.prototype.getInt16()`
- `DataView.prototype.getInt32()`
- `DataView.prototype.getInt8()`
- `DataView.prototype.getUint16()`
- `DataView.prototype.getUint32()`
- `DataView.prototype.getUint8()`
- `DataView.prototype.setFloat32()`
- `DataView.prototype.setFloat64()`

- `DataView.prototype.setInt16()`
- `DataView.prototype.setInt32()`
- `DataView.prototype.setInt8()`
- `DataView.prototype.setUint16()`
- `DataView.prototype.setUint32()`
- `DataView.prototype.setUint8()`

> **Note**
>
> Alle Prototypmethoden von Dataview ES 6 sind neu in JavaScript Runtime 2.0.

Symbol

Die folgenden ES-6-Methoden werden unterstützt:

- `Symbol.for()`
- `Symbol.keyfor()`

> **Note**
>
> Alle Symbol ES 6-Methoden sind neu in JavaScript Runtime 2.0.

Textdecoder

Die folgenden Prototyp-Methoden werden unterstützt:

- `TextDecoder.prototype.decode()`

Die folgenden Prototyp-Zugriffseigenschaften werden unterstützt:

- `TextDecoder.prototype.encoding`
- `TextDecoder.prototype.fatal`
- `TextDecoder.prototype.ignoreBOM`

Text-Encoder

Die folgenden Prototyp-Methoden werden unterstützt:

- `TextEncoder.prototype.encode()`
- `TextEncoder.prototype.encodeInto()`

Konsole

Dies ist ein Hilfsobjekt zum Debuggen. Es unterstützt nur die `log()`-Methode, um Protokollnachrichten aufzuzeichnen.

> **Note**
>
> CloudFront Functions unterstützt keine Kommasyntax, wie z. B. `console.log('a', 'b')` Verwenden Sie stattdessen das Format `console.log('a' + ' ' + 'b')`.

Fehlertypen

Die folgenden Fehlerobjekte werden unterstützt:

- `Error`
- `EvalError`
- `InternalError`
- `RangeError`
- `ReferenceError`
- `SyntaxError`
- `TypeError`
- `URIError`

Globale

Das `globalThis`-Objekt wird unterstützt.

Die folgenden globalen Funktionen von ES 5.1 werden unterstützt:

- `decodeURI()`
- `decodeURIComponent()`
- `encodeURI()`
- `encodeURIComponent()`
- `isFinite()`
- `isNaN()`

- `parseFloat()`
- `parseInt()`

Die folgenden globalen Funktionen von ES 6 werden unterstützt:

- `atob()`
- `btoa()`

> **Note**
>
> `atob()`und `btoa()` sind neu in JavaScript Runtime 2.0.

Die folgenden globalen Konstanten werden unterstützt:

- `NaN`
- `Infinity`
- `undefined`
- `arguments`

Integrierten Module

Die folgenden integrierten Module werden unterstützt.

Module

- Buffer
- Abfragezeichenfolge
- Crypto

Buffer

Das Modul bietet die folgenden Methoden:

- `Buffer.alloc(size[, fill[, encoding]])`

  Weisen Sie einen `Buffer` zu.

  - `size`: Puffergröße. Geben Sie eine Ganzzahl ein.
  - `fill`: Optional. Geben Sie eine Zeichenfolge, `Buffer`, Uint8Array oder eine Ganzzahl ein. Der Standardwert ist `0`.
  - `encoding`: Optional. Wenn `fill` eine Zeichenfolge ist, geben Sie eine der folgenden Optionen ein: `utf8`, `hex`, `base64`, `base64url`. Der Standardwert ist `utf8`.
- `Buffer.allocUnsafe(size)`

  Weisen Sie einen nicht initialisierten `Buffer` zu.
  - `size`: Geben Sie eine Ganzzahl ein.
- `Buffer.byteLength(value[, encoding])`

  Gibt die Länge eines Werts in Byte zurück.
  - `value`: Eine Zeichenfolge,, `Buffer` TypedArray, Dataview oder Arraybuffer.
  - `encoding`: Optional. Wenn `value` eine Zeichenfolge ist, geben Sie eine der folgenden Optionen ein: `utf8`, `hex`, `base64`, `base64url`. Der Standardwert ist `utf8`.
- `Buffer.compare(buffer1, buffer2)`

  Vergleichen Sie zwei `Buffer`, um Arrays besser sortieren zu können. Gibt `0` zurück, wenn sie identisch sind, `-1`, wenn `buffer1` an erster Stelle steht, oder `1`, wenn `buffer2` an erster Stelle steht.
  - `buffer1`: Geben Sie einen `Buffer` ein.
  - `buffer2`: Geben Sie einen anderen `Buffer` ein.
- `Buffer.concat(list[, totalLength])`

  Verketten Sie mehrere `Buffer`. Gibt `0` zurück, wenn keiner vorhanden ist. Gibt bis zu `totalLength` zurück.
  - `list`: Geben Sie eine Liste von `Buffern` ein. Beachten Sie, dass dies auf `totalLength` gekürzt wird.
  - `totalLength`: Optional. Geben Sie eine Ganzzahl ohne Vorzeichen ein. Wenn das Feld leer ist, wird die Summe der `Buffer`-Instances in der Liste verwendet.
- `Buffer.from(array)`

  Erstellen Sie einen `Buffer` aus einem Array.
  - `array`: Geben Sie ein Byte-Array von `0` bis `255` ein.
- `Buffer.from(arrayBuffer, byteOffset[, length]))`

Erstellen Sie eine Ansicht von `arrayBuffer`, beginnend beim Versatz `byteOffset` mit der Länge `length`.

- `arrayBuffer`: Geben Sie ein `Buffer`-Array ein.
- `byteOffset`: Geben Sie eine Ganzzahl ein.
- `length`: Optional. Geben Sie eine Ganzzahl ein.

- `Buffer.from(buffer)`

  Erstellen Sie eine Kopie des `Buffers`.

  - `buffer`: Geben Sie einen `Buffer` ein.

- `Buffer.from(object[, offsetOrEncoding[, length]])`

  Erstellen Sie einen `Buffer` aus einem Objekt. Gibt `Buffer.from(object.valueOf(), offsetOrEncoding, length)` zurück, wenn `valueOf()` nicht dem Objekt entspricht.

  - `object`: Geben Sie ein Objekt ein.
  - `offsetOrEncoding`: Optional. Geben Sie eine Ganzzahl oder eine Kodierungszeichenfolge ein.
  - `length`: Optional. Geben Sie eine Ganzzahl ein.

- `Buffer.from(string[, encoding])`

  Erstellen Sie einen `Buffer` aus einer Zeichenfolge.

  - `string`: Geben Sie eine Zeichenfolge ein.
  - `encoding`: Optional. Machen Sie eine der folgenden Eingaben: `utf8`, `hex`, `base64`, `base64url`. Der Standardwert ist `utf8`.

- `Buffer.isBuffer(object)`

  Prüfen Sie, ob `object` ein Puffer ist. Gibt `true` oder `false` zurück.

  - `object`: Geben Sie ein Objekt ein.

- `Buffer.isEncoding(encoding)`

  Prüfen Sie, ob `encoding` unterstützt wird. Gibt `true` oder `false` zurück.

  - `encoding`: Optional. Machen Sie eine der folgenden Eingaben: `utf8`, `hex`, `base64`, `base64url`. Der Standardwert ist `utf8`.

Das Modul bietet die folgenden Puffer-Prototyp-Methoden:

- `Buffer.prototype.compare(target[, targetStart[, targetEnd[, sourceStart[, sourceEnd]]]])`

  Vergleichen Sie `Buffer` mit dem Ziel. Gibt `0` zurück, wenn sie identisch sind, `1`, wenn `buffer` an erster Stelle steht, oder `-1`, wenn `target` an erster Stelle steht.

  - `target`: Geben Sie einen `Buffer` ein.
  - `targetStart`: Optional. Geben Sie eine Ganzzahl ein. Standard = 0.
  - `targetEnd`: Optional. Geben Sie eine Ganzzahl ein. Die Standardeinstellung ist die `target`-Länge.
  - `sourceStart`: Optional. Geben Sie eine Ganzzahl ein. Standard = 0.
  - `sourceEnd`: Optional. Geben Sie eine Ganzzahl ein. Die Standardeinstellung ist die `Buffer`-Länge.
- `Buffer.prototype.copy(target[, targetStart[, sourceStart[, sourceEnd]]])`

  Kopieren Sie den Puffer nach `target`.

  - `target`: Geben Sie einen `Buffer` oder `Uint8Array` ein.
  - `targetStart`: Optional. Geben Sie eine Ganzzahl ein. Standard = 0.
  - `sourceStart`: Optional. Geben Sie eine Ganzzahl ein. Standard = 0.
  - `sourceEnd`: Optional. Geben Sie eine Ganzzahl ein. Die Standardeinstellung ist die `Buffer`-Länge.
- `Buffer.prototype.equals(otherBuffer)`

  Vergleichen Sie `Buffer` mit `otherBuffer`. Gibt `true` oder `false` zurück.

  - `otherBuffer`: Geben Sie eine Zeichenfolge ein.
- `Buffer.prototype.fill(value[, offset[, end][, encoding])`

  Geben Sie für den `Buffer` den Wert `value` ein.

  - `value`: Geben Sie eine Zeichenfolge, einen `Buffer` oder eine Ganzzahl ein.
  - `offset`: Optional. Geben Sie eine Ganzzahl ein.
  - `end`: Optional. Geben Sie eine Ganzzahl ein.
  - `encoding`: Optional. Machen Sie eine der folgenden Eingaben: `utf8`, `hex`, `base64`, `base64url`. Der Standardwert ist `utf8`.
- `Buffer.prototype.includes(value[, byteOffset][, encoding])`

Suchen Sie nach `value` im `Buffer`. Gibt `true` oder `false` zurück.

- `value`: Geben Sie eine Zeichenfolge, einen `Buffer`, ein `Uint8Array` oder eine Ganzzahl ein.
- `byteOffset`: Optional. Geben Sie eine Ganzzahl ein.
- `encoding`: Optional. Machen Sie eine der folgenden Eingaben: `utf8`, `hex`, `base64`, `base64url`. Der Standardwert ist `utf8`.

- `Buffer.prototype.indexOf(value[, byteOffset][, encoding])`

  Suchen Sie nach dem ersten `value` im `Buffer`. Gibt `index` zurück, wenn er gefunden wurde, oder gibt `-1` zurück, wenn er nicht gefunden wurde.

  - `value`: Geben Sie eine Zeichenfolge, einen `Buffer`, ein Unit8Array oder eine Ganzzahl von 0 bis 255 ein.
  - `byteOffset`: Optional. Geben Sie eine Ganzzahl ein.
  - `encoding`: Optional. Geben Sie eine der folgenden Optionen ein, wenn `value` eine Zeichenfolge ist: `utf8`, `hex`, `base64`, `base64url`. Der Standardwert ist `utf8`.

- `Buffer.prototype.lastIndexOf(value[, byteOffset][, encoding])`

  Suchen Sie nach dem letzten `value` im `Buffer`. Gibt `index` zurück, wenn er gefunden wurde, oder gibt `-1` zurück, wenn er nicht gefunden wurde.

  - `value`: Geben Sie eine Zeichenfolge, einen `Buffer`, ein Unit8Array oder eine Ganzzahl von 0 bis 255 ein.
  - `byteOffset`: Optional. Geben Sie eine Ganzzahl ein.
  - `encoding`: Optional. Geben Sie eine der folgenden Optionen ein, wenn `value` eine Zeichenfolge ist: `utf8`, `hex`, `base64`, `base64url`. Der Standardwert ist `utf8`.

- `Buffer.prototype.readInt8(offset)`

  Lesen Sie `Int8` beim `offset` vom `Buffer`.

  - `offset`: Geben Sie eine Ganzzahl ein.

- `Buffer.prototype.readIntBE(offset, byteLength)`

  Lesen Sie `Int` als Big-Endian beim `offset` vom `Buffer`.

  - `offset`: Geben Sie eine Ganzzahl ein.
  - `byteLength`: Optional. Geben Sie eine Ganzzahl von 1 bis 6 ein.

- `Buffer.prototype.readInt16BE(offset)`

Lesen Sie `Int16` als Big-Endian beim `offset` vom `Buffer`.

- `offset`: Geben Sie eine Ganzzahl ein.

- `Buffer.prototype.readInt32BE(offset)`

  Lesen Sie `Int32` als Big-Endian beim `offset` vom `Buffer`.

  - `offset`: Geben Sie eine Ganzzahl ein.

- `Buffer.prototype.readIntLE(offset, byteLength)`

  Lesen Sie `Int` als Little-Endian beim `offset` vom `Buffer`.

  - `offset`: Geben Sie eine Ganzzahl ein.
  - `byteLength`: Geben Sie eine Ganzzahl von 1 bis 6 ein.

- `Buffer.prototype.readInt16LE(offset)`

  Lesen Sie `Int16` als Little-Endian beim `offset` vom `Buffer`.

  - `offset`: Geben Sie eine Ganzzahl ein.

- `Buffer.prototype.readInt32LE(offset)`

  Lesen Sie `Int32` als Little-Endian beim `offset` vom `Buffer`.

  - `offset`: Geben Sie eine Ganzzahl ein.

- `Buffer.prototype.readUInt8(offset)`

  Lesen Sie `UInt8` beim `offset` vom `Buffer`.

  - `offset`: Geben Sie eine Ganzzahl ein.

- `Buffer.prototype.readUIntBE(offset, byteLength)`

  Lesen Sie `UInt` als Big-Endian beim `offset` vom `Buffer`.

  - `offset`: Geben Sie eine Ganzzahl ein.
  - `byteLength`: Geben Sie eine Ganzzahl von 1 bis 6 ein.

- `Buffer.prototype.readUInt16BE(offset)`

  Lesen Sie `UInt16` als Big-Endian beim `offset` vom `Buffer`.

- • `offset`: Geben Sie eine Ganzzahl ein.

- `Buffer.prototype.readUInt32BE(offset)`

  Lesen Sie `UInt32` als Big-Endian beim `offset` vom `Buffer`.

  - `offset`: Geben Sie eine Ganzzahl ein.
- `Buffer.prototype.readUIntLE(offset, byteLength)`

  Lesen Sie `UInt` als Little-Endian beim `offset` vom `Buffer`.

  - `offset`: Geben Sie eine Ganzzahl ein.
  - `byteLength`: Geben Sie eine Ganzzahl von 1 bis 6 ein.
- `Buffer.prototype.readUInt16LE(offset)`

  Lesen Sie `UInt16` als Little-Endian beim `offset` vom `Buffer`.

  - `offset`: Geben Sie eine Ganzzahl ein.
- `Buffer.prototype.readUInt32LE(offset)`

  Lesen Sie `UInt32` als Little-Endian beim `offset` vom `Buffer`.

  - `offset`: Geben Sie eine Ganzzahl ein.
- `Buffer.prototype.readDoubleBE([offset])`

  Lesen Sie einen 64-Bit-Double-Wert als Big-Endian beim `offset` vom `Buffer`.

  - `offset`: Optional. Geben Sie eine Ganzzahl ein.
- `Buffer.prototype.readDoubleLE([offset])`

  Lesen Sie einen 64-Bit-Double-Wert als Little-Endian beim `offset` vom `Buffer`.

  - `offset`: Optional. Geben Sie eine Ganzzahl ein.
- `Buffer.prototype.readFloatBE([offset])`

  Lesen Sie einen 32-Bit-Float-Wert als Big-Endian beim `offset` vom `Buffer`.

  - `offset`: Optional. Geben Sie eine Ganzzahl ein.
- `Buffer.prototype.readFloatLE([offset])`

  Lesen Sie einen 32-Bit-Float-Wert als Little-Endian beim `offset` vom `Buffer`.

  - `offset`: Optional. Geben Sie eine Ganzzahl ein.
- `Buffer.prototype.subarray([start[, end]])`

  Gibt eine Kopie vom `Buffer` zurück, der versetzt und mit einem neuen `start` und `end` zugeschnitten wurde.

  - `start`: Optional. Geben Sie eine Ganzzahl ein. Standard = 0.

  - `end`: Optional. Geben Sie eine Ganzzahl ein. Die Standardeinstellung ist die Pufferlänge.
- `Buffer.prototype.swap16()`

  Tauschen Sie die Byte-Reihenfolge des `Buffer`-Arrays aus und behandeln Sie es wie ein Array von 16-Bit-Zahlen. Die `Buffer`-Länge muss durch 2 teilbar sein, sonst erhalten Sie eine Fehlermeldung.
- `Buffer.prototype.swap32()`

  Tauschen Sie die Byte-Reihenfolge des `Buffer`-Arrays aus und behandeln Sie es wie ein Array von 32-Bit-Zahlen. Die `Buffer`-Länge muss durch 4 teilbar sein, sonst erhalten Sie eine Fehlermeldung.
- `Buffer.prototype.swap64()`

  Tauschen Sie die Byte-Reihenfolge des `Buffer`-Arrays aus und behandeln Sie es wie ein Array von 64-Bit-Zahlen. Die `Buffer`-Länge muss durch 8 teilbar sein, sonst erhalten Sie eine Fehlermeldung.
- `Buffer.prototype.toJSON()`

  Gibt `Buffer` als JSON zurück.
- `Buffer.prototype.toString([encoding[, start[, end]]])`

  Konvertieren Sie den `Buffer` von `start` bis `end` in eine kodierte Zeichenfolge.

  - `encoding`: Optional. Machen Sie eine der folgenden Eingaben: `utf8`, `hex`, `base64` oder `base64url`. Der Standardwert ist `utf8`.
  - `start`: Optional. Geben Sie eine Ganzzahl ein. Standard = 0.
  - `end`: Optional. Geben Sie eine Ganzzahl ein. Die Standardeinstellung ist die Pufferlänge.
- `Buffer.prototype.write(string[, offset[, length]][, encoding])`

  Schreiben Sie die codierte `string` in den `Buffer`, wenn genügend Platz vorhanden ist, oder die gekürzte `string`, wenn nicht genügend Platz vorhanden ist.

  - `string`: Geben Sie eine Zeichenfolge ein.
  - `offset`: Optional. Geben Sie eine Ganzzahl ein. Standard = 0.
  - `length`: Optional. Geben Sie eine Ganzzahl ein. Die Standardeinstellung ist die Länge der Zeichenfolge.

  - `encoding`: Optional. Geben Sie optional eine der folgenden Optionen ein: `utf8`, `hex`, `base64` oder `base64url`. Der Standardwert ist `utf8`.
- `Buffer.prototype.writeInt8(value, offset, byteLength)`

  Schreiben Sie den `Int8`-`value` der `byteLength` beim `offset` des `Buffers`.

  - `value`: Geben Sie eine Ganzzahl ein.
  - `offset`: Geben Sie eine Ganzzahl ein
  - `byteLength`: Geben Sie eine Ganzzahl von 1 bis 6 ein.
- `Buffer.prototype.writeIntBE(value, offset, byteLength)`

  Schreiben Sie den `value` beim `offset` des `Buffers` mit Big-Endian.

  - `value`: Geben Sie eine Ganzzahl ein.
  - `offset`: Geben Sie eine Ganzzahl ein
  - `byteLength`: Geben Sie eine Ganzzahl von 1 bis 6 ein.
- `Buffer.prototype.writeInt16BE(value, offset, byteLength)`

  Schreiben Sie den `value` beim `offset` des `Buffers` mit Big-Endian.

  - `value`: Geben Sie eine Ganzzahl ein.
  - `offset`: Geben Sie eine Ganzzahl ein
  - `byteLength`: Geben Sie eine Ganzzahl von 1 bis 6 ein.
- `Buffer.prototype.writeInt32BE(value, offset, byteLength)`

  Schreiben Sie den `value` beim `offset` des `Buffers` mit Big-Endian.

  - `value`: Geben Sie eine Ganzzahl ein.
  - `offset`: Geben Sie eine Ganzzahl ein
  - `byteLength`: Geben Sie eine Ganzzahl von 1 bis 6 ein.
- `Buffer.prototype.writeIntLE(offset, byteLength)`

  Schreiben Sie den `value` beim `offset` des `Buffers` mit Little-Endian.

  - `offset`: Geben Sie eine Ganzzahl ein.
  - `byteLength`: Geben Sie eine Ganzzahl von 1 bis 6 ein.
- `Buffer.prototype.writeInt16LE(offset, byteLength)`

  Schreiben Sie den `value` beim `offset` des `Buffers` mit Little-Endian.

  - `offset`: Geben Sie eine Ganzzahl ein.
  - `byteLength`: Geben Sie eine Ganzzahl von 1 bis 6 ein.
- `Buffer.prototype.writeInt32LE(offset, byteLength)`

  Schreiben Sie den `value` beim `offset` des `Buffers` mit Little-Endian.
  - `offset`: Geben Sie eine Ganzzahl ein.
  - `byteLength`: Geben Sie eine Ganzzahl von 1 bis 6 ein.
- `Buffer.prototype.writeUInt8(value, offset, byteLength)`

  Schreiben Sie den `UInt8-value` der `byteLength` beim `offset` des `Buffers`.
  - `value`: Geben Sie eine Ganzzahl ein.
  - `offset`: Geben Sie eine Ganzzahl ein
  - `byteLength`: Geben Sie eine Ganzzahl von 1 bis 6 ein.
- `Buffer.prototype.writeUIntBE(value, offset, byteLength)`

  Schreiben Sie den `value` beim `offset` des `Buffers` mit Big-Endian.
  - `value`: Geben Sie eine Ganzzahl ein.
  - `offset`: Geben Sie eine Ganzzahl ein
  - `byteLength`: Geben Sie eine Ganzzahl von 1 bis 6 ein.
- `Buffer.prototype.writeUInt16BE(value, offset, byteLength)`

  Schreiben Sie den `value` beim `offset` des `Buffers` mit Big-Endian.
  - `value`: Geben Sie eine Ganzzahl ein.
  - `offset`: Geben Sie eine Ganzzahl ein
  - `byteLength`: Geben Sie eine Ganzzahl von 1 bis 6 ein.
- `Buffer.prototype.writeUInt32BE(value, offset, byteLength)`

  Schreiben Sie den `value` beim `offset` des `Buffers` mit Big-Endian.
  - `value`: Geben Sie eine Ganzzahl ein.
  - `offset`: Geben Sie eine Ganzzahl ein
  - `byteLength`: Geben Sie eine Ganzzahl von 1 bis 6 ein.
- `Buffer.prototype.writeUIntLE(value, offset, byteLength)`

  Schreiben Sie den `value` beim `offset` des `Buffers` mit Little-Endian.

  - `value`: Geben Sie eine Ganzzahl ein.
  - `offset`: Geben Sie eine Ganzzahl ein
  - `byteLength`: Geben Sie eine Ganzzahl von 1 bis 6 ein.
- `Buffer.prototype.writeUInt16LE(value, offset, byteLength)`

  Schreiben Sie den `value` beim `offset` des `Buffers` mit Little-Endian.
  - `value`: Geben Sie eine Ganzzahl ein.
  - `offset`: Geben Sie eine Ganzzahl ein
  - `byteLength`: Geben Sie eine Ganzzahl von 1 bis 6 ein.
- `Buffer.prototype.writeUInt32LE(value, offset, byteLength)`

  Schreiben Sie den `value` beim `offset` des `Buffers` mit Little-Endian.
  - `value`: Geben Sie eine Ganzzahl ein.
  - `offset`: Geben Sie eine Ganzzahl ein
  - `byteLength`: Geben Sie eine Ganzzahl von 1 bis 6 ein.
- `Buffer.prototype.writeDoubleBE(value, [offset])`

  Schreiben Sie den `value` beim `offset` des `Buffers` mit Big-Endian.
  - `value`: Geben Sie eine Ganzzahl ein.
  - `offset`: Optional. Geben Sie eine Ganzzahl ein. Standard = 0.
- `Buffer.prototype.writeDoubleLE(value, [offset])`

  Schreiben Sie den `value` beim `offset` des `Buffers` mit Little-Endian.
  - `value`: Geben Sie eine Ganzzahl ein.
  - `offset`: Optional. Geben Sie eine Ganzzahl ein. Standard = 0.
- `Buffer.prototype.writeFloatBE(value, [offset])`

  Schreiben Sie den `value` beim `offset` des `Buffers` mit Big-Endian.
  - `value`: Geben Sie eine Ganzzahl ein.
  - `offset`: Optional. Geben Sie eine Ganzzahl ein. Standard = 0.
- `Buffer.prototype.writeFloatLE(value, [offset])`

  Schreiben Sie den `value` beim `offset` des `Buffers` mit Little-Endian.
  - `value`: Geben Sie eine Ganzzahl ein.

- `offset`: Optional. Geben Sie eine Ganzzahl ein. Standard = 0.

Die folgenden Instanzmethoden werden unterstützt:

- `buffer[index]`

  Rufen Sie Oktett (Byte) beim `index` im `Buffer` ab oder legen Sie es fest.

  - Rufen Sie eine Zahl von `0` bis `255` ab. Oder legen Sie eine Zahl von `0` bis `255` fest.

Die folgenden Instanzeigenschaften werden unterstützt:

- `buffer`

  Rufen Sie das `ArrayBuffer`-Objekt für den Puffer ab.
- `byteOffset`

  Rufen Sie das `byteOffset` vom `Arraybuffer`-Objekt des Puffers ab.
- `length`

  Rufen Sie die Byteanzahl des Puffers ab.

> **Note**
>
> Alle Methoden des Buffer-Moduls sind neu in Runtime 2.0. JavaScript

Abfragezeichenfolge

> **Note**
>
> Das CloudFront Functions-Ereignisobjekt analysiert automatisch URL-Abfragezeichenfolgen für Sie. Das bedeutet, dass Sie dieses Modul in den meisten Fällen nicht verwenden müssen.

Das Modul für Abfragezeichenfolgen (`querystring`) bietet Methoden zum Analysieren und Formatieren von URL-Abfragezeichenfolgen. Sie können das Modul mit `require('querystring')` laden. Das Modul bietet die folgenden Methoden.

`querystring.escape(`*`string`*`)`

URL-kodiert die angegebene `string` und gibt eine entflohene Abfragezeichenfolge zurück. Die Methode wird von `querystring.stringify()` verwendet und sollte nicht direkt verwendet werden.

`querystring.parse(`*`string`*`[, `*`separator`*`[, `*`equal`*`[, `*`options`*`]]])`

Analysiert eine Abfragezeichenfolge (`string`) und gibt ein Objekt zurück.

Der `separator`-Parameter ist eine Teilzeichenfolge zum Abgrenzen von Schlüssel- und Wertepaaren in der Abfragezeichenfolge. Standardmäßig ist dies &.

Der `equal`-Parameter ist eine Teilzeichenfolge zum Abgrenzen von Schlüsseln und Werten in der Abfragezeichenfolge. Standardmäßig ist dies =.

Der `options`-Parameter ist ein Objekt mit den folgenden Schlüsseln:

`decodeURIComponent` *`function`*

Eine Funktion zum Entschlüsseln von prozentkodierten Zeichen in der Abfragezeichenfolge. Standardmäßig ist dies `querystring.unescape()`.

`maxKeys` *`number`*

Die maximale Anzahl der Schlüssel zum Parsen. Standardmäßig ist dies `1000`. Verwenden Sie den Wert `0`, um die Beschränkungen für das Zählen von Schlüsseln aufzuheben.

Standardmäßig wird davon ausgegangen, dass prozentcodierte Zeichen innerhalb der Abfragezeichenfolge die UTF-8-Codierung verwenden. Ungültige UTF-8-Sequenzen werden durch das Ersatzzeichen `U+FFFD` ersetzt.

Zum Beispiel für die folgende Abfragezeichenfolge:

```
'name=value&abc=xyz&abc=123'
```

Der Rückgabewert von `querystring.parse()` ist:

```
{
name: 'value',
abc: ['xyz', '123']
```

```
}
```

`querystring.decode()` ist ein Alias für `querystring.parse()`.

`querystring.stringify(`*`object`*`[, `*`separator`*`[, `*`equal`*`[, `*`options`*`]]])`

Serialisiert ein `object` und gibt eine Abfragezeichenfolge zurück.

Der `separator`-Parameter ist eine Teilzeichenfolge zum Abgrenzen von Schlüssel- und Wertepaaren in der Abfragezeichenfolge. Standardmäßig ist dies &.

Der `equal`-Parameter ist eine Teilzeichenfolge zum Abgrenzen von Schlüsseln und Werten in der Abfragezeichenfolge. Standardmäßig ist dies =.

Der `options`-Parameter ist ein Objekt mit den folgenden Schlüsseln:

`encodeURIComponent` *`function`*

Die Funktion, die zum Konvertieren von URL-unsicheren Zeichen in die prozentuale Kodierung in der Abfragezeichenfolge verwendet wird. Standardmäßig ist dies `querystring.escape()`.

Standardmäßig werden Zeichen, die eine prozentuale Kodierung innerhalb der Abfragezeichenfolge erfordern, als UTF-8 codiert. Um eine andere Codierung zu verwenden, geben Sie die Option `encodeURIComponent` an.

Zum Beispiel für den folgenden Code:

```
querystring.stringify({ name: 'value', abc: ['xyz', '123'], anotherName: '' });
```

Der Rückgabewert ist:

```
'name=value&abc=xyz&abc=123&anotherName='
```

`querystring.encode()` ist ein Alias für `querystring.stringify()`.

`querystring.unescape(`*`string`*`)`

Dekodiert die prozentualen Zeichen der URL in der angegebenen `string` und gibt eine nicht entdeckene Abfragezeichenfolge zurück. Diese Methode wird von `querystring.parse()` verwendet und sollte nicht direkt verwendet werden.

Crypto

Das kryptographische Modul (`crypto`) bietet standardmäßige Hashing- und HMAC-Helfer (Hash-basierter Nachrichtenauthentifizierungscode). Sie können das Modul mit `require('crypto')` laden.

Hashing-Methoden

`crypto.createHash(`*`algorithm`*`)`

Erstellt und gibt ein Hash-Objekt zurück, mit dem Sie Hash-Digests mit dem angegebenen Algorithmus generieren können: `md5`, `sha1` oder `sha256`.

`hash.update(`*`data`*`)`

Aktualisiert den Hash-Inhalt mit dem angegebenen `data`.

`hash.digest([`*`encoding`*`])`

Berechnet den Digest aller mit `hash.update()` übergebenen Daten. Die Codierung kann `hex`, `base64` oder `base64url` sein.

HMAC-Methoden

`crypto.createHmac(`*`algorithm, secret key`*`)`

Erstellt und gibt ein HMAC-Objekt zurück, das das angegebene `algorithm` und `secret key` verwendet. Der Algorithmus kann `md5`, `sha1` oder `sha256` sein.

`hmac.update(`*`data`*`)`

Aktualisiert den HMAC-Inhalt mit den angegebenen `data`.

`hmac.digest([`*`encoding`*`])`

Berechnet den Digest aller mit `hmac.update()` übergebenen Daten. Die Codierung kann `hex`, `base64` oder `base64url` sein.

Eingeschränkte Features

Die folgenden JavaScript Sprachfunktionen werden entweder nicht unterstützt oder sind aus Sicherheitsgründen eingeschränkt.

Dynamische Codeauswertung

Die dynamische Codeauswertung wird nicht unterstützt. Sowohl `eval()`- als auch `Function`-Konstruktoren geben einen Fehler aus, wenn sie versucht werden. Zum Beispiel gibt `const sum = new Function('a', 'b', 'return a + b')` einen Fehler aus.

Timer

Die Funktionen `setTimeout()`, `setImmediate()` und `clearTimeout()` werden nicht unterstützt. Es gibt keine Bestimmung, innerhalb einer Funktionsausführung zu verschieben oder zu ergeben. Ihre Funktion muss synchron bis zum Abschluss ausgeführt werden.

Datum und Zeitstempel

Aus Sicherheitsgründen besteht kein Zugang zu hochauflösenden Timern. Alle `Date`-Methoden zum Abfragen der aktuellen Uhrzeit geben während der Lebensdauer einer einzelnen Funktionsausführung immer den gleichen Wert zurück. Der zurückgegebene Zeitstempel ist die Zeit, zu der die Funktion gestartet wurde. Folglich können Sie die verstrichene Zeit in Ihrer Funktion nicht messen.

Zugriff auf das Dateisystem

Es gibt keinen Zugriff auf das Dateisystem. Zum Beispiel gibt es kein `fs`-Modul für den Dateisystemzugriff wie in Node.js.

Prozesszugriff

Es gibt keinen Prozesszugriff. Zum Beispiel gibt es kein globales `process`-Objekt für die Verarbeitung von Informationszugriffen wie in Node.js.

Umgebungsvariablen

Es gibt keinen Zugriff auf Umgebungsvariablen. Stattdessen können Sie verwenden, CloudFront KeyValueStore um einen zentralen Datenspeicher mit Schlüssel-Wert-Paaren für Ihre Funktionen zu erstellen. CloudFront CloudFront KeyValueStore ermöglicht dynamische Aktualisierungen Ihrer Konfigurationsdaten, ohne dass Codeänderungen vorgenommen werden müssen. Weitere Informationen finden Sie unter Amazon CloudFront KeyValueStore.

Netzwerkzugriff

Es gibt keine Unterstützung für Netzwerkaufrufe. Zum Beispiel werden XHR, HTTP(S) und Socket nicht unterstützt.

Hilfsmethoden für Schlüsselwertspeicher

> **Note**
>
> Aufrufe von Hilfsmethoden zum Speichern von Schlüsselwerten aus CloudFront Functions lösen kein AWS CloudTrail Datenereignis aus. Diese Ereignisse werden nicht im CloudTrail Ereignisverlauf protokolliert. Weitere Informationen finden Sie unter Protokollieren Amazon CloudFront Amazon-API-Aufrufen mit AWS CloudTrail.

Dieser Abschnitt gilt, wenn Sie den CloudFront Key Value Store verwenden, um Schlüsselwerte in die von Ihnen erstellte Funktion aufzunehmen. CloudFront Functions verfügt über ein Modul, das drei Hilfsmethoden zum Lesen von Werten aus dem Schlüsselwertspeicher bereitstellt.

Um dieses Modul im Funktionscode zu verwenden, stellen Sie sicher, dass Sie der Funktion einen Schlüsselwertspeicher zugeordnet haben.

Fügen Sie als Nächstes die folgenden Anweisungen in die ersten Zeilen des Funktionscodes ein:

```
import cf from 'cloudfront';
const kvsHandle = cf.kvs();
```

Methode **`get ()`**

Verwenden Sie diese Methode, um den Schlüsselwert für den Schlüsselnamen abzurufen, den Sie angeben.

Anforderung

```
get("key", options);
```

- `key`: Der Name des Schlüssels, dessen Wert abgerufen werden muss
- `options`: Es gibt eine Option, `format`. Sie stellt sicher, dass die Funktion die Daten korrekt analysiert. Mögliche Werte:
  - `string`: (Standard) UTF8-kodiert
  - `json`
  - `bytes`: Roher Binärdatenpuffer

Beispiel anfordern

```
const value = await kvsHandle.get("myFunctionKey", { format: "string"});
```

Antwort

Die Antwort ist ein `promise`, das zu einem Wert in dem von `options` angeforderten Format aufgelöst wird. Standardmäßig wird der Wert als Zeichenfolge zurückgegeben.

Fehlerbehandlung

Die `get()`-Methode gibt einen Fehler zurück, wenn der von Ihnen angeforderte Schlüssel nicht im zugehörigen Schlüsselwertspeicher vorhanden ist. Um dieses Problem zu beheben, können Sie Ihrem Code einen `try`- und `catch`-Block hinzufügen.

Warning

Die Verwendung von Promise-Kombinatoren (z. B. `Promise.all`, `Promise.any`) und Promise-Verkettungsmethoden (z. B. `then` und `catch`) kann eine hohe Auslastung des Funktionsspeichers erfordern. Wenn Ihre Funktion das maximale Funktionsspeicherkontingent überschreitet, kann sie nicht ausgeführt werden. Um diesen Fehler zu vermeiden, empfehlen wir, die `await`-Syntax sequentiell oder in Schleifen zu verwenden, um mehrere Werte anzufordern.
Beispiel

```
var value1 = await kvs.get('key1');
var value2 = await kvs.get('key2');
```

Derzeit verbessert die Verwendung von Promise-Kombinatoren zum Abrufen mehrerer Werte die Leistung nicht, wie im folgenden Beispiel.

```
var values = await Promise.all([kvs.get('key1'), kvs.get('key2'),]);
```

Methode **`exists()`**

Verwenden Sie diese Methode, um zu ermitteln, ob der Schlüssel im Schlüsselwertspeicher vorhanden ist.

Anforderung

```
exists("key");
```

Beispiel anfordern

```
const exist = await kvsHandle.exists("myFunctionkey");
```

Antwort

Die Antwort ist ein `promise`, das einen booleschen Wert (`true` oder `false`) zurückgibt. Dieser Wert gibt an, ob der Schlüssel im Schlüsselwertspeicher vorhanden ist.

`meta ()` -Methode

Verwenden Sie diese Methode, um Metadaten über den Schlüsselwertspeicher zurückzugeben.

Anforderung

```
meta();
```

Beispiel anfordern

```
const meta = await kvsHandle.meta();
```

Antwort

Die Antwort ist ein `promise`, das in ein Objekt mit den folgenden Eigenschaften aufgelöst wird:

- `creationDateTime`: Erstellungsdatum und -uhrzeit des Schlüsselwertspeichers im ISO 8601-Format.
- `lastUpdatedDateTime`: Datum und Uhrzeit der letzten Synchronisierung des Schlüsselwertspeichers mit der Quelle im ISO 8601-Format. Der Wert beinhaltet nicht die Ausbreitungszeit bis zum Edge.
- `keyCount`: Die Gesamtzahl der Schlüssel im KVS nach der letzten Synchronisierung mit der Quelle.

Beispiel für eine Antwort

```
{keyCount:3,creationDateTime:2023-11-30T23:07:55.765Z,lastUpdatedDateTime:2023-12-15T03:57:52.4
```

Hilfsmethoden für die Änderung des Ursprungs

Dieser Abschnitt gilt, wenn Sie den für die Anfrage verwendeten Ursprung in Ihrem CloudFront Functions-Code dynamisch aktualisieren oder ändern. Sie können den Ursprung nur auf Anfrage des Betrachters aktualisieren (CloudFront Funktionen). CloudFront Functions verfügt über ein Modul, das Hilfsmethoden zur dynamischen Aktualisierung oder Änderung des Ursprungs bereitstellt.

Um dieses Modul zu verwenden, erstellen Sie eine CloudFront Funktion mit JavaScript Runtime 2.0 und fügen Sie die folgende Anweisung in die erste Zeile des Funktionscodes ein:

```
import cf from 'cloudfront';
```

Weitere Informationen finden Sie unter JavaScript Runtime 2.0-Funktionen für CloudFront Funktionen.

> **Note**
>
> Auf den Seiten der Test-API und Testkonsole wird nicht getestet, ob eine Ursprungsänderung stattgefunden hat. Beim Testen wird jedoch sichergestellt, dass der Funktionscode fehlerfrei ausgeführt wird.

Wählen Sie zwischen CloudFront Functions und Lambda @Edge

Sie können Ihre Ursprünge aktualisieren, indem Sie entweder CloudFront Functions oder Lambda @Edge verwenden.

Wenn Sie CloudFront Functions verwenden, um Ursprünge zu aktualisieren, verwenden Sie den Viewer-Request-Event-Trigger, was bedeutet, dass diese Logik bei jeder Anfrage ausgeführt wird, wenn diese Funktion verwendet wird. Bei der Verwendung von Lambda@Edge befinden sich die Funktionen zur Aktualisierung des Ursprungs im Ereignisauslöser für Ursprungsanforderungen, sodass diese Logik nur bei Cache-Fehlern ausgeführt wird.

Ihre Wahl hängt weitgehend von Ihrer Arbeitslast und der bestehenden Nutzung von CloudFront Functions und Lambda @Edge in Ihren Distributionen ab. Die folgenden Überlegungen können Ihnen bei der Entscheidung helfen, ob Sie CloudFront Functions oder Lambda @Edge verwenden möchten, um Ihre Ursprünge zu aktualisieren.

CloudFront Functions ist in den folgenden Situationen am nützlichsten:

- Wenn Ihre Anfragen dynamisch sind (was bedeutet, dass sie nicht zwischengespeichert werden können) und immer an den Ursprung weitergeleitet werden. CloudFront Functions bietet eine bessere Leistung und niedrigere Gesamtkosten.
- Wenn Sie bereits über eine CloudFront Viewer-Anforderungsfunktion verfügen, die bei jeder Anfrage ausgeführt wird, können Sie die ursprüngliche Aktualisierungslogik zur vorhandenen Funktion hinzufügen.

Informationen zur Verwendung von CloudFront Funktionen zur Aktualisierung von Ursprüngen finden Sie in den folgenden Themen in den Hilfsmethoden.

Lambda@Edge ist in folgenden Situationen am besten geeignet:

- Wenn Sie Inhalte haben, die in hohem Maße zwischengespeichert werden können, kann Lambda @Edge kostengünstiger sein, da es nur bei Cache-Fehlern ausgeführt wird, während CloudFront Functions bei jeder Anfrage ausgeführt wird.
- Wenn Sie bereits über eine Lambda@Edge-Funktion für Ursprungsanforderungen verfügen, können Sie die Ursprungsaktualisierungslogik zur vorhandenen Funktion hinzufügen.
- Diese Option ist auch von Vorteil, wenn Ihre Ursprungsaktualisierungslogik das Abrufen von Daten aus Datenquellen von Drittanbietern wie Amazon DynamoDB oder Amazon S3 erfordert.

Weitere Informationen zu Lambda@Edge finden Sie unter Anpassen am Edge mit Lambda@Edge.

update RequestOrigin () -Methode

Verwenden Sie die Methode `updateRequestOrigin()`, um die Ursprungseinstellungen für eine Anforderung zu aktualisieren. Sie können diese Methode verwenden, um bestehende Ursprungseigenschaften für Ursprünge zu aktualisieren, die bereits in Ihrer Distribution definiert sind, oder um einen neuen Ursprung für die Anforderung zu definieren. Geben Sie hierfür die zu ändernden Eigenschaften an.

> **Important**
>
> Alle Einstellungen, die Sie in der Methode `updateRequestOrigin()` nicht angeben, werden aus der vorhandenen Ursprungskonfiguration übernommen.

Der von der `updateRequestOrigin()` Methode festgelegte Ursprung kann ein beliebiger HTTP-Endpunkt sein und muss kein vorhandener Ursprung in Ihrer CloudFront Distribution sein.

Hinweise

- Wenn Sie einen Ursprung aktualisieren, der Teil einer Ursprungsgruppe ist, wird nur der primäre Ursprung der Ursprungsgruppe aktualisiert. Der sekundäre Ursprung bleibt unverändert. Jeder Antwortcode des geänderten Ursprungs, der den Failover-Kriterien entspricht, löst einen Failover zum sekundären Ursprung aus.
- Wenn Sie den Ursprungstyp ändern und OAC aktiviert ist, stellen Sie sicher, dass der Ursprungstyp in `originAccessControlConfig` mit dem neuen Ursprungstyp übereinstimmt.
- Sie können die Methode `updateRequestOrigin()` nicht verwenden, um VPC-Ursprünge zu aktualisieren. Eine solche Anforderung würde fehlschlagen.

Anforderung

```
updateRequestOrigin({origin properties})
```

`origin properties` kann die folgenden Werte enthalten:

domainName (optional)

Der Domänenname des Ursprungs. Bei fehlender Angabe wird der Domainname des zugewiesenen Ursprungs verwendet.

Für benutzerdefinierte Ursprünge

Geben Sie einen DNS-Domainnamen an, z. B. `www.example.com`. Der Domainname darf keinen Doppelpunkt (`:`) enthalten und darf keine IP-Adresse sein. Der Domänenname kann bis zu 253 Zeichen lang sein.

Für S3-Ursprünge

Geben Sie den DNS-Domainnamen des Amazon-S3-Buckets an, z. B. `amzn-s3-demo-bucket.s3.eu-west-1.amazonaws.com`. Der Name kann bis zu 128 Zeichen lang sein und muss in Kleinbuchstaben geschrieben werden.

HostHeader (optional, für benutzerdefinierte Ursprünge außerhalb von S3)

Der Host-Header, der verwendet werden soll, wenn die Anfrage an den Ursprung gestellt wird. Wenn dies nicht angegeben wird, wird der Wert aus dem DomainName-Parameter verwendet.

Wenn weder der Host-Header noch der Domain-Name-Parameter angegeben werden, wird der Domainname des zugewiesenen Ursprungs verwendet oder der Host-Header der eingehenden Anfrage, wenn die FTO-Richtlinie (Forward to Origin) den Host einschließt. Der Host-Header darf keinen Doppelpunkt (`:`) enthalten und darf keine IP-Adresse sein. Der Host-Header kann bis zu 253 Zeichen lang sein.

originPath (optional)

Der Verzeichnispfad auf dem Ursprung, aus dem die Anforderung den Inhalt abrufen soll. Der Pfad sollte mit einem Schrägstrich (/) beginnen, aber nicht damit enden. Zum Beispiel sollte er nicht mit `example-path/` enden. Bei fehlender Angabe wird der Ursprungspfad des zugewiesenen Ursprungs verwendet.

Für benutzerdefinierte Ursprünge

Der Pfad sollte URL-kodiert sein und eine maximale Länge von 255 Zeichen haben.

customHeaders (optional)

Sie können benutzerdefinierte Header in die Anforderung aufnehmen, indem Sie für jeden benutzerdefinierten Header einen Header-Namen und einen -Wert angeben. Das Format unterscheidet sich von dem der Anforderungs- und Antwortheader in der Ereignisstruktur. Verwenden Sie die folgende Syntax für das Schlüssel-Wert-Paar:

```
{"key1": "value1", "key2": "value2", ...}
```

Sie können keine Header hinzufügen, die nicht erlaubt sind, und ein Header mit dem gleichen Namen kann in den `headers` eingehender Anforderungen nicht vorhanden sein. Der Headername muss im Funktionscode in Kleinbuchstaben geschrieben werden. Wenn CloudFront Functions das Ereignisobjekt wieder in eine HTTP-Anforderung konvertiert, wird der erste Buchstabe jedes Worts in Header-Namen groß geschrieben und die Wörter werden durch einen Bindestrich getrennt.

Wenn der Funktionscode beispielsweise einen Header namens`example-header-name`, CloudFront konvertiert diesen in der HTTP-Anfrage `Example-Header-Name` in einen Header hinzufügt. Weitere Informationen erhalten Sie unter Benutzerdefinierte Header, die nicht zu CloudFront Quellanfragen hinzugefügt werden können und Einschränkungen für Edge-Funktionen.

Bei fehlender Angabe werden benutzerdefinierte Header des zugewiesenen Ursprungs verwendet.

connectionAttempts (optional)

Gibt an, wie oft CloudFront versucht wird, eine Verbindung zum Ursprung herzustellen. Der Mindestwert ist 1 und der Höchstwert 3. Bei fehlender Angabe wird die Anzahl von Verbindungsversuchen des zugewiesenen Ursprungs verwendet.

originShield (optional)

Dadurch wird CloudFront Origin Shield aktiviert oder aktualisiert. Die Verwendung von Origin Shield kann dazu beitragen, die Belastung Ihres Ursprungs zu reduzieren. Weitere Informationen finden Sie unter Verwenden Sie Amazon CloudFront Origin Shield. Bei fehlender Angabe werden die Origin-Shield-Einstellungen des zugewiesenen Ursprungs verwendet.

aktiviert (erforderlich)

Boolescher Ausdruck zur Aktivierung oder Deaktivierung von Origin Shield. Akzeptierte Werte sind `true` oder `false`.

Region (erforderlich, wenn aktiviert)

Das AWS-Region für Origin Shield. Geben Sie die AWS-Region an, die die niedrigste Latenz zu Ihrem Ursprung aufweist. Verwenden Sie den Regionscode, nicht den Namen der Region. Geben Sie beispielsweise `us-east-2` für die Region USA Ost (Ohio) an.

Wenn du CloudFront Origin Shield aktivierst, musst du das AWS-Region dafür angeben. Eine Liste der verfügbaren AWS-Regionen und Hilfe bei der Auswahl der besten Region für Ihren Ursprung finden Sie unter Wählen Sie das Symbol AWS Region für Origin Shield.

Herkunft AccessControlConfig (optional)

Die eindeutige Kennung der Ursprungszugriffssteuerung (OAC) für diesen Ursprung. Dies wird nur verwendet, wenn der Ursprung ein CloudFront OAC unterstützt, z. B. Amazon S3, Lambda-Funktions-URLs, Lambda Web Functions und V2 MediaStore. MediaPackage Bei fehlender Angabe werden die OAC-Einstellungen des zugewiesenen Ursprungs verwendet.

Die veraltete Ursprungszugriffsidentität (OAI) wird nicht unterstützt. Weitere Informationen finden Sie unter Beschränken Sie den Zugriff auf ein AWS Ursprungsserver.

aktiviert (erforderlich)

Boolescher Ausdruck zum Aktivieren oder Deaktivieren von OAC. Akzeptierte Werte sind `true` oder `false`.

signingBehavior (erforderlich, wenn aktiviert)

Gibt an, welche Anfragen CloudFront signiert werden (fügt Authentifizierungsinformationen hinzu). Geben Sie `always` für den häufigsten Anwendungsfall an. Weitere Informationen finden Sie unter Erweiterte Einstellungen für die Ursprungszugriffssteuerung.

Folgende Werte sind in diesem Feld möglich:

- `always`— CloudFront signiert alle ursprünglichen Anfragen und überschreibt dabei den `Authorization` Header der Viewer-Anfrage, falls vorhanden.
- `never`— signiert CloudFront keine ursprünglichen Anfragen. Dieser Wert deaktiviert die Ursprungszugriffssteuerung für den Ursprung.
- `no-override`— Wenn die Viewer-Anfrage den `Authorization` Header nicht enthält, wird die ursprüngliche Anfrage CloudFront signiert. Wenn die Viewer-Anfrage den `Authorization` Header enthält, wird die ursprüngliche Anfrage CloudFront nicht signiert und stattdessen der `Authorization` Header aus der Viewer-Anfrage weitergegeben.

  > **Warning**
  >
  > Um den `Authorization`-Header aus der Viewer-Anforderung zu übergeben, müssen Sie ihn einer Ursprungsanforderungsrichtlinie für alle Cache-Verhaltensweisen hinzufügen, die Ursprünge verwenden, die dieser Ursprungszugriffssteuerung zugeordnet sind. Weitere Informationen finden Sie unter Steuern von Ursprungsanforderungen anhand einer Richtlinie.

- `always-amz-auth`— CloudFront signiert alle ursprünglichen Anfragen, ersetzt aber nicht den `Authorization` Header aus der Viewer-Anfrage. CloudFront fügt seine Signatur in einen speziellen CloudFront Header ein und übergibt den `Authorization` Header des Viewers unverändert an den Ursprung. Dieser Wert ist für Lambda-Webfunktions-Ursprünge erforderlich und wird für keinen anderen Quelltyp unterstützt. Weitere Informationen finden Sie unter Verhalten beim Signieren.

signingProtocol (erforderlich, wenn aktiviert)

Das Signaturprotokoll des OAC, das festlegt, wie Anfragen CloudFront signiert (authentifiziert) werden. Geben Sie dies `sigv4` für die meisten Ursprünge an. Geben Sie `sigv4a` einen Amazon S3 Multi-Region Access Point-Ursprung oder einen globalen Lambda Web Function-Ursprung an.

originType (erforderlich, wenn aktiviert)

Der Typ des Ursprungs für diese OAC. Gültige Werte sind: `s3`, `mediapackagev2`, `mediastore` und `lambda`.

Timeouts (optional)

Timeouts, die Sie angeben können, wie lange versucht CloudFront werden soll, darauf zu warten, dass Origins antwortet oder Daten sendet. Bei fehlender Angabe werden die Timeout-Einstellungen des zugewiesenen Ursprungs verwendet.

> **Note**
>
> Sofern nicht anders vermerkt, unterstützen diese Timeouts sowohl benutzerdefinierte Ursprünge als auch Amazon-S3-Ursprünge.

readTimeout (optional)

`readTimeout` gilt für die beiden folgenden Werte:

- Wie lange (in Sekunden) auf eine Antwort CloudFront gewartet wird, nachdem eine Anfrage an den Ursprung weitergeleitet wurde.
- Wie lange (in Sekunden) nach dem Empfang eines Antwortpakets vom Ursprung und vor dem Empfang des nächsten Pakets CloudFront gewartet wird.

Das Mindestwert für das Timeout ist 1 Sekunde und der Höchstwert 120 Sekunden. Weitere Informationen finden Sie unter Antwort-Timeout.

Antwort CompletionTimeout (optional)

Die Zeit (in Sekunden), in der eine Anfrage vom CloudFront Ursprung geöffnet bleiben und auf eine Antwort warten kann. Wenn bis zu diesem Zeitpunkt noch keine vollständige Antwort vom Ursprung eingegangen ist, wird die Verbindung CloudFront beendet.

Der Wert für `responseCompletionTimeout` muss größer oder gleich dem für `readTimeout` angegebenen Wert sein. Weitere Informationen finden Sie unter Timeout für den Abschluss der Antwort.

behalten AliveTimeout (optional)

Dieses Timeout gilt nur für benutzerdefinierte Ursprünge, nicht für Amazon-S3-Ursprünge. (Bei S3-Ursprungskonfigurationen werden diese Einstellungen ignoriert.)

Der `keepAliveTimeout` gibt an, wie lange versucht CloudFront werden soll, die Verbindung zum Ursprung aufrechtzuerhalten, nachdem das letzte Paket der Antwort empfangen wurde. Das Mindestwert für das Timeout ist 1 Sekunde und der Höchstwert 120 Sekunden. Weitere Informationen finden Sie unter Keep-alive Timeout (nur benutzerdefinierte und VPC-Ursprünge).

connectionTimeout (optional)

Die Anzahl der Sekunden, die CloudFront gewartet wird, wenn versucht wird, eine Verbindung zum Ursprung herzustellen. Das Mindestwert für das Timeout ist 1 Sekunde und der Höchstwert 10 Sekunden. Weitere Informationen finden Sie unter Verbindungstimeout.

benutzerdefiniert OriginConfig (optional)

Verwenden Sie `customOriginConfig`, um Verbindungseinstellungen für Ursprünge anzugeben, die kein Amazon-S3-Bucket sind. Es gibt eine Ausnahme: Sie können diese Einstellung festlegen, wenn der S3-Bucket mit statischem Website-Hosting konfiguriert wird. (Bei anderen Arten von S3-Bucket-Konfigurationen werden diese Einstellungen ignoriert.) Wenn `customOriginConfig` nicht angegeben wird, werden die Einstellungen des zugewiesenen Ursprungs verwendet.

Port (erforderlich)

Der HTTP-Port, der für die Verbindung zum Ursprung CloudFront verwendet wird. Geben Sie den HTTP-Port an, den der Ursprung überwacht.

Protokoll (erforderlich)

Gibt das Protokoll (HTTP oder HTTPS) an, das für die Verbindung zum Ursprung CloudFront verwendet wird. Gültige Werte sind:

- `http`— verwendet CloudFront immer HTTP, um eine Verbindung zum Ursprung herzustellen
- `https`— verwendet CloudFront immer HTTPS, um eine Verbindung zum Ursprung herzustellen

sslProtocols (erforderlich)

Eine Liste, die das SSL/TLS Mindestprotokoll angibt, das CloudFront verwendet wird, wenn Sie über HTTPS eine Verbindung zu Ihrem Ursprung herstellen. Gültige Werte sind: `SSLv3`, `TLSv1`, `TLSv1.1` und `TLSv1.2`. Weitere Informationen finden Sie unter Mindest-SSL-Protokoll für Ursprung.

ip AddressType (optional)

Gibt den IP-Adresstyp an, der für die Verbindung zum Ursprung CloudFront verwendet wird. Gültige Werte sind: `ipv4`, `ipv6` und `dualstack`. Die Änderung von `ipAddressType` wird nur unterstützt, wenn auch die `domainName`-Eigenschaft geändert wird.

sni (optional, für benutzerdefinierte Ursprünge außerhalb von S3)

Die Server Name Indication (SNI) ist eine Erweiterung des Transport Layer Security (TLS) - Protokolls, mit der ein Client zu Beginn des TLS-Handshake-Prozesses angibt, mit welchem Hostnamen er eine Verbindung herstellen möchte. Dieser Wert sollte mit einem allgemeinen Namen auf einem TLS-Zertifikat auf Ihrem Ursprungsserver übereinstimmen. Andernfalls könnte Ihr Ursprungsserver einen Fehler ausgeben.

Wenn dies nicht angegeben wird, wird der Wert aus dem `hostHeader` Parameter verwendet. Wenn der Host-Header nicht angegeben wird, wird der Wert aus dem `domainName` Parameter verwendet.

Wenn weder der Host-Header noch der Domain-Name-Parameter angegeben werden, wird der Domainname des zugewiesenen Ursprungs verwendet oder der Host-Header der eingehenden Anfrage, wenn die FTO-Richtlinie (Forward to Origin) den Host einschließt. Das SNI darf keinen Doppelpunkt (`:`) enthalten und darf keine IP-Adresse sein. Das SNI kann bis zu 253 Zeichen lang sein.

erlaubt CertificateNames (optional, für benutzerdefinierte Ursprünge außerhalb von S3)

Sie können eine Liste gültiger Zertifikatsnamen hinzufügen, die verwendet werden sollen, CloudFront um die Domainübereinstimmung mit Ihrem Ursprungsserver-TLS-Zertifikat während des TLS-Handshakes mit Ihrem Ursprungsserver zu validieren. Dieses Feld erwartet eine Reihe gültiger Domainnamen und kann Platzhalterdomänen enthalten, wie z. `*.example.com`

Sie können bis zu 20 zulässige Zertifikatsnamen angeben. Jeder Zertifikatsname kann bis zu 64 Zeichen lang sein.

Example– Aktualisierung auf den Ursprung der Amazon-S3-Anforderung

Das folgende Beispiel ändert den Ursprung der Viewer-Anforderung in einen S3-Bucket, aktiviert OAC und setzt benutzerdefinierte Header zurück, die an den Ursprung gesendet wurden.

```
cf.updateRequestOrigin({
```

```
    "domainName" : "amzn-s3-demo-bucket-in-us-east-1.s3.us-east-1.amazonaws.com",
    "originAccessControlConfig": {
        "enabled": true,
        "signingBehavior": "always",
        "signingProtocol": "sigv4",
        "originType": "s3"
    },
    // Empty object resets any header configured on the assigned origin
    "customHeaders": {}
});
```

Example– Aktualisierung auf Application Load Balancer als Anforderungsursprung

Das folgende Beispiel ändert den Ursprung der Viewer-Anforderung in einen Application Load Balancer und legt einen benutzerdefinierten Header und Timeouts fest.

```
cf.updateRequestOrigin({
    "domainName" : "example-1234567890.us-east-1.elb.amazonaws.com",
    "timeouts": {
        "readTimeout": 30,
        "connectionTimeout": 5
    },
    "customHeaders": {
        "x-stage": "production",
        "x-region": "us-east-1"
    }
});
```

Example– Aktualisierung auf einen Ursprung mit aktiviertem Origin Shield

Im folgenden Beispiel ist Origin Shield für den Ursprung in der Distribution aktiviert. Der Funktionscode aktualisiert nur den für den Ursprung verwendeten Domainnamen und lässt alle anderen optionalen Parameter weg. In diesem Fall wird Origin Shield weiterhin mit dem geänderten Ursprungsdomainnamen verwendet, da die Origin-Shield-Parameter nicht aktualisiert wurden.

```
cf.updateRequestOrigin({
    "domainName" : "www.example.com"
});
```

Example— Aktualisieren Sie den Host-Header, das SNI und die Namen der zulässigen Zertifikate

> **Warning**
>
> In den meisten Anwendungsfällen müssen Sie diese Art der Änderung nicht für Anfragen verwenden, die an Ihren Absender gesendet werden. Diese Parameter sollten nur verwendet werden, wenn Sie wissen, welche Auswirkungen eine Änderung dieser Werte hat.

Im folgenden Beispiel werden der Domainname, der Host-Header, SNI und zulässige Zertifikate für die Anfrage auf den Ursprung geändert.

```
cf.updateRequestOrigin({
    "domainName": "www.example.com",
    "hostHeader": "test.example.com",
    "sni": "test.example.net",
    "allowedCertificateNames": ["*.example.com", "*.example.net"],
});
```

Methode select RequestOriginById ()

Verwenden Sie `selectRequestOriginById()`, um einen vorhandenen Ursprung zu aktualisieren, indem Sie einen anderen Ursprung auswählen, der bereits in Ihrer Distribution konfiguriert ist. Diese Methode verwendet dieselben Einstellungen, die durch den aktualisierten Ursprung definiert sind.

Diese Methode akzeptiert nur Ursprünge, die bereits in derselben Distribution definiert sind, die beim Ausführen der Funktion verwendet wurde. Ursprünge werden durch die Ursprungs-ID referenziert. Dabei handelt es sich um den Ursprungsnamen, den Sie bei der Einrichtung des Ursprungs definiert haben.

Wenn in Ihrer Distribution ein VPC-Ursprung konfiguriert ist, können Sie mit dieser Methode Ihren Ursprung auf Ihren VPC-Ursprung aktualisieren. Weitere Informationen finden Sie unter Zugriffsbeschränkung mit VPC-Ursprüngen.

Wenn Sie in Ihrer Distribution einen Ursprung mit aktiviertem Mutual TLS (mTLS) haben, können Sie diese Methode verwenden, um Ihren Ursprung auf Ihren TLS-enabled M-Ursprung zu aktualisieren. Weitere Informationen finden Sie unter Verwenden von CloudFront Funktionen mit gegenseitigem Ursprungs-TLS.

Anforderung

```
cf.selectRequestOriginById(origin_id, {origin_overrides})
```

Im vorherigen Beispiel `origin_id` handelt es sich um eine Zeichenfolge, die auf den Ursprungsnamen eines Ursprungs in der Distribution verweist, auf der die Funktion ausgeführt wird. Der `origin_overrides` Parameter kann Folgendes enthalten:

HostHeader (optional, für benutzerdefinierte Ursprünge außerhalb von S3)

Der Host-Header, der verwendet werden soll, wenn die Anfrage an den Ursprung gestellt wird. Wenn dies nicht angegeben wird, wird der Wert aus dem `domainName` Parameter verwendet.

Wenn weder der Host-Header noch der Domain-Namen-Parameter angegeben werden, wird der Domainname des zugewiesenen Ursprungs verwendet oder der Host-Header der eingehenden Anfrage, wenn die FTO-Richtlinie (Forward to Origin) den Host einschließt. Der Host-Header darf keinen Doppelpunkt (`:`) enthalten und darf keine IP-Adresse sein. Der Host-Header kann bis zu 253 Zeichen lang sein.

sni (optional, für benutzerdefinierte Ursprünge außerhalb von S3)

Die Server Name Indication (SNI) ist eine Erweiterung des Transport Layer Security (TLS) - Protokolls, mit der ein Client zu Beginn des TLS-Handshake-Prozesses angibt, mit welchem Hostnamen er eine Verbindung herstellen möchte. Dieser Wert sollte mit einem allgemeinen Namen auf einem TLS-Zertifikat auf Ihrem Ursprungsserver übereinstimmen. Andernfalls könnte Ihr Ursprungsserver einen Fehler ausgeben.

Wenn dies nicht angegeben wird, wird der Wert aus dem `hostHeader` Parameter verwendet. Wenn der Host-Header nicht angegeben wird, wird der Wert aus dem `domainName` Parameter verwendet.

Wenn weder der Host-Header noch der Domain-Name-Parameter angegeben werden, wird der Domainname des zugewiesenen Ursprungs verwendet oder der Host-Header der eingehenden Anfrage, wenn die FTO-Richtlinie (Forward to Origin) den Host einschließt. Das SNI darf keinen Doppelpunkt (`:`) enthalten und darf keine IP-Adresse sein. Das SNI kann bis zu 253 Zeichen lang sein.

erlaubt CertificateNames (optional, für benutzerdefinierte Ursprünge außerhalb von S3)

Sie können eine Liste gültiger Zertifikatsnamen hinzufügen, die verwendet werden sollen, CloudFront um die Domainübereinstimmung mit Ihrem Ursprungsserver-TLS-Zertifikat während des TLS-Handshakes mit Ihrem Ursprungsserver zu validieren. Dieses Feld erwartet eine Reihe gültiger Domainnamen und kann Platzhalterdomänen enthalten, wie z. `*.example.com`

Sie können bis zu 20 zulässige Zertifikatsnamen angeben. Jeder Zertifikatsname kann bis zu 64 Zeichen lang sein.

Anforderung

```
selectRequestOriginById(origin_id)
```

Im vorherigen Beispiel handelt es sich bei `origin_id` um eine Zeichenfolge, die auf den Namen eines Ursprungs in der Distribution verweist, in der die Funktion ausgeführt wird.

Example– Auswahl des Ursprungs der Amazon-S3-Anforderung

Im folgenden Beispiel wird der Ursprung namens `amzn-s3-demo-bucket-in-us-east-1` aus der Liste der mit der Distribution verknüpften Ursprünge ausgewählt und die Konfigurationseinstellungen des Ursprungs `amzn-s3-demo-bucket-in-us-east-1` werden auf die Anforderung angewendet.

```
cf.selectRequestOriginById("amzn-s3-demo-bucket-in-us-east-1");
```

Example– Auswahl des Anforderungsursprungs von Application Load Balancer

Im folgenden Beispiel wird ein Application Load Balancer namens `myALB-prod` als Ursprung aus der Liste der mit der Distribution verknüpften Ursprünge ausgewählt und die Konfigurationseinstellungen von `myALB-prod` werden auf die Anforderung angewendet.

```
cf.selectRequestOriginById("myALB-prod");
```

Example— Wählen Sie den Ursprung der Application Load Balancer Balancer-Anforderung aus und überschreiben Sie den Host-Header

Wie im vorherigen Beispiel wählt das folgende Beispiel einen Application Load Balancer Balancer-Ursprung aus, der `myALB-prod` aus der Liste der mit der Verteilung verknüpften Ursprünge benannt ist, und wendet die Konfigurationseinstellungen von `myALB-prod` auf die Anfrage an. Dieses Beispiel überschreibt jedoch den Host-Header-Wert mit. `origin_overrides`

```
cf.overrideRequestOrigin("myALB-prod",{
        "hostHeader" : "test.example.com"
});
```

Methode create RequestOriginGroup ()

Verwenden Sie `createRequestOriginGroup()`, um zwei Ursprünge zu definieren, die als Ursprungsgruppe für Failover in Szenarien verwendet werden sollen, die hohe Verfügbarkeit erfordern.

Eine Ursprungsgruppe umfasst zwei Ursprünge (einen primären und einen sekundären) und ein Failover-Kriterium, das Sie angeben. Sie erstellen eine Ursprungsgruppe, um Origin-Failover in CloudFront zu unterstützen. Wenn Sie mit dieser Methode eine Ursprungsgruppe erstellen oder aktualisieren, können Sie die Ursprungsgruppe anstelle eines einzelnen Ursprungs angeben. CloudFront führt unter Verwendung der Failover-Kriterien ein Failover vom primären Ursprung zum sekundären Ursprung durch.

Wenn in Ihrer Distribution ein VPC-Ursprung konfiguriert ist, können Sie diese Methode verwenden, um eine Ursprungsgruppe mithilfe eines VPC-Ursprungs zu erstellen. Weitere Informationen finden Sie unter Zugriffsbeschränkung mit VPC-Ursprüngen.

Wenn Sie in Ihrer Distribution einen Ursprung mit aktiviertem Mutual TLS (mTLS) haben, können Sie diese Methode verwenden, um eine Ursprungsgruppe zu erstellen, die Ihren M-Ursprung umfasst. TLS-enabled Weitere Informationen finden Sie unter Verwenden von CloudFront Funktionen mit gegenseitigem Ursprungs-TLS.

Anforderung

```
createRequestOriginGroup({origin_group_properties})
```

In den vorangegangenen Beispielen kann `origin_group_properties` Folgendes enthalten:

originIds (erforderlich)

Array von `origin_ids`, wobei `origin_id` eine Zeichenfolge ist, die auf den Namen eines Ursprungs in der Distribution verweist, in der die Funktion ausgeführt wird. Sie müssen zwei Ursprünge als Teil des Arrays angeben. Der erste Ursprung in der Liste ist der primäre Ursprung und der zweite dient als sekundärer Ursprung für Failover-Zwecke.

originOverrides (optional)

Einige erweiterte Einstellungen können mithilfe des Parameters überschrieben werden. `{origin_overrides}` `origin overrides` kann die folgenden Werte enthalten:

HostHeader (optional, für benutzerdefinierte Ursprünge außerhalb von S3)

Der Host-Header, der verwendet werden soll, wenn die Anfrage an den Ursprung gestellt wird. Wenn dies nicht angegeben wird, wird der Wert aus dem `domainName` Parameter verwendet.

Wenn weder der Host-Header noch der Domain-Namen-Parameter angegeben werden, wird der Domainname des zugewiesenen Ursprungs verwendet oder der Host-Header der eingehenden Anfrage, wenn die FTO-Richtlinie (Forward to Origin) den Host einschließt. Der Host-Header darf keinen Doppelpunkt (`:`) enthalten und darf keine IP-Adresse sein. Der Host-Header kann bis zu 253 Zeichen lang sein.

sni (optional, für benutzerdefinierte Ursprünge außerhalb von S3)

Die Server Name Indication (SNI) ist eine Erweiterung des TLS-Protokolls (Transport Layer Security), mit der ein Client zu Beginn des TLS-Handshaking-Prozesses angibt, mit welchem Hostnamen er eine Verbindung herstellen möchte. Dieser Wert sollte mit einem allgemeinen Namen auf einem TLS-Zertifikat auf Ihrem Ursprungsserver übereinstimmen, da Ihr Ursprungsserver andernfalls möglicherweise einen Fehler ausgibt.

Wenn dies nicht angegeben wird, wird der Wert aus dem `hostHeader` Parameter verwendet. Wenn der Host-Header nicht angegeben wird, wird der Wert aus dem `domainName` Parameter verwendet.

Wenn weder der Host-Header noch der Domain-Name-Parameter angegeben werden, wird der Domainname des zugewiesenen Ursprungs verwendet oder der Host-Header der eingehenden Anfrage, wenn die FTO-Richtlinie (Forward to Origin) den Host einschließt. Das SNI darf keinen Doppelpunkt (`:`) enthalten und darf keine IP-Adresse sein. Das SNI kann bis zu 253 Zeichen lang sein.

erlaubt CertificateNames (optional, für benutzerdefinierte Ursprünge außerhalb von S3)

Sie können eine Liste gültiger Zertifikatsnamen hinzufügen, die verwendet werden sollen, CloudFront um die Domainübereinstimmung mit Ihrem Ursprungsserver-TLS-Zertifikat während des TLS-Handshakes mit Ihrem Ursprungsserver zu validieren. Dieses Feld erwartet eine Reihe gültiger Domainnamen und kann Platzhalterdomänen enthalten, wie z. `*.example.com`

Sie können bis zu 20 zulässige Zertifikatsnamen angeben. Jeder Zertifikatsname kann bis zu 64 Zeichen lang sein.

selectionCriteria (optional)

Wählen Sie `default` aus, um die standardmäßigen Ursprungs-Failover-Kriterien zu verwenden, oder verwenden Sie die auf `media-quality-score` basierte Failover-Logik. Gültige Werte sind:

- `default` verwendet die Failover-Kriterien basierend auf den Statuscodes, die in `failoverCriteria` angegeben sind. Bei fehlender Angabe für `selectionCriteria` wird `default` verwendet.
- `media-quality-score` wird verwendet, wenn die Funktion zum Routing unter Berücksichtigung der Medienqualität verwendet wird.

failoverCriteria (erforderlich)

Eine Reihe von Statuscodes, die, wenn sie vom primären Ursprung zurückgegeben werden, einen Failover CloudFront zum sekundären Ursprung auslösen. Wenn Sie eine bestehende Ursprungsgruppe überschreiben, überschreibt dieses Array alle Failover-Statuscodes, die in der ursprünglichen Konfiguration der Ursprungsgruppe festgelegt sind.

Wenn Sie diese Option verwenden `media-quality-scoreselectionCriteria`, CloudFront wird versucht, Anfragen auf der Grundlage des Medienqualitätsfaktors weiterzuleiten. Wenn der ausgewählte Ursprung einen in diesem Array festgelegten Fehlercode zurückgibt, CloudFront erfolgt ein Failover zum anderen Ursprung.

Example– Erstellen einer Ursprungsgruppe für Anforderungen

Das folgende Beispiel erstellt eine Ursprungsgruppe für eine Anforderung unter Verwendung der Ursprungs-IDs. Diese Ursprungs-IDs stammen aus der Konfiguration der Ursprungsgruppe für die zur Ausführung dieser Funktion verwendete Distribution.

Optional können Sie diese Option verwenden, `originOverrides` um die ursprünglichen Gruppenkonfigurationen für `snihostHeader`, und `allowedCertificateNames` zu überschreiben.

```
import cf from 'cloudfront';

function handler(event) {
    cf.createRequestOriginGroup({
        "originIds": [
            {
                "originId": "origin-1",
                "originOverrides": {
                    "hostHeader": "hostHeader.example.com",
                    "sni": "sni.example.com",
```

```
                    "allowedCertificateNames": ["cert1.example.com",
 "cert2.example.com", "cert3.example.com"]
                }
            },
            {
                "originId": "origin-2",
                "originOverrides": {
                    "hostHeader": "hostHeader2.example.com",
                    "sni": "sni2.example.com",
                    "allowedCertificateNames": ["cert4.example.com",
 "cert5.example.com"]
                }
            }
        ],
        "failoverCriteria": {
            "statusCodes": [500]
        }
    });

    event.request.headers['x-hookx'] = { value: 'origin-overrides' };
    return event.request;
}
```

Hilfsmethoden für CloudFront SaaS Manager-Eigenschaften

Verwenden Sie die folgenden Hilfsfunktionen für CloudFront SaaS Manager, um Werte für Ihre Multi-Tenant-Verteilungen in der von Ihnen erstellten Funktion abzurufen. Um die Beispiele auf dieser Seite verwenden zu können, müssen Sie zuerst eine CloudFront Funktion mithilfe von JavaScript Runtime 2.0 erstellen. Weitere Informationen finden Sie unter JavaScript Runtime 2.0-Funktionen für CloudFront Funktionen.

Themen

- Verbindungsgruppen
- Distributionsmandanten

Verbindungsgruppen

Die Verbindungsgruppe, die Ihren Distributionsmandanten zugeordnet ist, hat einen Domainnamen.

Verwenden Sie das Feld `endpoint` des Unterobjekts `context` des Ereignisobjekts, um diesen Wert abzurufen.

Anforderung

```
const value = event.context.endpoint;
```

Antwort

Die Antwort ist ein `string`, der den Domainnamen der Verbindungsgruppe wie z. B. d111111abcdef8.cloudfront.net enthält. Das Feld `endpoint` wird nur angezeigt, wenn Ihre Funktion für eine Multi-Tenant-Distribution mit einer zugeordneten Verbindungsgruppe aufgerufen wird. Weitere Informationen finden Sie unter Context-Objekt.

Distributionsmandanten

CloudFront Functions verfügt über ein Modul, das den Zugriff auf bestimmte Werte für den Distributionsmandanten ermöglicht.

Um dieses Modul zu verwenden, fügen Sie die folgende Anweisung in die erste Zeile Ihres Funktionscodes ein:

```
import cf from 'cloudfront';
```

Sie können die folgenden Beispiele nur in der `handler`-Funktion verwenden, entweder direkt oder über eine Funktion mit verschachtelten Aufrufen.

`Tenant.idVerteilungsfeld`

Verwenden Sie dieses Feld, um den Wert der Distributionsmandanten-ID abzurufen.

Anforderung

```
const value = cf.distributionTenant.id;
```

Antwort

Die Antwort ist ein `string`, der die Distributionsmandanten-ID wie z. B. `dt_1a2b3c4d5e6f7` enthält.

Fehlerbehandlung

Wenn Ihre Funktion für eine Standarddistribution aufgerufen wird, wird bei einer Angabe im Feld `distributionTenant.id` der Typfehler `distributionTenant module is not available`

zurückgegeben. Um dieses Problem zu beheben, können Sie Ihrem Code einen `try`- und `catch`-Block hinzufügen.

Methode **`distribution Tenant.parameters.get ()`**

Verwenden Sie diese Methode, um den Wert für die von Ihnen angegebenen Parameternamen für den Distributionsmandanten zurückzugeben.

```
distributionTenant.parameters.get("key");
```

`key`: der Name des Parameters des Distributionsmandanten, für den Sie den Wert abrufen möchten

Anfrage

```
const value = distributionTenant.parameters.get("key");
```

Antwort

Die Antwort ist ein `string`, der den Wert für den Parameter des Distributionsmandanten enthält. Wenn Ihr Schlüsselname beispielsweise `TenantPath` lautet, könnte der Wert für diesen Parameter `tenant1` lauten.

Fehlerbehandlung

Sie können die folgenden Fehler erhalten:

- Wenn Ihre Funktion für eine Standarddistribution aufgerufen wird, gibt die Methode `distributionTenant.parameters.get()` den Typfehler `distributionTenant module is not available` zurück.
- Der Fehler `DistributionTenantParameterKeyNotFound` wird zurückgegeben, wenn der von Ihnen angegebene Parameter für den Distributionsmandanten nicht existiert.

Um diese Probleme zu beheben, können Sie Ihrem Code einen `try`- und `catch`-Block hinzufügen.

Verwenden Sie async und await

CloudFront Funktionen, die in JavaScript Runtime Functions 2.0 ausgeführt werden, stellen eine `async await` Syntax für den Umgang mit `Promise` Objekten bereit. Versprechen stehen für verzögerte Ergebnisse, auf die mit dem `await` Schlüsselwort in Funktionen zugegriffen werden kann, die als gekennzeichnet sind`async`. Verschiedene neue WebCrypto Funktionen verwenden Promises.

Weitere Informationen über `Promise`-Objekte finden Sie unter Promise.

> **Note**
>
> Sie müssen JavaScript Runtime 2.0 für die folgenden Codebeispiele verwenden. `await` kann nur innerhalb von `async`-Funktionen verwendet werden. `async`-Argumente und -Schließungen werden nicht unterstützt.

```
async function answer() {
    return 42;
}

// Note: async, await can be used only inside an async function. async arguments and
 closures are not supported.

async function handler(event) {
    // var answer_value = answer(); // returns Promise, not a 42 value
    let answer_value = await answer(); // resolves Promise, 42
    console.log("Answer"+answer_value);
    event.request.headers['answer'] = { value : ""+answer_value };
    return event.request;
}
```

Der folgende JavaScript Beispielcode zeigt, wie Versprechen mit der `then` Chain-Methode angezeigt werden. Sie können `catch` verwenden, um Fehler anzuzeigen.

> **Warning**
>
> Die Verwendung von Promise-Kombinatoren (z. B. `Promise.all`, `Promise.any`) und Promise-Verkettungsmethoden (z. B. `then` und `catch`) kann eine hohe Auslastung des Funktionsspeichers erfordern. Wenn Ihre Funktion das maximale Funktionsspeicherkontingent überschreitet, kann sie nicht ausgeführt werden. Um diesen Fehler zu vermeiden, empfehlen wir Ihnen, die `await`-Syntax anstelle von `promise`-Methoden zu verwenden.

```
async function answer() {
    return 42;
}
```

```
async function squared_answer() {
   return answer().then(value => value * value)
}
// Note: async, await can be used only inside an async function. async arguments and
 closures are not supported.
async function handler(event) {
    // var answer_value = answer(); // returns Promise, not a 42 value
    let answer_value = await squared_answer(); // resolves Promise, 42
    console.log("Answer"+answer_value);
    event.request.headers['answer'] = { value : ""+answer_value };
    return event.request;
}
```

CWT-Unterstützung für Funktionen CloudFront

Dieser Abschnitt enthält Einzelheiten zur Unterstützung von CBOR-Web-Tokens (CWT) in Ihren CloudFront Funktionen, die eine sichere tokenbasierte Authentifizierung und Autorisierung an Edge-Standorten ermöglichen. CloudFront Diese Unterstützung wird als Modul bereitgestellt, auf das Sie in Ihrer Funktion zugreifen können. CloudFront

Um dieses Modul zu verwenden, erstellen Sie eine CloudFront Funktion mit JavaScript Runtime 2.0 und fügen Sie die folgende Anweisung in die erste Zeile des Funktionscodes ein:

```
import cf from 'cloudfront';
```

Auf die mit diesem Modul verknüpften Methoden kann über zugegriffen werden (wobei * ein Platzhalter ist, der die verschiedenen im Modul vorhandenen Funktionen darstellt):

```
cf.cwt.*
```

Weitere Informationen finden Sie unter JavaScript Runtime 2.0-Funktionen für CloudFront Funktionen.

Derzeit unterstützt das Modul nur die MAC0-Struktur mit dem HS256 (HMAC-SHA256) -Algorithmus mit einer Grenze von 1 KB für die maximale Tokengröße.

Token-Struktur

Dieser Abschnitt behandelt die Tokenstruktur, die vom CWT-Modul erwartet wird. Das Modul erwartet, dass das Token korrekt gekennzeichnet und identifizierbar ist (zum Beispiel COSE MAC0).

Darüber hinaus folgt das Modul in Bezug auf die Struktur des Tokens den von CBOR Object Signing and Encryption (COSE) [RFC 8152] festgelegten Standards.

```
( // CWT Tag (Tag value: 61) --- optional
    ( // COSE MAC0 Structure Tag (Tag value: 17) --- required
        [
            protectedHeaders,
            unprotectedHeaders,
            payload,
            tag,
        ]
    )
)
```

Example: CWT unter Verwendung der COSE MAC0-Struktur

```
61( // CWT tag
    17( // COSE_MAC0 tag
        [
            { // Protected Headers
                1: 4  // algorithm : HMAC-256-64
            },
            { // Unprotected Headers
                4: h'53796d6d6574726963323536' // kid : Symmetric key id
            },
            { // Payload
                1: "https://iss.example.com", // iss
                2: "exampleUser", // sub
                3: "https://aud.example.com", // aud
                4: 1444064944, // exp
                5: 1443944944, // nbf
                6: 1443944944, // iat
            },
            h'093101ef6d789200' // tag
        ]
    )
)
```

> **Note**
>
> Das CWT-Tag ist beim Generieren von Tokens optional. Das COSE-Struktur-Tag ist jedoch erforderlich.

validateToken () -Methode

Die Funktion dekodiert und validiert ein CWT-Token mithilfe des angegebenen Schlüssels. Wenn die Validierung erfolgreich ist, gibt sie das dekodierte CWT-Token zurück. Andernfalls wird ein Fehler ausgegeben. Bitte beachten Sie, dass diese Funktion keine Validierung des Anspruchssatzes vornimmt.

Anforderung

```
cf.cwt.validateToken(token, handlerContext{key})
```

Parameters

Token (erforderlich)

Verschlüsseltes Token für die Validierung. Dies muss ein JavaScript Puffer sein.

HandlerContext (erforderlich)

Ein JavaScript Objekt, das den Kontext für den ValidateToken-Aufruf speichert. Derzeit wird nur die Schlüsseleigenschaft unterstützt.

Schlüssel (Erforderlich)

Geheimer Schlüssel für die Berechnung von Nachrichtenübersichten. Kann entweder als Zeichenfolge oder JavaScript als Buffer bereitgestellt werden.

Antwort

Wenn die `validateToken()` Methode ein erfolgreich validiertes Token zurückgibt, hat die Antwort `CWTObject` der Funktion das folgende Format. Nach der Decodierung werden alle Claim-Schlüssel als Zeichenketten dargestellt.

```
CWTObject {
```

```
    protectedHeaders,
    unprotectedHeaders,
    payload
}
```

Beispiel: Validieren Sie das Token, wobei das Kind als Teil des Tokens gesendet wird

Dieses Beispiel zeigt die CWT-Token-Validierung, bei der das Kind aus dem Header extrahiert wird. Das Kind wird dann an CloudFront Functions übergeben KeyValueStore , um den geheimen Schlüssel abzurufen, der zur Validierung des Tokens verwendet wurde.

```
import cf from 'cloudfront'

const CwtClaims = {
   iss: 1,
   aud: 3,
   exp: 4
}

async function handler(event) {
    try {
        let request = event.request;
        let encodedToken = request.headers['x-cwt-token'].value;
        let kid = request.headers['x-cwt-kid'].value;

        // Retrieve the secret key from the kvs
        let secretKey = await cf.kvs().get(kid);

        // Now you can use the secretKey to decode & validate the token.
        let tokenBuffer = Buffer.from(encodedToken, 'base64url');

        let handlerContext = {
           key: secretKey,
        }

        try {
            let cwtObj = cf.cwt.validateToken(tokenBuffer, handlerContext);

            // Check if token is expired
            const currentTime = Math.floor(Date.now() / 1000); // Current time in
 seconds
            if (cwtObj[CwtClaims.exp] && cwtObj[CwtClaims.exp] < currentTime) {
                return {
```

```
                    statusCode: 401,
                    statusDescription: 'Token expired'
                };
            }
        } catch (error) {
            return {
               statusCode: 401,
               statusDescription: 'Invalid token'
            };
         }
    } catch (error) {
        return {
            statusCode: 402,
            statusDescription: 'Token processing failed'
        };
     }
    return request;
}
```

generateToken () -Methode

Diese Funktion generiert ein neues CWT-Token unter Verwendung der bereitgestellten Nutzlast- und Kontexteinstellungen.

Anforderung

```
cf.cwt.generateToken(generatorContext, payload)
```

Parameters

GeneratorContext (erforderlich)

Dies ist ein JavaScript Objekt, das als Kontext für die Generierung des Tokens verwendet wird und die folgenden Schlüssel-Wert-Paare enthält:

cwtTag (optional)

Dieser Wert ist ein boolescher Wert, der, falls `true` angegeben, hinzugefügt werden soll. `cwtTag`

coseTag (erforderlich)

Gibt den COSE-Tag-Typ an. Unterstützt derzeit nur`MAC0`.

Schlüssel (Erforderlich)

Geheimer Schlüssel zur Berechnung des Nachrichtenüberblicks. Dieser Wert kann entweder eine Zeichenfolge oder JavaScript `Buffer` sein.

Nutzlast (erforderlich)

Token-Nutzlast für die Kodierung. Die Nutzlast muss ein Format haben. `CWTObject`

Antwort

Gibt einen JavaScript Buffer zurück, der das kodierte Token enthält.

Example: Generieren Sie ein CWT-Token

```
import cf from 'cloudfront';

const CwtClaims = {
    iss: 1,
    sub: 2,
    exp: 4
};

const CatClaims = {
    catu: 401,
    catnip: 402,
    catm: 403,
    catr: 404
};

const Catu = {
    host: 1,
    path: 2,
    ext: 3
};

const CatuMatchTypes = {
    prefix_match: 1,
    suffix_match: 2,
    exact_match: 3
};

const Catr = {
    renewal_method: 1,
```

```
    next_renewal_time: 2,
    max_uses: 3
};

async function handler(event) {
    try {
        const response = {
            statusCode: 200,
            statusDescription: 'OK',
            headers: {}
        };

        const commonAccessToken = {
            protected: {
                1: "5",
            },
            unprotected: {},
            payload: {
                [CwtClaims.iss]: "cloudfront-documentation",
                [CwtClaims.sub]: "cwt-support-on-cloudfront-functions",
                [CwtClaims.exp]: 1740000000,
                [CatClaims.catu]: {
                    [Catu.host]: {
                        [CatuMatchTypes.suffix_match]: ".cloudfront.net"
                    },
                    [Catu.path]: {
                        [CatuMatchTypes.prefix_match]: "/media/live-stream/cf-4k/"
                    },
                    [Catu.ext]: {
                        [CatuMatchTypes.exact_match]: [
                            ".m3u8",
                            ".ts",
                            ".mpd"
                        ]
                    }
                },
                [CatClaims.catnip]: [
                    "[IP_ADDRESS]",
                    "[IP_ADDRESS]"
                ],
                [CatClaims.catm]: [
                    "GET",
                    "HEAD"
                ],
```

```
                [CatClaims.catr]: {
                    [Catr.renewal_method]: "header_renewal",
                    [Catr.next_renewal_time]: 1750000000,
                    [Catr.max_uses]: 5
                }
            }
        };

        if (!request.headers['x-cwt-kid']) {
            throw new Error('Missing x-cwt-kid header');
        }

        const kid = request.headers['x-cwt-kid'].value;
        const secretKey = await cf.kvs().get(kid);

        if (!secretKey) {
            throw new Error('Secret key not found for provided kid');
        }

        try {
            const genContext = {
                cwtTag: true,
                coseTag: "MAC0",
                key: secretKey
            };

            const tokenBuffer = cf.cwt.generateToken(commonAccessToken, genContext);
            response.headers['x-generated-cwt-token'] = { value:
tokenBuffer.toString('base64url') };

            return response;
        } catch (tokenError) {
            return {
                statusCode: 401,
                statusDescription: 'Could not generate the token'
            };
        }
    } catch (error) {
        return {
            statusCode: 402,
            statusDescription: 'Token processing failed'
        };
    }
```

```
}
```

Example: Aktualisieren Sie das Token auf der Grundlage einer bestimmten Logik

```
import cf from 'cloudfront'

const CwtClaims = {
   iss: 1,
   aud: 3,
   exp: 4
}

async function handler(event) {
    try {
        let request = event.request;
        let encodedToken = request.headers['x-cwt-token'].value;
        let kid = request.headers['x-cwt-kid'].value;
        let secretKey = await cf.kvs().get(kid); // Retrieve the secret key from the
 kvs

        // Now you can use the secretKey to decode & validate the token.
        let tokenBuffer = Buffer.from(encodedToken, 'base64url');

        let handlerContext = {
           key: secretKey,
        }

        try {
            let cwtJSON = cf.cwt.validateToken(tokenBuffer, handlerContext);

            // Check if token is expired
            const currentTime = Math.floor(Date.now() / 1000); // Current time in
 seconds
            if (cwtJSON[CwtClaims.exp] && cwtJSON[CwtClaims.exp] < currentTime) {
                // We can regnerate the token and add 8 hours to the expiry time
                cwtJSON[CwtClaims.exp] = Math.floor(Date.now() / 1000) + (8 * 60 * 60);

                let genContext = {
                  coseTag: "MAC0",
                  key: secretKey
                }

                let newTokenBuffer = cf.cwt.generateToken(cwtJSON, genContext);
```

```
                request.headers['x-cwt-regenerated-token'] =
 newTokenBuffer.toString('base64url');
            }
        } catch (error) {
            return {
               statusCode: 401,
               statusDescription: 'Invalid token'
            };
         }
    }
    catch (error) {
        return {
            statusCode: 402,
            statusDescription: 'Token processing failed'
        };
     }
    return request;
}
```

Allgemeine Hilfsmethoden

Diese Seite bietet zusätzliche Hilfsmethoden innerhalb von CloudFront Functions. Um diese Methoden zu verwenden, erstellen Sie eine CloudFront Funktion mit JavaScript Runtime 2.0.

```
import cf from 'cloudfront';
```

Weitere Informationen finden Sie unter JavaScript Runtime 2.0-Funktionen für CloudFront Funktionen.

`EdgeLocation-Metadaten`

Diese Methode erfordert die Verwendung des `cloudfront` Moduls.

> **Note**
>
> Sie können diese Methode nur für Viewer-Request-Funktionen verwenden. Für Viewer-Response-Funktionen ist diese Methode leer.

Verwenden Sie dieses JavaScript Objekt, um den Flughafencode für den Edge-Standort, die erwartete regionale Edge-Cache-Region oder die CloudFront Server-IP-Adresse abzurufen, die

zur Bearbeitung der Anfrage verwendet wurde. Diese Metadaten sind nur für den Event-Trigger der Viewer-Anforderung verfügbar.

```
cf.edgeLocation = {
    name: SEA
    serverIp: 1.2.3.4
    region: us-west-2
}
```

Das `cf.edgeLocation` Objekt kann Folgendes enthalten:

Name

Der aus drei Buchstaben bestehende IATA-Code des Edge-Standorts, der die Anfrage bearbeitet hat.

Server-IP

Die IPv4- oder IPv6-Adresse des Servers, der die Anfrage bearbeitet hat.

Region

Der CloudFront Regional Edge Cache (REC), den die Anfrage voraussichtlich verwenden wird, wenn ein Cache-Fehler auftritt. Dieser Wert wird nicht aktualisiert, falls der erwartete REC nicht verfügbar ist und ein Backup-REC für die Anfrage verwendet wird. Dies beinhaltet nicht, dass der Origin Shield-Standort verwendet wird, außer in Fällen, in denen sich der primäre REC und der Origin Shield am selben Standort befinden.

> **Note**
>
> CloudFront Functions wird kein zweites Mal aufgerufen, wenn CloudFront es für die Verwendung von Origin-Failover konfiguriert ist. Weitere Informationen finden Sie unter Optimieren Sie die Hochverfügbarkeit mit CloudFront Origin-Failover.

`raw QueryString () -Methode`

Für diese Methode ist das `cloudFront` Modul nicht erforderlich.

Verwenden Sie die `rawQueryString()` Methode, um die ungeparste und unveränderte Abfragezeichenfolge als Zeichenfolge abzurufen.

Anforderung

```
function handler(event) {
    var request = event.request;
    const qs = request.rawQueryString();
}
```

Antwort

Gibt die vollständige Abfragezeichenfolge der eingehenden Anfrage als Zeichenkettenwert ohne vorangestellte Zeichenfolge zurück. ?

- Wenn es keine Abfragezeichenfolge gibt, diese aber vorhanden ? ist, gibt die Funktion eine leere Zeichenfolge zurück.
- Wenn keine Abfragezeichenfolge vorhanden ist und die ? nicht vorhanden ist, kehrt die Funktion zurück`undefined`.

Fall 1: Die gesamte Abfragezeichenfolge wird zurückgegeben (ohne vorangestelltes Zeichen**?**)

URL der eingehenden Anfrage: `https://example.com/page?name=John&age=25&city=Boston`

`rawQueryString()`gibt zurück: `"name=John&age=25&city=Boston"`

Fall 2: Es wird eine leere Zeichenfolge zurückgegeben (wenn sie vorhanden **?** ist, aber ohne Parameter)

URL der eingehenden Anfrage: `https://example.com/page?`

`rawQueryString()`gibt zurück: `""`

Fall 3: **`undefined`** zurückgegeben (keine Abfragezeichenfolge und nein**?**)

URL der eingehenden Anfrage: `https://example.com/page`

`rawQueryString()`gibt zurück: `undefined`

`log CustomData ()` -Methode

Um diese Methode zu verwenden, importieren Sie das `cloudfront` Modul.

Verwenden Sie die `logCustomData()` Methode, um benutzerdefinierte Daten aus Ihrer CloudFront Amazon-Funktion in CloudFront Zugriffsprotokolle zu senden. Die Daten werden in das `viewer-response-log-data` Protokollfeld `viewer-request-log-data` oder geschrieben, je nachdem, ob die Funktion mit einer Zuschaueranfrage oder einem Zuschauerantwortereignis verknüpft ist.

Die Methode akzeptiert ein einzelnes Zeichenkettenargument. Um ein Objekt zu protokollieren, `JSON.stringify()` konvertieren Sie es zuerst in eine Zeichenfolge. Beispiel:

```
const resp_obj = JSON.stringify(some_obj);
cf.logCustomData(resp_obj);
```

Jedes Feld unterstützt bis zu 800 Byte an Daten. CloudFront automatisch URL-encodes die Daten, bevor sie in das Protokollfeld geschrieben werden. CloudFront schneidet danach alle Daten ab, die größer als 800 Byte sind. URL-encoding Wenn Ihre Funktion innerhalb einer einzigen Ausführung `logCustomData()` mehrmals aufgerufen wird, CloudFront wird nur der letzte Wert verwendet.

Fügen Sie Ihrer Protokollkonfiguration Felder hinzu

Um diese Daten in Ihren CloudFront Zugriffsprotokollen zu erhalten, müssen Sie die `viewer-response-log-data` Felder `viewer-request-log-data` oder zu Ihrer ??? Echtzeit-Protokollkonfiguration oder zur Standardprotokollierung (v2) hinzufügen.

Unterschied zu console.log ()

Die `logCustomData()` Methode ersetzt `console.log()` CloudFront Funktionen nicht. Wird verwendet`console.log()`, um Protokollzeilen an Amazon CloudWatch Logs zu senden. Wird verwendet`cf.logCustomData()`, um benutzerdefinierte Daten für den CloudFront Zugriff auf Protokolle zu schreiben (Echtzeitprotokolle und Standardprotokollierung (v2)). Sie können beide Methoden in derselben Funktion verwenden. Weitere Informationen finden Sie unter CloudFront Funktionsprotokolle.

Die Methode verwendet die folgende Syntax:

```
cf.logCustomData(String);
```

Example Protokollieren Sie einen Header in einer Viewer-Anforderungsfunktion

```
import cf from 'cloudfront';

function handler(event) {
    var request = event.request;

    // Check if the debug header exists, if so log it
    if (request.headers['x-debug-header']) {
        // Log the debug header value
        cf.logCustomData("debug header found: " + request.headers['x-debug-
header'].value);
    }

    return request;
}
```

Hilfsmethoden für gegenseitiges TLS

CloudFront stellt m TLS-specific Hilfsfunktionen im `cf.mtls` Namespace für Viewer-Anforderungsfunktionen bereit. Diese Funktionen benennen die Header von Client-Zertifikaten um, formatieren sie neu oder kombinieren sie, bevor sie die Anfrage an Ihren Ursprung weiterleiten.

```
import cf from "cloudfront";
```

`cf.mtls.rename ClientCertHeaders`

Benennt die Header der Zertifikatsmetadaten in benutzerdefinierte Header-Namen um.

```
cf.mtls.renameClientCertHeaders({
    "CloudFront-Viewer-Cert-Pem": "X-Client-Cert",
    "CloudFront-Viewer-Cert-Serial-Number": "X-Client-Cert-Serial",
    "CloudFront-Viewer-Cert-Issuer": "X-Client-Cert-Issuer",
});
```

Zulässige Quell-Header:

- `CloudFront-Viewer-Cert-Pem`
- `CloudFront-Viewer-Cert-Serial-Number`
- `CloudFront-Viewer-Cert-Issuer`
- `CloudFront-Viewer-Cert-Subject`

- `CloudFront-Viewer-Cert-Validity`
- `CloudFront-Viewer-Cert-Present`
- `CloudFront-Viewer-Cert-Sha256`
- `Client-Cert`
- `Client-Cert-Chain`

`cf.mtls.rename PemHeaders`

Benennt die PEM-Zertifikatsheader um und formatiert optional die Zertifikatskodierung neu.

```
cf.mtls.renamePemHeaders({
    "Client-Cert": {
        "newHeaderName": "X-Leaf-Cert",
        "pemCertFormatInfo": {
            "certHeader": "-----CUSTOM HEADER-----",
            "certFooter": "-----CUSTOM FOOTER-----",
            "certEndMarker": "",
            "keepNewlinesInCertData": true
        }
    },
    "Client-Cert-Chain": {
        "newHeaderName": "X-Intermediate-Certs",
        "pemCertFormatInfo": {
            "certHeader": "-----CUSTOM HEADER-----",
            "certFooter": "-----CUSTOM FOOTER-----",
            "certEndMarker": "",
            "keepNewlinesInCertData": true
        }
    }
});
```

Zulässige Quell-Header:

- Im Passthrough-Modus:, `Client-Cert` `Client-Cert-Chain`
- Im required/optional Modus: `Cloudfront-Viewer-Cert-PEM`

`pemCertFormatInfo`Felder:

Für `Cloudfront-Viewer-Cert-PEM`:

- `certHeader` ersetzt `-----BEGIN CERTIFICATE-----`.
- `certFooter` ersetzt `-----END CERTIFICATE-----`.
- `certEndMarker`legt nach dem CertFooter eine benutzerdefinierte Zeichenfolge fest.
- `keepNewlinesInCertData`(Standard:`true`) behält Zeilenumbrüche in Base64-Daten bei, wenn der Wert wahr ist.

Für `Client-Cert` und `Client-Cert-Chain`:

- `certHeader` ersetzt `:`.
- `certFooter` ersetzt `:`.
- `certEndMarker`legt nach dem CertFooter eine benutzerdefinierte Zeichenfolge fest.
- `keepNewlinesInCertData`(Standard:`false`) behält Zeilenumbrüche in Base64-Daten bei, wenn der Wert wahr ist.

`cf.mtls.combine PemHeaders`

Kombiniert `Client-Cert` und `Client-Cert-Chain` zu einem einzigen Header, der die gesamte Zertifikatskette enthält.

```
cf.mtls.combinePemHeaders({
    "newHeaderName": "X-Full-Chain",
    "pemCertFormatInfo": {
        "certHeader": "-----BEGIN CERTIFICATE-----",
        "certFooter": "-----END CERTIFICATE-----",
        "certEndMarker": "\n",
        "keepNewlinesInCertData": false
    }
});
```

`certEndMarker`legt das Trennzeichen zwischen Zertifikaten fest.

> **Note**
>
> - Diese Hilfsfunktionen können in allen mTLS-Modi (erforderlich, optional und Passthrough) verwendet werden. Hat jedoch `cf.mtls.combinePemHeaders` nur im Passthrough-Modus eine Wirkung — im erforderlichen und optionalen Modus sind die `Client-Cert-`

`Chain` Header `Client-Cert` und nicht vorhanden, sodass die Funktion nicht verfügbar ist.

- Wenn die `cf.mtls.*` Methoden und `cf.updateRequestOrigin()` das `customHeaders` Ziel denselben Header-Namen haben, wird ein CloudFront 502-Fehler zurückgegeben.
- Löscht im Passthrough-Modus alle CloudFront eingehenden Header `Client-Cert` oder `Client-Cert-Chain` Header und fügt sie erneut aus dem eigentlichen Client-Zertifikat hinzu.
- CloudFront präsentiert in keiner Edge-Funktion den vollständigen Inhalt des PEM-Rohzertifikats.

Erstellen von Funktionen

Sie erstellen eine Funktion in zwei Phasen:

1. Erstellen Sie den Funktionscode als JavaScript. Sie können das Standardbeispiel aus der CloudFront Konsole verwenden oder Ihr eigenes schreiben. Weitere Informationen finden Sie unter den folgenden Themen:
   - Schreiben von Funktionscode
   - the section called “Ereignisstruktur”
   - CloudFront Funktionen, Beispiele für CloudFront
2. Verwenden Sie CloudFront , um die Funktion zu erstellen und Ihren Code einzufügen. Der Code ist innerhalb der Funktion vorhanden (nicht als Referenz).

Console

Eine Funktion erstellen

1. Melden Sie sich bei der CloudFront Konsole unter an https://console.aws.amazon.com/cloudfront/v4/home#/functions und wählen Sie die Seite Funktionen.
2. Wählen Sie Funktion erstellen.
3. Geben Sie einen Funktionsnamen ein, der innerhalb von eindeutig ist AWS-Konto, wählen Sie die JavaScript Version aus, und klicken Sie dann auf Weiter. Die Seite mit den Details der neuen Funktion wird angezeigt.

> **Note**
>
> Um Schlüssel-Wert-Paare in der Funktion zu verwenden, müssen Sie JavaScript Runtime 2.0 wählen.
> Wenn beim Erstellen einer Funktion mit Tags ein Fehler `TaggingConflict` (HTTP 409) auftritt, warten Sie ein paar Minuten und versuchen Sie es erneut, oder erstellen Sie die Funktion zunächst ohne Tags und fügen Sie sie anschließend hinzu.

4. Wählen Sie im Abschnitt Funktionscode die Registerkarte Erstellen aus und geben Sie Ihren Funktionscode ein. Der auf der Registerkarte Erstellen angezeigte Beispielcode veranschaulicht die grundlegende Syntax für den Funktionscode.
5. Wählen Sie Änderungen speichern aus.
6. Wenn der Funktionscode Schlüssel-Wert-Paare verwendet, müssen Sie einen Schlüsselwertspeicher zuordnen.

   Sie können den Schlüsselwertspeicher beim Erstellen der Funktion zuordnen. Sie können ihn auch später zuordnen, indem Sie die Funktion aktualisieren.

   Gehen Sie wie folgt vor, um jetzt einen Schlüsselwertspeicher zuzuordnen:

   - Gehen Sie zum KeyValueStore Abschnitt Associate und wählen Sie Associate existing aus. KeyValueStore
   - Wählen Sie den Schlüsselwertspeicher aus, der die Schlüssel-Wert-Paare in der Funktion enthält, und wählen Sie dann Associate. KeyValueStore

   CloudFront ordnet den Speicher sofort der Funktion zu. Sie müssen die Funktion nicht speichern.

CLI

Wenn Sie die CLI verwenden, erstellen Sie normalerweise zuerst den Funktionscode in einer Datei und dann die Funktion mit der AWS CLI.

Eine Funktion erstellen

1. Erstellen Sie den Funktionscode in einer Datei und speichern Sie ihn in einem Verzeichnis, mit dem Ihr Computer eine Verbindung herstellen kann.

2. Führen Sie den Befehl wie im Beispiel veranschaulicht aus. In diesem Beispiel wird die Notation `fileb://` verwendet, um die Datei zu übergeben. Es sind Zeilenumbrüche enthalten, um den Befehl lesbarer zu machen.

```
aws cloudfront create-function \
    --name MaxAge \
    --function-config '{"Comment":"Max Age 2 years","Runtime":"cloudfront-
js-2.0","KeyValueStoreAssociations":{"Quantity":1,"Items":
[{"KeyValueStoreARN":"arn:aws:cloudfront::111122223333:key-value-store/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"}]}}' \
    --function-code fileb://function-max-age-v1.js
```

> **Hinweise**
>
> - `Runtime`— Die Version von JavaScript. Wenn Sie Schlüssel-Wert-Paare in der Funktion verwenden möchten, müssen Sie Version 2.0 angeben.
> - `KeyValueStoreAssociations` – Wenn Ihre Funktion Schlüssel-Wert-Paare verwendet, können Sie den Schlüsselwertspeicher beim Erstellen der Funktion zuordnen. Oder Sie können ihn später mit `update-function` zuordnen. Die `Quantity` ist immer `1`, weil jeder Funktion nur ein Schlüsselwertspeicher zugeordnet werden kann.

Wenn der Befehl erfolgreich ausgeführt wurde, wird die Ausgabe folgendermaßen angezeigt.

```
ETag: ETVABCEXAMPLE
FunctionSummary:
  FunctionConfig:
    Comment: Max Age 2 years
    Runtime: cloudfront-js-2.0
    KeyValueStoreAssociations= \
      {Quantity=1, \
      Items=[{KeyValueStoreARN='arn:aws:cloudfront::111122223333:key-value-
store/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111'}]} \
  FunctionMetadata:
    CreatedTime: '2021-04-18T20:38:56.915000+00:00'
    FunctionARN: arn:aws:cloudfront::111122223333:function/MaxAge
    LastModifiedTime: '2023-11-19T20:38:56.915000+00:00'
    Stage: DEVELOPMENT
```

```
  Name: MaxAge
  Status: UNPUBLISHED
Location: https://cloudfront.amazonaws.com/2020-05-31/function/
arn:aws:cloudfront:::function/MaxAge
```

Die meisten Informationen werden aus der Anfrage wiederholt. Weitere Informationen werden hinzugefügt von CloudFront.

> **Hinweise**
>
> - `ETag` – Dieser Wert ändert sich jedes Mal, wenn Sie den Schlüsselwertspeicher ändern. Sie verwenden diesen Wert und den Funktionsnamen, um in Zukunft auf die Funktion zu verweisen. Stellen Sie sicher, dass Sie immer das aktuelle `ETag` verwenden.
> - `FunctionARN`— Der ARN für Ihre CloudFront Funktion.
> - 111122223333 — Das. AWS-Konto
> - `Stage` – die Phase der Funktion (`LIVE` oder `DEVELOPMENT`)
> - `Status` – der Status der Funktion (`PUBLISHED` oder `UNPUBLISHED`)

Nachdem Sie die Funktion erstellt haben, wird sie der Phase `DEVELOPMENT` hinzugefügt. Wir empfehlen Ihnen, Ihre Funktion zu testen, bevor Sie sie veröffentlichen. Nachdem Sie Ihre Funktion veröffentlicht haben, wechselt sie zur Phase `LIVE`.

> **TaggingConflict Fehler beim Erstellen von Funktionen**
>
> CloudFront Funktionen verwenden ARNs für Funktionen mit demselben Namen wieder. Beim Erstellen einer Funktion mit Tags kann ein Fehler `TaggingConflict` (HTTP 409) auftreten. Dies kann in den folgenden Szenarien auftreten:
>
> - Nach dem Löschen einer Funktion — Unmittelbar nach dem Löschen wird eine Funktion mit demselben Namen und denselben Tags neu erstellt.
> - Nach einem fehlgeschlagenen Erstellungsversuch — Wenn eine vorherige Erstellungsanforderung fehlgeschlagen ist (z. B. aufgrund ungültiger Parameter, einer ungültigen KVS-Zuordnung oder eines fehlgeschlagenen CloudFormation Stack-Rollbacks).

Wenn Sie eine `TaggingConflict` Fehlermeldung erhalten, können Sie:

- Warten Sie ein paar Minuten und wiederholen Sie den Vorgang.
- Verwenden Sie einen anderen Funktionsnamen. Dadurch wird der ARN-Wiederverwendungskonflikt vollständig vermieden.
- Wenn Ihre Vorlage KeyValueStore sowohl eine Funktion als auch eine Funktion erstellt, die darauf verweist, fügen Sie ein Explizit hinzu, `DependsOn` um sicherzustellen, dass das KVS vollständig bereitgestellt ist, bevor die Funktion erstellt wird.

Testfunktionen

Bevor Sie die Funktion in der Live-Phase (Produktion) bereitstellen, können Sie sie testen, um sicherzustellen, dass sie wie vorgesehen funktioniert. Um eine Funktion zu testen, geben Sie ein Ereignisobjekt an, das eine HTTP-Anfrage oder -Antwort darstellt, die Ihre CloudFront Distribution in der Produktion erhalten könnte.

CloudFront Functions macht Folgendes:

1. Führt die Funktion aus, wobei das bereitgestellte Ereignisobjekt als Eingabe verwendet wird.
2. Gibt das Ergebnis der Funktion (das geänderte Ereignisobjekt) zusammen mit allen Funktionsprotokollen oder Fehlermeldungen und der Rechenauslastung der Funktion zurück. Weitere Informationen zur Computing-Nutzung finden Sie unter the section called “Informationen zur Rechenauslastung”.

> **Note**
>
> Wenn Sie eine Funktion testen, überprüft sie CloudFront nur anhand von Fehlern bei der Funktionsausführung. CloudFrontüberprüft nicht, ob die Anfrage nach der Veröffentlichung erfolgreich bearbeitet wird. Wenn Ihre Funktion beispielsweise einen erforderlichen Header löscht, ist der Test erfolgreich, da kein Problem mit dem Code vorliegt. Wenn Sie die Funktion jedoch veröffentlichen und sie einer Verteilung zuordnen, schlägt die Funktion fehl, wenn eine Anfrage gestellt wurde CloudFront.

Inhalt

- Einrichten des Ereignisobjekts
- Testen der Funktion
- Informationen zur Rechenauslastung

Einrichten des Ereignisobjekts

Bevor Sie eine Funktion testen, müssen Sie das Ereignisobjekt einrichten, mit dem Sie sie testen können. Es gibt mehrere Möglichkeiten, dies zu tun.

Option 1: Einrichten eines Ereignisobjekts, ohne es zu speichern

Sie können ein Ereignisobjekt im Visual Editor in der CloudFront Konsole einrichten, ohne es zu speichern.

Sie können dieses Ereignisobjekt verwenden, um die Funktion von der CloudFront Konsole aus zu testen, auch wenn es nicht gespeichert ist.

Option 2: Erstellen eines Ereignisobjekts im visuellen Editor

Sie können ein Ereignisobjekt im Visual Editor in der CloudFront Konsole einrichten und es nicht speichern. Sie können für jede Funktion 10 Ereignisobjekte erstellen, so dass Sie beispielsweise verschiedene mögliche Eingaben testen können.

Wenn Sie das Ereignisobjekt auf diese Weise erstellen, können Sie das Ereignisobjekt verwenden, um die Funktion in der CloudFront Konsole zu testen. Sie können es nicht verwenden, um die Funktion mit einer AWS API oder einem SDK zu testen.

Option 3: Erstellen eines Ereignisobjekts mit einem Texteditor

Sie können einen Texteditor verwenden, um ein Ereignisobjekt im JSON-Format zu erstellen. Informationen zur Struktur eines Ereignisobjekts finden Sie unter Ereignisstruktur.

Sie können dieses Ereignisobjekt verwenden, um die Funktion mit der CLI zu testen. Sie können es jedoch nicht verwenden, um die Funktion in der CloudFront Konsole zu testen.

So erstellen Sie ein Ereignisobjekt (Option 1 oder 2)

1. Melden Sie sich bei der CloudFront Konsole unter an https://console.aws.amazon.com/cloudfront/v4/home#/functions und wählen Sie die Seite Funktionen aus.

Wählen Sie die Funktion, die Sie testen möchten.

2. Wählen Sie auf der Seite der Funktionsdetails die Registerkarte Test aus.
3. Wählen Sie für Ereignistyp eine der folgenden Optionen aus:

   - Wenn die Funktion eine HTTP-Anfrage ändert oder basierend auf der Anfrage eine Antwort generiert, wählen Sie Betrachteranfrage aus. Der Abschnitt Anforderung wird angezeigt.
   - Wählen Sie Viewer-Antwort. Die Abschnitte Anforderung und Antwort werden angezeigt.
4. Füllen Sie die Felder aus, die Sie in das Ereignis aufnehmen möchten. Sie können JSON bearbeiten wählen, um den unformatierten JSON-Code anzuzeigen.
5. (Optional) Um das Ereignis zu speichern, wählen Sie Speichern aus, geben im Feld Testereignis speichern einen Namen ein und wählen Sie dann Speichern aus.

   Sie können auch „JSON bearbeiten“ wählen und das unformatierte JSON kopieren und es in Ihrer eigenen Datei außerhalb von speichern CloudFront.

So erstellen Sie ein Ereignisobjekt (Option 3)

Erstellen Sie das Ereignisobjekt mit einem Texteditor. Speichern Sie die Datei in einem Verzeichnis, mit dem Ihr Computer eine Verbindung herstellen kann.

Vergewissern Sie sich, diese Richtlinien zu befolgen:

- Ignorieren Sie die Felder `distributionDomainName`, `distributionId` und `requestId`.
- Die Namen von Headern, Cookies und Abfragezeichenfolgen müssen in Kleinbuchstaben geschrieben sein.

Eine Möglichkeit, ein Ereignisobjekt auf diese Weise zu erstellen, besteht darin, mit dem visuellen Editor ein Beispiel zu erstellen. Sie können sicher sein, dass das Beispiel korrekt formatiert ist. Sie können dann den unformatierten JSON-Code kopieren, in einen Text-Editor einfügen und die Datei speichern.

Weitere Informationen zur Struktur eines Ereignisses finden Sie unter Ereignisstruktur.

Testen der Funktion

Sie können eine Funktion in der CloudFront Konsole oder mit der AWS Command Line Interface (AWS CLI) testen.

Console

So testen Sie die -Funktion

1. Melden Sie sich bei der CloudFront Konsole unter an https://console.aws.amazon.com/cloudfront/v4/home#/functions und wählen Sie die Seite Funktionen aus.
2. Wählen Sie die Funktion, die Sie testen möchten.
3. Wählen Sie die Registerkarte Test.
4. Stellen Sie sicher, dass das korrekte Ereignis angezeigt wird. Um vom aktuell angezeigten Ereignis zu einem anderen zu wechseln, wählen Sie im Feld Testereignis auswählen ein anderes Ereignis aus.
5. Wählen Sie Testfunktion aus. Die Konsole zeigt die Ausgabe der Funktion, einschließlich der Funktionsprotokolle und Rechenauslastung.

CLI

Sie können eine Funktion mit dem Befehl aws cloudfront test-function testen.

So testen Sie die -Funktion

1. Öffnen Sie ein Befehlszeilenfenster.
2. Führen Sie den folgenden Befehl über dasselbe Verzeichnis aus, das die angegebene Datei enthält.

   In diesem Beispiel wird die Notation `fileb://` verwendet, um die Ereignisobjektdatei zu übergeben. Es sind Zeilenumbrüche enthalten, um den Befehl lesbarer zu machen.

   ```
   aws cloudfront test-function \
       --name MaxAge \
       --if-match ETVABCEXAMPLE \
       --event-object fileb://event-maxage-test01.json \
       --stage DEVELOPMENT
   ```

   Hinweise

   - Sie verweisen mit ihrem Namen und ETag (im `if-match` Parameter) auf die Funktion. Sie verweisen auf das Ereignisobjekt anhand seines Speicherorts in Ihrem Dateisystem.

- Die Phase kann `DEVELOPMENT` oder `LIVE` sein.

Wenn der Befehl erfolgreich ausgeführt wurde, wird die Ausgabe folgendermaßen angezeigt.

```
TestResult:
  ComputeUtilization: '21'
  FunctionErrorMessage: ''
  FunctionExecutionLogs: []
  FunctionOutput: '{"response":{"headers":{"cloudfront-functions":
{"value":"generated-by-CloudFront-Functions"},"location":{"value":"https://
aws.amazon.com/cloudfront/"}},"statusDescription":"Found","cookies":
{},"statusCode":302}}'
  FunctionSummary:
    FunctionConfig:
      Comment: MaxAge function
      Runtime: cloudfront-js-2.0
      KeyValueStoreAssociations= \
      {Quantity=1, \
      Items=[{KeyValueStoreARN='arn:aws:cloudfront::111122223333:key-value-
store/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111'}]} \
    FunctionMetadata:
      CreatedTime: '2021-04-18T20:38:56.915000+00:00'
      FunctionARN: arn:aws:cloudfront::111122223333:function/MaxAge
      LastModifiedTime: '2023-17-20T10:38:57.057000+00:00'
      Stage: DEVELOPMENT
    Name: MaxAge
    Status: UNPUBLISHED
```

Hinweise

- `FunctionExecutionLogs` enthält eine Liste von Protokollzeilen, die die Funktion in `console.log()`-Anweisungen geschrieben hat (falls vorhanden).
- `ComputeUtilization` enthält Informationen zur Ausführung Ihrer Funktion. Siehe the section called “Informationen zur Rechenauslastung”.
- `FunctionOutput` enthält das von der Funktion zurückgegebene Ereignisobjekt.

Informationen zur Rechenauslastung

Rechenauslastung ist die Zeit, die die Ausführung der Funktion in Anspruch genommen hat, als Prozentsatz der maximal zulässigen Zeit. Zum Beispiel bedeutet ein Wert von 35, dass die Funktion in 35 % der maximal zulässigen Zeit abgeschlossen wurde.

Wenn eine Funktion kontinuierlich die maximal zulässige Zeit überschreitet, wird die Funktion CloudFront gedrosselt. Die folgende Liste zeigt, mit welcher Wahrscheinlichkeit eine Funktion basierend auf dem Wert der Rechenauslastung gedrosselt wird.

Rechenauslastungswert:

- 1 – 50 – Die Funktion liegt deutlich unter der maximal zulässigen Zeit und sollte ohne Drosselung ausgeführt werden.
- 51 – 70 – Die Funktion nähert sich der maximal zulässigen Zeit. Erwägen Sie die Optimierung des Funktionscodes.
- 71 — 100 — Die Funktion kommt der maximal zulässigen Zeit sehr nahe oder überschreitet sie. CloudFront wird diese Funktion wahrscheinlich drosseln, wenn Sie sie einer Verteilung zuordnen.

Aktualisieren von Funktionen

Sie können eine Funktion jederzeit aktualisieren. Die Änderungen werden nur an der Version der Funktion vorgenommen, die sich in der `DEVELOPMENT`-Phase befindet. Um die Updates von der `DEVELOPMENT`-Phase in die `LIVE`-Phase zu kopieren, müssen Sie die Funktion veröffentlichen.

Sie können den Code einer Funktion in der CloudFront Konsole oder mit der AWS Command Line Interface (AWS CLI) aktualisieren.

Console

So aktualisieren Sie den Funktionscode

1. Melden Sie sich bei der CloudFront Konsole unter an https://console.aws.amazon.com/cloudfront/v4/home#/functions und wählen Sie die Seite Funktionen aus.

   Wählen Sie die zu aktualisierende Funktion aus.
2. Wählen Sie Bearbeiten aus und nehmen Sie die folgenden Änderungen vor.

   - Aktualisieren Sie die entsprechenden Felder im Abschnitt Details.

- Ändern oder entfernen Sie den zugeordneten Schlüsselwertspeicher. Weitere Hinweise zu Schlüsselwertspeichern finden Sie unter the section called “ CloudFront KeyValueStore”.

- Ändern Sie den Funktionscode. Wählen Sie die Registerkarte Erstellen aus, nehmen Sie die entsprechenden Änderungen vor und wählen Sie dann Änderungen speichern aus, um die Änderungen am Code zu speichern.

CLI

So aktualisieren Sie den Funktionscode:

1. Öffnen Sie ein Befehlszeilenfenster.
2. Führen Sie den folgenden Befehl aus.

   In diesem Beispiel wird die Notation `fileb://` verwendet, um die Datei zu übergeben. Es sind Zeilenumbrüche enthalten, um den Befehl lesbarer zu machen.

```
aws cloudfront update-function \
    --name MaxAge \
    --function-config '{"Comment":"Max Age 2 years","Runtime":"cloudfront-
js-2.0","KeyValueStoreAssociations":{"Quantity":1,"Items":
[{"KeyValueStoreARN":"arn:aws:cloudfront::111122223333:key-value-store/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"}]}}' \
    --function-code fileb://function-max-age-v1.js \
    --if-match ETVABCEXAMPLE
```

   Hinweise

   - Sie können die Funktion sowohl anhand ihres Namens als auch ETag (im `if-match` Parameter) identifizieren. Stellen Sie sicher, dass Sie den aktuellen verwenden ETag. Sie können diesen Wert aus der DescribeFunctionAPI-Operation abrufen.
   - Sie müssen den `function-code` angeben, auch wenn Sie ihn nicht ändern möchten.

- Seien Sie vorsichtig mit der `function-config`. Sie sollten alles übergeben, was Sie in der Konfiguration beibehalten möchten. Gehen Sie insbesondere mit dem Schlüsselwertspeicher folgendermaßen vor:
  - Um die bestehende Schlüsselwertspeicher-Zuordnung (falls vorhanden) beizubehalten, geben Sie den Namen des vorhandenen Speichers an.
  - Um die Zuordnung zu ändern, geben Sie den Namen des neuen Schlüsselwertspeichers an.
  - Um die Zuordnung zu entfernen, lassen Sie den `KeyValueStoreAssociations`-Parameter weg.

Wenn der Befehl erfolgreich ausgeführt wurde, wird die Ausgabe folgendermaßen angezeigt.

```
ETag: ETVXYZEXAMPLE
FunctionSummary:
  FunctionConfig:
    Comment: Max Age 2 years \
    Runtime: cloudfront-js-2.0 \
    KeyValueStoreAssociations= \
      {Quantity=1, \
      Items=[{KeyValueStoreARN='arn:aws:cloudfront::111122223333:key-value-
store/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111'}]} \
  FunctionMetadata: \
    CreatedTime: '2021-04-18T20:38:56.915000+00:00' \
    FunctionARN: arn:aws:cloudfront::111122223333:function/MaxAge \
    LastModifiedTime: '2023-12-19T23:41:15.389000+00:00' \
    Stage: DEVELOPMENT \
  Name: MaxAge \
  Status: UNPUBLISHED
```

Die meisten Informationen werden aus der Anfrage wiederholt. Weitere Informationen werden von hinzugefügt CloudFront.

Hinweise

- `ETag` – Dieser Wert ändert sich jedes Mal, wenn Sie den Schlüsselwertspeicher ändern.
- `FunctionARN`— Der ARN für Ihre CloudFront Funktion.

- `Stage` – die Phase der Funktion (`LIVE` oder `DEVELOPMENT`)
- `Status` – der Status der Funktion (`PUBLISHED` oder `UNPUBLISHED`)

Veröffentlichen von Funktionen

Wenn Sie Ihre Funktion veröffentlichen, wird die Funktion von der `DEVELOPMENT`-Phase in die `LIVE`-Phase kopiert.

Wenn der Funktion kein Cacheverhalten zugeordnet ist, können Sie sie beim Veröffentlichen mit einem Cacheverhalten verknüpfen. Sie können Cache-Verhalten nur Funktionen zuordnen, die sich in der `LIVE`-Phase befinden.

Important

- Wir empfehlen, die Funktion vor dem Veröffentlichen zu testen.
- Nachdem Sie die Funktion veröffentlicht haben, verwenden alle Cacheverhaltensweisen, die mit der Funktion verknüpft sind, automatisch die neu veröffentlichte Kopie, sobald die Bereitstellung der Distributionen abgeschlossen ist.

Sie können eine Funktion in der CloudFront-Konsole oder mit der veröffentlichen AWS CLI.

Console

So veröffentlichen Sie eine Funktion

1. Melden Sie sich auf https://console.aws.amazon.com/cloudfront/v4/home#/functions bei der CloudFront-Konsole an und öffnen Sie die Seite Funktionen.
2. Wählen Sie die zu aktualisierende Funktion aus.
3. Wählen Sie die Registerkarte Veröffentlichen und dann die Option Veröffentlichen aus. Wenn Ihre Funktion bereits an ein oder mehrere Cacheverhaltensweisen angehängt ist, wählen Sie Veröffentlichen und aktualisieren.
4. (Optional) Um die Verteilungen anzuzeigen, die mit der Funktion verknüpft sind, wählen Sie Associated CloudFront distributions, um diesen Abschnitt zu erweitern.

Wenn dies gelingt, wird oben auf der Seite ein Banner mit dem Hinweis ***Funktionsname*** erfolgreich veröffentlicht angezeigt. Sie können auch den Tab Build und dann Live auswählen, um die Live-Version des Funktionscodes anzuzeigen.

CLI

So veröffentlichen Sie eine Funktion

1. Öffnen Sie ein Befehlszeilenfenster.
2. Führen Sie den Befehl aws cloudfront publish-function aus. Im Beispiel werden Zeilenumbrüche bereitgestellt, um das Beispiel besser lesbar zu machen.

```
aws cloudfront publish-function \
    --name MaxAge \
    --if-match ETVXYZEXAMPLE
```

Wenn der Befehl erfolgreich ausgeführt wurde, wird die Ausgabe folgendermaßen angezeigt.

```
FunctionSummary:
  FunctionConfig:
    Comment: Max Age 2 years
    Runtime: cloudfront-js-2.0
  FunctionMetadata:
    CreatedTime: '2021-04-18T21:24:21.314000+00:00'
    FunctionARN: arn:aws:cloudfront::111122223333:function/ExampleFunction
    LastModifiedTime: '2023-12-19T23:41:15.389000+00:00'
    Stage: LIVE
  Name: MaxAge
  Status: UNASSOCIATED
```

Verknüpfen von Funktionen mit Distributionen

Um eine Funktion mit einer Distribution zu verwenden, verknüpfen Sie die Funktion einem oder mehreren Cacheverhalten in der Distribution. Sie können eine Funktion mit mehreren Cache-Verhaltensweisen in mehreren Verteilungen verknüpfen.

Sie können eine Funktion den folgenden Verhaltensweisen zuordnen:

- einem vorhandenen Cacheverhalten
- einem neuen Cacheverhalten in einer vorhandenen Distribution

- einem neuen Cacheverhalten in einer neuen Distribution

Wenn Sie eine Funktion mit einem Cache-Verhalten verknüpfen, müssen Sie einen Ereignistyp auswählen. Der Ereignistyp bestimmt, wann CloudFront die Funktion ausführt.

Sie können aus den folgenden Ereignistypen auswählen:

- Betrachteranfrage – Die Funktion wird ausgeführt, wenn CloudFront eine Anfrage von einem Betrachter erhält.
- Betrachterantwort – Die Funktion wird ausgeführt, bevor CloudFront eine Antwort an den Betrachter zurückgibt.

Sie können keine Ursprungsereignistypen (Ursprungsanforderung und Ursprungsantwort) mit CloudFront Functions verwenden. Stattdessen können Sie Lambda@Edge verwenden. Weitere Informationen finden Sie unter CloudFront Ereignisse, die eine Lambda @Edge -Funktion auslösen können.

> **Note**
>
> Bevor Sie eine Funktion verknüpfen, müssen Sie sie in der `LIVE`-Phase veröffentlichen.

Sie können eine Funktion mit einer Distribution in der CloudFront-Konsole oder mit der AWS Command Line Interface (AWS CLI) verknüpfen. Das folgende Verfahren zeigt, wie eine Funktion mit einem vorhandenen Cache-Verhalten verknüpft wird.

Console

So verknüpfen Sie eine Funktion mit einem vorhandenen Cacheverhalten

1. Melden Sie sich auf https://console.aws.amazon.com/cloudfront/v4/home#/functions bei der CloudFront-Konsole an und öffnen Sie die Seite Funktionen.
2. Wählen Sie die Funktion aus, die Sie verknüpfen möchten.
3. Wählen Sie auf der Seite Funktion die Registerkarte Veröffentlichen aus.
4. Wählen Sie die Funktion „Veröffentlichen“ aus.
5. Wählen Sie Add association. Wählen Sie im daraufhin angezeigten Dialogfeld eine Distribution, einen Ereignistyp und/oder ein Cacheverhalten aus.

Wählen Sie für den Ereignistyp aus, wann diese Funktion ausgeführt werden soll:

- Viewer-Anforderung – Die Funktion wird jedes Mal ausgeführt, wenn CloudFront eine Anforderung erhält.
- Viewer-Antwort – Die Funktion wird jedes Mal ausgeführt, wenn CloudFront eine Antwort zurückgibt.

6. Um die Konfiguration zu speichern, wählen Sie Zuordnung hinzufügen.

CloudFront ordnet die Verteilung der Funktion zu. Warten Sie ein paar Minuten, bis die zugehörige Verteilung die Bereitstellung abgeschlossen hat. Sie können auf der Seite mit den Funktionsdetails die Option Distribution anzeigen auswählen, um den Fortschritt zu überprüfen.

CLI

So verknüpfen Sie eine Funktion mit einem vorhandenen Cacheverhalten

1. Öffnen Sie ein Befehlszeilenfenster.
2. Geben Sie den folgenden Befehl ein, um die Distributionskonfiguration für die Distribution zu speichern, deren Cacheverhalten Sie einer Funktion zuordnen möchten. Dieser Befehl speichert die Verteilungskonfiguration in einer Datei mit dem Namen `dist-config.yaml`. Um diesen Befehl zu verwenden, gehen Sie folgendermaßen vor:

   - Ersetzen Sie *`DistributionID`* durch die ID der Verteilung.
   - Führen Sie den Befehl in einer Zeile aus. Im Beispiel werden Zeilenumbrüche bereitgestellt, um das Beispiel besser lesbar zu machen.

```
aws cloudfront get-distribution-config \
    --id DistributionID \
    --output yaml > dist-config.yaml
```

   Wenn der Befehl erfolgreich ist, gibt die AWS CLI keine Ausgabe zurück.

3. Öffnen Sie die Datei mit dem Namen `dist-config.yaml`, die Sie erstellt haben. Bearbeiten Sie die Datei, indem Sie die folgenden Änderungen vornehmen.

   a. Benennen Sie das `ETag`-Feld in `IfMatch` um, ändern Sie jedoch nicht den Wert des Feldes.

b. Suchen Sie im Cache-Verhalten das Objekt mit dem Namen `FunctionAssociations`. Aktualisieren Sie dieses Objekt, um eine Funktionszuordnung hinzuzufügen. Die YAML-Syntax für eine Funktionszuordnung sieht wie im folgenden Beispiel aus.

   - Das folgende Beispiel zeigt den Ereignistyp Betrachteranfrage (Auslöser). Um den Ereignistyp Betrachterantwort zu verwenden, ersetzen Sie `viewer-request` durch `viewer-response`.
   - Ersetzen Sie *`arn:aws:cloudfront::111122223333:function/ExampleFunction`* durch den Amazon-Ressourcennamen (ARN) der Funktion, die Sie diesem Cache-Verhalten zuordnen. Um die Funktion ARN zu erhalten, können Sie den Befehl aws cloudfront list-functions verwenden.

```
FunctionAssociations:
  Items:
    - EventType: viewer-request
      FunctionARN: arn:aws:cloudfront::111122223333:function/ExampleFunction
  Quantity: 1
```

c. Nachdem Sie diese Änderungen vorgenommen haben, speichern Sie die Datei.

4. Verwenden Sie den folgenden Befehl, um die Verteilung zu aktualisieren und die Funktionszuordnung hinzuzufügen. Um diesen Befehl zu verwenden, gehen Sie folgendermaßen vor:

   - Ersetzen Sie *`DistributionID`* durch die ID der Verteilung.
   - Führen Sie den Befehl in einer Zeile aus. Im Beispiel werden Zeilenumbrüche bereitgestellt, um das Beispiel besser lesbar zu machen.

```
aws cloudfront update-distribution \
    --id DistributionID \
    --cli-input-yaml file://dist-config.yaml
```

Wenn der Befehl erfolgreich ist, sehen Sie eine Ausgabe wie die folgende, die die Verteilung beschreibt, die gerade mit der Funktionszuordnung aktualisiert wurde. Die folgende Beispielausgabe wird zur besseren Lesbarkeit abgeschnitten.

```
Distribution:
  ARN: arn:aws:cloudfront::111122223333:distribution/EBEDLT3BGRBBW
```

```
  ... truncated ...
  DistributionConfig:
    ... truncated ...
    DefaultCacheBehavior:
      ... truncated ...
      FunctionAssociations:
        Items:
        - EventType: viewer-request
          FunctionARN: arn:aws:cloudfront::111122223333:function/ExampleFunction
        Quantity: 1
      ... truncated ...
  DomainName: d111111abcdef8.cloudfront.net
  Id: EDFDVBD6EXAMPLE
  LastModifiedTime: '2021-04-19T22:39:09.158000+00:00'
  Status: InProgress
ETag: E2VJGGQEG1JT8S
```

Der `Status` der Verteilung ändert sich in `InProgress` während der erneuten Bereitstellung der Verteilung. Wenn die neue Distributionskonfiguration einen CloudFront-Edge-Standort erreicht, beginnt dieser Edge-Standort mit der Verwendung der zugehörigen Funktion. Wenn die Distribution vollständig bereitgestellt ist, ändert sich der `Status` wieder in `Deployed`. Dies weist darauf hin, dass die zugehörige CloudFront-Funktion an allen CloudFront-Edge-Standorten weltweit aktiv ist. Dies dauert in der Regel einige Minuten.

Amazon CloudFront KeyValueStore

CloudFront KeyValueStore ist ein sicherer, globaler Schlüsselwert-Datenspeicher mit niedriger Latenz, der Lesezugriff von CloudFront Functions aus ermöglicht und so eine erweiterte, anpassbare Logik an den Edge-Standorten ermöglicht. CloudFront

Mit CloudFront KeyValueStore können Sie unabhängig voneinander Aktualisierungen am Funktionscode und an den mit einer Funktion verknüpften Daten vornehmen. Diese Trennung vereinfacht den Funktionscode und erleichtert die Aktualisierung von Daten, ohne dass Codeänderungen vorgenommen werden müssen.

> **Note**
>
> Um diese CloudFront Funktion verwenden zu können CloudFront KeyValueStore, muss sie JavaScript Runtime 2.0 verwenden.

Dies ist das allgemeine Verfahren für die Verwendung von Schlüssel-Wert-Paaren:

- Erstellen Sie Schlüsselwertspeicher und füllen Sie sie mit einer Reihe von Schlüssel-Wert-Paaren. Sie können Ihre Schlüsselwertspeicher zu einem Amazon-S3-Bucket hinzufügen oder sie manuell eingeben.
- Ordnen Sie die Schlüsselwertspeicher Ihrer CloudFront Funktion zu.
- Verwenden Sie in Ihrem Funktionscode den Namen des Schlüssels, um entweder den mit dem Schlüssel verknüpften Wert abzurufen oder um zu bestimmen, ob ein Schlüssel existiert. Weitere Informationen zur Verwendung von Schlüssel-Wert-Paaren im Funktionscode und zu Hilfsmethoden finden Sie unter the section called “Hilfsmethoden für Schlüsselwertspeicher”.

Anwendungsfälle

Sie können Schlüssel-Wert-Paare für die folgenden Beispiele verwenden:

- URL-Umschreibungen oder Weiterleitungen — Das Schlüssel-Wert-Paar kann die umgeschriebene URL oder die Weiterleitung URLs enthalten. URLs
- A/B-Tests und Feature-Flags – Sie können eine Funktion zur Durchführung von Experimenten erstellen, indem Sie einer bestimmten Version Ihrer Website einen Prozentsatz des Datenverkehrs zuweisen.
- Zugriffsberechtigung – Sie können eine Zugriffskontrolle implementieren, um Anforderungen auf der Grundlage von Kriterien zuzulassen oder abzulehnen, die von Ihnen und den in einem Schlüsselwertspeicher gespeicherten Daten definiert wurden.

Unterstützte Formate für Werte

Der Wert in einem Schlüssel-Wert-Paar kann in einem der folgenden Formate gespeichert werden:

- Zeichenfolge
- bytekodierte Zeichenfolge
- JSON

Sicherheit

Die CloudFront Funktion und all ihre Schlüsselwertspeicher-Daten werden wie folgt sicher behandelt:

- CloudFront verschlüsselt alle Schlüsselwertspeicher im Ruhezustand und während der Übertragung (beim Lesen oder Schreiben in die Schlüsselwertspeicher), wenn Sie die CloudFront KeyValueStoreAPI-Operationen aufrufen.
- Wenn die Funktion ausgeführt wird, CloudFront entschlüsselt jedes Schlüssel-Wert-Paar im Speicher an den Edge-Standorten. CloudFront

Informationen zu den ersten Schritten finden Sie in den folgenden Themen. CloudFront KeyValueStore

Themen

- Arbeiten mit einem Schlüsselwertspeicher
- Arbeiten mit Schlüsselwertdaten

- Weitere Informationen zu den ersten Schritten finden Sie im CloudFront KeyValueStore AWS Blogbeitrag Introducing Amazon. CloudFront KeyValueStore

Arbeiten mit einem Schlüsselwertspeicher

Sie müssen einen Schlüsselwertspeicher für die Schlüssel-Wert-Paare erstellen, die Sie in CloudFront Functions verwenden möchten.

Nachdem Sie die Schlüsselwertspeicher erstellt und Schlüssel-Wert-Paare hinzugefügt haben, können Sie die Schlüsselwerte in Ihrem Funktionscode verwenden. CloudFront

Die ersten Schritte werden in den folgenden Themen behandelt:

Topics

- Erstellen eines Schlüsselwertspeichers
- Zuordnen eines Schlüsselwertspeichers zu einer Funktion
- Aktualisieren eines Schlüsselwertspeichers
- Abrufen eines Verweises auf einen Schlüsselwertspeicher
- Löschen eines Schlüsselwertspeichers
- Dateiformat für Schlüssel-Wert-Paare

> **Note**
>
> Die JavaScript Runtime 2.0 enthält einige Hilfsmethoden für die Arbeit mit Schlüsselwerten im Funktionscode. Weitere Informationen finden Sie unter the section called “Hilfsmethoden für Schlüsselwertspeicher”.

Erstellen eines Schlüsselwertspeichers

Sie können einen Schlüsselwertspeicher und seine Schlüssel-Wert-Paare gleichzeitig erstellen. Sie können auch einen leeren Schlüsselwertspeicher erstellen und später die Schlüssel-Wert-Paare hinzufügen.

> **Note**
>
> Wenn Sie einen Amazon-S3-Bucket als Ihre Datenquelle angeben, benötigen Sie die `s3:GetBucketLocation`- und `s3:GetObject`-Berechtigungen für diesen Bucket. Wenn Sie nicht über diese Berechtigungen verfügen, CloudFront können Sie Ihren Key Value Store nicht erfolgreich erstellen.

Entscheiden Sie, ob Sie Schlüssel-Wert-Paare zeitgleich zur Erstellung des Schlüsselwertspeichers hinzufügen möchten. Sie können Ihre Schlüssel-Wert-Paare mithilfe der CloudFront Konsole, der CloudFront API oder importieren. AWS SDKs Sie können Ihre Datei mit Schlüssel-Wert-Paaren jedoch nur dann importieren, wenn Sie den Schlüsselwertspeicher neu erstellen.

Informationen zum Erstellen einer Datei mit Schlüssel-Wert-Paaren finden Sie unter Dateiformat für Schlüssel-Wert-Paare.

Console

So erstellen Sie einen Schlüsselwertspeicher

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Seite Funktionen in der CloudFront Konsole unter. https://console.aws.amazon.com/cloudfront/v4/home#/functions
2. Wählen Sie die KeyValueStoresRegisterkarte und dann Erstellen aus KeyValueStore.
3. Geben Sie für den Schlüsselwertspeicher einen Namen und optional eine Beschreibung ein.

4. Vollständige S3-URI:

   - Wenn Sie über eine Datei mit Schlüssel-Wert-Paaren verfügen, geben Sie den Pfad zum Amazon-S3-Bucket ein, in dem Sie die Datei gespeichert haben.
   - Lassen Sie dieses Feld leer, wenn Sie die Schlüssel-Wert-Paare manuell eingeben möchten.

5. Wählen Sie Erstellen aus. Der Schlüsselwertspeicher ist jetzt vorhanden.

   Die Seite mit den Details für den neuen Schlüsselwertspeicher wird angezeigt. Die Informationen auf der Seite umfassen die ID und den ARN des Schlüsselwertspeichers.

   - Die ID ist eine zufällige Zeichenfolge, die für Sie einzigartig ist AWS-Konto.
   - Der ARN hat folgende Syntax:

     ```
     AWS-Konto:key-value-store/the key value stores ID
     ```

6. Sehen Sie sich den Abschnitt Schlüssel-Wert-Paare an. Wenn Sie eine Datei importiert haben, werden in diesem Abschnitt einige Schlüssel-Wert-Paare angezeigt. Sie haben die folgenden Möglichkeiten:

   - Wenn Sie eine Datei importiert haben, können Sie weitere Werte manuell hinzufügen.
   - Wenn Sie keine Datei aus einem Amazon-S3-Bucket importiert haben und jetzt Schlüssel-Wert-Paare hinzufügen möchten, können Sie den nächsten Schritt ausführen.
   - Sie können diesen Schritt überspringen und die Schlüssel-Wert-Paare später hinzufügen.

7. So fügen Sie die Paare jetzt hinzu:

   a. Wählen Sie Schlüssel-Wert-Paare hinzufügen aus.

   b. Wählen Sie Paar hinzufügen und geben Sie einen Namen und einen Wert ein. Wiederholen Sie diesen Schritt, um weitere Paare hinzuzufügen.

   c. Wenn Sie fertig sind, wählen Sie Änderungen speichern, um alle Schlüssel-Wert-Paare im Schlüsselwertspeicher zu speichern. Wählen Sie im angezeigten Dialogfeld Ja aus.

8. Wenn Sie den Schlüsselwertspeicher jetzt einer Funktion zuordnen möchten, füllen Sie den Abschnitt Zugeordnete Funktionen aus. Für weitere Informationen siehe ??? oder ???.

   Sie können die Funktion auch später zuordnen und zwar entweder auf der Detailseite des Schlüsselwertspeichers oder auf der Detailseite der Funktionen.

AWS CLI

So erstellen Sie einen Schlüsselwertspeicher

- Führen Sie den folgenden Befehl aus, um einen Schlüsselwertspeicher zu erstellen und die Schlüssel-Wert-Paare aus einem Amazon-S3-Bucket zu importieren.

```
aws cloudfront create-key-value-store \
    --name=keyvaluestore1 \
    --comment="This is my key value store file" \
    --import-source=SourceType=S3,SourceARN=arn:aws:s3:::amzn-s3-demo-
bucket1/kvs-input.json
```

Antwort

```
{
    "ETag": "ETVABCEXAMPLE",
    "Location": "https://cloudfront.amazonaws.com/2020-05-31/key-value-store/
arn:aws:cloudfront::123456789012:key-value-store/8aa76c93-3198-462c-aaf6-
example",
    "KeyValueStore": {
        "Name": "keyvaluestore1",
        "Id": "8aa76c93-3198-462c-aaf6-example",
        "Comment": "This is my key value store file",
        "ARN": "arn:aws:cloudfront::123456789012:key-value-
store/8aa76c93-3198-462c-aaf6-example",
        "Status": "PROVISIONING",
        "LastModifiedTime": "2024-08-06T22:19:10.813000+00:00"
    }
}
```

API

So erstellen Sie einen Schlüsselwertspeicher

1. Verwenden Sie die CloudFrontCreateKeyValueStoreOperation. Für diese Operation sind mehrere Parameter erforderlich:
   - ein `name` des Schlüsselwertspeichers.
   - Ein `comment`-Parameter mit einem Kommentar.

- ein `import-source`-Parameter, mit dem Sie Schlüssel-Wert-Paare aus einer Datei importieren können, die in einem Amazon-S3-Bucket gespeichert ist. Der Import aus einer Datei ist nur beim Erstellen des Schlüsselwertspeichers möglich. Weitere Informationen zur Dateistruktur finden Sie unter the section called “Dateiformat für Schlüssel-Wert-Paare”.

Die Operationsantwort enthält die folgenden Informationen:

- Die in der Anforderung übergebenen Werte, einschließlich des von Ihnen zugewiesenen Namens.
- Daten wie die Erstellungszeit.
- ein `ETag` (z. B. `ETVABCEXAMPLE`), der ARN mit dem Namen des Schlüsselwertspeichers (z. B. `arn:aws:cloudfront::123456789012:key-value-store/keyvaluestore1`).

  Sie verwenden eine Kombination aus `ETag`, ARN und dem Namen, um programmgesteuert mit dem Schlüsselwertspeicher zu arbeiten.

Statuswerte des Schlüsselwertspeichers

Wenn Sie einen Schlüsselwertspeicher erstellen, kann der Datenspeicher die folgenden Statuswerte haben.

Wert	Description
Bereitstellung	Der Schlüsselwertspeicher wurde erstellt und verarbeitet CloudFront derzeit die von Ihnen angegebene Datenquelle.
Bereit	Der Schlüsselwertspeicher wurde erstellt und die von Ihnen angegebene Datenquelle wurde CloudFront erfolgreich verarbeitet.
Import fehlgeschlagen	CloudFront konnte die von Ihnen angegebene Datenquelle nicht verarbeiten. Dieser Status kann angezeigt werden, wenn Ihr Dateiformat nicht gültig ist oder wenn die Datei die Größenbeschränkung überschreitet. Weitere Informationen finden Sie unter Dateiformat für Schlüssel-Wert-Paare.

Zuordnen eines Schlüsselwertspeichers zu einer Funktion

Nachdem Sie Ihren Schlüsselwertspeicher erstellt haben, können Sie Ihre Funktion aktualisieren, um sie Ihrem Schlüsselwertspeicher zuzuordnen. Sie müssen diese Zuordnung vornehmen, um die Schlüssel-Wert-Paare aus diesem Speicher in dieser Funktion verwenden zu können. Die folgenden Regeln gelten:

- Eine Funktion kann nur einen Schlüsselwertspeicher haben.
- Ein Schlüsselwertspeicher kann mehreren Funktionen zugeordnet werden.

Console

So ordnen Sie einen Schlüsselwertspeicher einer Funktion zu

1. Melden Sie sich bei der CloudFront Konsole unter an https://console.aws.amazon.com/cloudfront/v4/home#/functions und wählen Sie die Seite Funktionen aus.
2. Wählen Sie den Funktionsnamen aus.
3. Gehen Sie zum KeyValueStore Bereich Associate und wählen Sie Associate existing aus KeyValueStore.
4. Wählen Sie den Schlüsselwertspeicher aus, der die Schlüssel-Wert-Paare in der Funktion enthält, und wählen Sie dann Assoziieren aus. KeyValueStore

   CloudFront ordnet den Speicher sofort der Funktion zu. Sie müssen die Funktion nicht speichern.
5. Um einen anderen Schlüsselwertspeicher anzugeben, wählen Sie Zugeordnet aktualisieren KeyValueStore, wählen Sie einen anderen Namen für den Schlüsselwertspeicher aus und klicken Sie dann auf Zuordnen KeyValueStore.

Weitere Informationen finden Sie unter the section called “Aktualisieren von Funktionen”.

AWS CLI

So ordnen Sie einen Schlüsselwertspeicher einer Funktion zu

- Führen Sie den folgenden Befehl aus, um die *MaxAge*-Funktion zu aktualisieren und eine Schlüsselwertspeicherressource zuzuordnen.

```
aws cloudfront update-function \
```

```
    --name MaxAge \
    --function-config '{"Comment":"Max Age 2 years","Runtime":"cloudfront-
js-2.0","KeyValueStoreAssociations":{"Quantity":1,"Items":
[{"KeyValueStoreARN":"arn:aws:cloudfront::123456789012:key-value-
store/8aa76c93-3198-462c-aaf6-example"}]}}' \
    --function-code fileb://function-max-age-v1.js \
    --if-match ETVABCEXAMPLE
```

- Um einen Schlüsselwertspeicher einer Funktion zuzuordnen, geben Sie den Parameter `KeyValueStoreAssociations` und den ARN des Schlüsselwertspeichers an.
- Um die Zuordnung zu ändern, geben Sie den ARN eines anderen Schlüsselwertspeichers an.
- Um die Zuordnung zu entfernen, entfernen Sie den Parameter `KeyValueStoreAssociations`.

Weitere Informationen finden Sie unter the section called “Aktualisieren von Funktionen”.

API

So ordnen Sie einen Schlüsselwertspeicher einer Funktion zu

- Verwenden Sie die API-Operation UpdateFunction. Weitere Informationen finden Sie unter the section called “Aktualisieren von Funktionen”.

Hinweise

- Wenn Sie einen Schlüsselwertspeicher bearbeiten, ohne die Schlüssel-Wert-Paare zu ändern, oder wenn Sie nur die Schlüssel-Wert-Paare im Schlüsselwertspeicher bearbeiten, müssen Sie den Schlüsselwertspeicher nicht erneut zuordnen. Sie müssen die Funktion auch nicht erneut veröffentlichen.

  Wir empfehlen jedoch, die Funktion zu testen, um zu überprüfen, ob sie erwartungsgemäß funktioniert. Weitere Informationen finden Sie unter Testfunktionen.
- Sie können alle Funktionen anzeigen, die bestimmte Schlüsselwertspeicher verwenden. Wählen Sie in der CloudFront Konsole die Detailseite des Key-Value-Speichers aus.

Aktualisieren eines Schlüsselwertspeichers

Wenn Sie einen Schlüsselwertspeicher aktualisieren, können Sie die Schlüssel-Wert-Paare ändern oder die Zuordnung zwischen dem Schlüsselwertspeicher und der Funktion ändern.

Console

So aktualisieren Sie einen Schlüsselwertspeicher

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Seite Funktionen in der CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home#/functions.
2. Wählen Sie die Registerkarte KeyValueStores aus.
3. Wählen Sie den Schlüsselwertspeicher aus, den Sie aktualisieren möchten.
   - Um die Schlüssel-Wert-Paare zu aktualisieren, wählen Sie Bearbeiten im Abschnitt Schlüssel-Wert-Paare aus. Sie können beliebige Schlüssel-Wert-Paare hinzufügen oder löschen. Sie können auch den Wert eines vorhandenen Schlüssel-Wert-Paares ändern. Wenn Sie fertig sind, wählen Sie Änderungen speichern aus.
   - Um die Zuordnung dieses Schlüsselwertspeichers zu aktualisieren, wählen Sie Gehe zu Funktionen aus. Weitere Informationen finden Sie unter the section called “Zuordnen eines Schlüsselwertspeichers zu einer Funktion”.

AWS CLI

So aktualisieren Sie einen Schlüsselwertspeicher

1. Ändern der Schlüssel-Wert-Paare – Sie können weitere Schlüssel-Wert-Paare hinzufügen, ein oder mehrere Schlüssel-Wert-Paare löschen und den Wert für ein vorhandenes Schlüssel-Wert-Paar ändern. Weitere Informationen finden Sie unter Arbeiten mit Schlüsselwertdaten.
2. Ändern der Funktionszuordnung für den Schlüsselwertspeicher – Informationen zum Aktualisieren der Funktionszuordnung für den Schlüsselwertspeicher finden Sie unter Zuordnen eines Schlüsselwertspeichers zu einer Funktion.

> **Tip**
>
> Sie benötigen den ARN des Schlüsselwertspeichers. Weitere Informationen finden Sie unter the section called “Abrufen eines Verweises auf einen Schlüsselwertspeicher”.

API

So aktualisieren Sie einen Schlüsselwertspeicher

1. Ändern der Schlüssel-Wert-Paare – Sie können weitere Schlüssel-Wert-Paare hinzufügen, ein oder mehrere Schlüssel-Wert-Paare löschen und den Wert für ein vorhandenes Schlüssel-Wert-Paar ändern. Weitere Informationen finden Sie unter Arbeiten mit Schlüsselwertdaten.
2. Ändern Sie die Funktionszuordnung für den Schlüsselwertspeicher — Verwenden Sie den UpdateFunctionAPI-Vorgang, um die Funktionszuordnung für den Schlüsselwertspeicher zu aktualisieren. Weitere Informationen finden Sie unter the section called “Aktualisieren von Funktionen”.

> **Tip**
>
> Sie benötigen den ARN des Schlüsselwertspeichers. Weitere Informationen finden Sie unter the section called “Abrufen eines Verweises auf einen Schlüsselwertspeicher”.

Abrufen eines Verweises auf einen Schlüsselwertspeicher

Sie benötigen das `ETag` und den Namen des Schlüsselwertspeichers, um programmgesteuert mit dem Schlüsselwertspeicher arbeiten zu können.

Um beide Werte abzurufen, können Sie die AWS Command Line Interface (AWS CLI) oder die CloudFront API verwenden.

AWS CLI

So rufen Sie den Verweis auf einen Schlüsselwertspeicher ab

1. Um eine Liste von Schlüsselwertspeichern abzurufen, führen Sie den folgenden Befehl aus. Suchen Sie den Namen des Schlüsselwertspeichers, den Sie bearbeiten möchten.

```
aws cloudfront list-key-value-stores
```

2. Suchen Sie in der Antwort den Namen des gewünschten Schlüsselwertspeichers.

   Antwort

```
{
    "KeyValueStoreList": {
        "Items": [
            {
                "Name": "keyvaluestore3",
                "Id": "37435e19-c205-4271-9e5c-example3",
                "ARN": "arn:aws:cloudfront::123456789012:key-value-
store/37435e19-c205-4271-9e5c-example3",
                "Status": "READY",
                "LastModifiedTime": "2024-05-08T14:50:18.876000+00:00"
            },
            {
                "Name": "keyvaluestore2",
                "Id": "47970d59-6408-474d-b850-example2",
                "ARN": "arn:aws:cloudfront::123456789012:key-value-
store/47970d59-6408-474d-b850-example2",
                "Status": "READY",
                "LastModifiedTime": "2024-05-30T21:06:22.113000+00:00"
            },
            {
                "Name": "keyvaluestore1",
                "Id": "8aa76c93-3198-462c-aaf6-example",
                "ARN": "arn:aws:cloudfront::123456789012:key-value-
store/8aa76c93-3198-462c-aaf6-example",
                "Status": "READY",
                "LastModifiedTime": "2024-08-06T22:19:30.510000+00:00"
            }
        ]
    }
```

```
}
```

3. Führen Sie den folgenden Befehl aus, um das `ETag` für den angegebenen Schlüsselwertspeicher abzurufen.

```
aws cloudfront describe-key-value-store \
    --name=keyvaluestore1
```

Antwort

```
{
    "ETag": "E3UN6WX5RRO2AG",
    "KeyValueStore": {
        "Name": "keyvaluestore1",
        "Id": "8aa76c93-3198-462c-aaf6-example",
        "Comment": "This is an example KVS",
        "ARN": "arn:aws:cloudfront::123456789012:key-value-
store/8aa76c93-3198-462c-aaf6-example",
        "Status": "READY",
        "LastModifiedTime": "2024-08-06T22:19:30.510000+00:00"
    }
}
```

API

So rufen Sie den Verweis auf einen Schlüsselwertspeicher ab

1. Verwenden Sie die API-Operation CloudFront ListKeyValueStores, um eine Liste von Schlüsselwertspeichern abzurufen. Suchen Sie den Namen des Schlüsselwertspeichers aus, den Sie ändern möchten.
2. Verwenden Sie die API-Operation CloudFront DescribeKeyValueStore und geben Sie den Namen des im vorherigen Schritt zurückgegebenen Schlüsselwertspeichers an.

Die Antwort enthält eine UUID, den ARN des Schlüsselwertspeichers und das `ETag` des Schlüsselwertspeichers.

- Das `ETag` ähnelt dem im Beispiel angegebenen `E3UN6WX5RRO2AG`.
- Die UUID ist 128 Bit groß, wie z. B. `8aa76c93-3198-462c-aaf6-example`.

- Der ARN enthält die AWS-Konto Zahl, die Konstante `key-value-store` und die UUID, wie im folgenden Beispiel:

  ```
  arn:aws:cloudfront::123456789012:key-value-store/8aa76c93-3198-462c-aaf6-example
  ```

Weitere Informationen über die `DescribeKeyValueStore`-Operation finden Sie unter the section called “Über den CloudFront KeyValueStore”.

Löschen eines Schlüsselwertspeichers

Sie können Ihren Key Value Store mithilfe der CloudFront Amazon-Konsole oder API löschen.

Console

So löschen Sie einen Schlüsselwertspeicher

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Seite Funktionen in der CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home#/functions.
2. Wählen Sie den Funktionsnamen aus.
3. Überprüfen Sie im KeyValueStore Abschnitt Zugeordnet, ob der Funktion ein Schlüsselwertspeicher zugeordnet ist. Ist dies der Fall, entfernen Sie die Zuordnung, indem Sie „Zuordnung trennen“ KeyValueStore und anschließend „Zuordnung entfernen“ wählen.
4. Wählen Sie im Navigationsbereich die Seite Funktionen und dann die KeyValueStoresRegisterkarte aus.
5. Wählen Sie den zu löschenden Schlüsselwertspeicher und dann Löschen aus.

AWS CLI

So löschen Sie einen Schlüsselwertspeicher

1. Rufen Sie das `ETag` und den Namen des Schlüsselwertspeichers ab. Weitere Informationen finden Sie unter the section called “Abrufen eines Verweises auf einen Schlüsselwertspeicher”.
2. Überprüfen Sie, ob der Schlüsselwertspeicher einer Funktion zugeordnet ist. Wenn dies der Fall ist, entfernen Sie die Zuordnung. Weitere Informationen zu diesen beiden Schritten finden Sie unter ???.

3. Wenn Sie den Namen und das `ETag` des Schlüsselwertspeichers abgerufen haben und dieser nicht mehr mit einer Funktion verknüpft ist, können Sie ihn löschen.

   Führen Sie den folgenden Befehl aus, um den angegebenen Schlüsselwertspeicher zu löschen.

   ```
   aws cloudfront delete-key-value-store \
       --name=keyvaluestore1 \
       --if-match=E3UN6WX5RRO2AG
   ```

API

So löschen Sie einen Schlüsselwertspeicher

1. Rufen Sie das `ETag` und den Namen des Schlüsselwertspeichers ab. Weitere Informationen finden Sie unter the section called “Abrufen eines Verweises auf einen Schlüsselwertspeicher”.
2. Überprüfen Sie, ob der Schlüsselwertspeicher einer Funktion zugeordnet ist. Wenn dies der Fall ist, entfernen Sie die Zuordnung. Weitere Informationen zu diesen beiden Schritten finden Sie unter ???.
3. Verwenden Sie den CloudFront DeleteKeyValueStoreAPI-Vorgang, um den Schlüsselwertspeicher zu löschen.

Dateiformat für Schlüssel-Wert-Paare

Verwenden Sie das folgende JSON-Format, wenn Sie eine UTF-8-kodierte Datei erstellen:

```
{
  "data":[
    {
      "key":"key1",
      "value":"value"
    },
    {
      "key":"key2",
      "value":"value"
    }
  ]
}
```

Ihre Datei darf keine doppelten Schlüssel enthalten. Wenn Sie in Ihrem Amazon-S3-Bucket eine ungültige Datei angegeben haben, können Sie die Datei aktualisieren, um alle Duplikate zu entfernen, und dann erneut versuchen, Ihren Schlüsselwertspeicher zu erstellen.

Weitere Informationen finden Sie unter Erstellen eines Schlüsselwertspeichers.

> **Note**
>
> Die Datei für Ihre Datenquelle und ihre Schlüssel-Wert-Paare haben die folgenden Beschränkungen:
>
> - Dateigröße – 5 MB
> - Schlüsselgröße – 512 Zeichen
> - Wertgröße – 1 024 Zeichen

Arbeiten mit Schlüsselwertdaten

In diesem Thema wird beschrieben, wie Schlüssel-Wert-Paare zu einem vorhandenen Schlüsselwertspeicher hinzugefügt werden. Informationen zum Einbeziehen von Schlüssel-Wert-Paaren bei der anfänglichen Erstellung des Schlüsselwertspeichers finden Sie unter the section called “Erstellen eines Schlüsselwertspeichers”.

Themen

- Arbeiten mit Schlüssel-Wert-Paaren (Konsole)
- Über den CloudFront KeyValueStore
- Arbeiten mit Schlüssel-Wert-Paaren (AWS CLI)
- Arbeiten mit Schlüssel-Wert-Paaren (API)

Arbeiten mit Schlüssel-Wert-Paaren (Konsole)

Sie können die CloudFront Konsole verwenden, um mit Ihren Schlüssel-Wert-Paaren zu arbeiten.

So arbeiten Sie mit Schlüssel-Wert-Paaren

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Seite Funktionen in der CloudFront Konsole unter. https://console.aws.amazon.com/cloudfront/v4/home#/functions

2. Wählen Sie die Registerkarte KeyValueStores aus.
3. Wählen Sie den Schlüsselwertspeicher aus, den Sie bearbeiten möchten.
4. Wählen Sie im Abschnitt Schlüssel-Wert-Paare die Option Bearbeiten aus.
5. Sie können ein Schlüssel-Wert-Paar hinzufügen oder löschen oder den Wert für ein vorhandenes Schlüssel-Wert-Paar ändern.
6. Wenn Sie fertig sind, wählen Sie Änderungen speichern aus.

Über den CloudFront KeyValueStore

> **Tip**
>
> Die CloudFront KeyValueStore API ist ein globaler Dienst, der Signature Version 4A (Sigv4A) zur Authentifizierung verwendet. Für die Verwendung temporärer Anmeldeinformationen mit SigV4A sind Sitzungstoken der Version 2 erforderlich. Weitere Informationen finden Sie unter Verwenden temporärer Anmeldeinformationen mit der CloudFront KeyValueStore API.

Wenn Sie AWS Command Line Interface (AWS CLI) oder Ihren eigenen Code zum Aufrufen der CloudFront KeyValueStore API verwenden, finden Sie weitere Informationen in den folgenden Abschnitten.

Wenn Sie mit einem Schlüsselwertspeicher und dessen Schlüssel-Wert-Paaren arbeiten, hängt der Service, den Sie aufrufen, von Ihrem Anwendungsfall ab:

- Verwenden Sie den Dienst, um mit Schlüssel-Wert-Paaren in einem vorhandenen Schlüsselwertspeicher zu arbeiten. CloudFront KeyValueStore
- Verwenden Sie den Dienst, um Schlüsselwertpaare in den Schlüsselwertspeicher aufzunehmen, wenn Sie den Schlüsselwertspeicher zum ersten Mal erstellen. CloudFront

Sowohl die CloudFront API als auch die CloudFront KeyValueStore API haben einen `DescribeKeyValueStore` Vorgang. Sie werden zu unterschiedlichen Zwecken aufgerufen. Informationen zu den Unterschieden finden Sie in der folgenden Tabelle.

	CloudFront DescribeKeyValueStore-API	CloudFront KeyValueStore DescribeKeyValueStore-API
Daten über den Schlüsselwertspeicher	Gibt Daten zurück, z. B. den Status und das Datum, an dem der Schlüsselwertspeicher zuletzt geändert wurde.	Gibt Daten über die Inhalte der Speicherressource zurück – die Schlüssel-Wert-Paare im Speicher und die Größe der Inhalte.
Daten, die den Schlüsselwertspeicher identifizieren	Gibt ein `ETag`, die UUID und den ARN des Schlüsselwertspeichers zurück.	Gibt ein `ETag` und den ARN des Schlüsselwertspeichers zurück.

Hinweise

- Jede DescribeKeyValueStore-Operation gibt ein anderes `ETag` zurück. Die `ETags` sind nicht austauschbar.
- Wenn Sie einen API-Vorgang aufrufen, um eine Aktion abzuschließen, müssen Sie das `ETag` der entsprechenden API angeben. In der CloudFront KeyValueStore DeleteKeyOperation geben Sie beispielsweise die an, `ETag` die Sie von der CloudFront KeyValueStore DescribeKeyValueStoreOperation zurückgegeben haben.
- Wenn Sie Ihre CloudFront Funktionen mithilfe von aufrufen CloudFront KeyValueStore, werden die Werte im Schlüsselwertspeicher während des Aufrufs der Funktion nicht aktualisiert oder geändert. Aktualisierungen werden zwischen den Aufrufen einer Funktion verarbeitet.

Arbeiten mit Schlüssel-Wert-Paaren (AWS CLI)

Sie können die folgenden AWS Command Line Interface Befehle für ausführen. CloudFront KeyValueStore

Inhalt

- Auflisten von Schlüssel-Wert-Paaren
- Abrufen von Schlüssel-Wert-Paaren
- Beschreiben eines Schlüsselwertspeichers
- Erstellen eines Schlüssel-Wert-Paares
- Löschen eines Schlüssel-Wert-Paares
- Aktualisieren eines Schlüssel-Wert-Paares

Auflisten von Schlüssel-Wert-Paaren

Führen Sie den folgenden Befehl aus, um Schlüssel-Wert-Paare in Ihrem Schlüsselwertspeicher aufzulisten.

```
aws cloudfront-keyvaluestore list-keys \
    --kvs-arn=arn:aws:cloudfront::123456789012:key-value-store/37435e19-c205-4271-9e5c-
example
```

Antwort

```
{
    "Items": [
        {
            "Key": "key1",
            "Value": "value1"
        }
    ]
}
```

Abrufen von Schlüssel-Wert-Paaren

Um ein Schlüssel-Wert-Paar in Ihrem Schlüsselwertspeicher abzurufen, führen Sie den folgenden Befehl aus.

```
aws cloudfront-keyvaluestore get-key \
    --key=key1 \
    --kvs-arn=arn:aws:cloudfront::123456789012:key-value-store/37435e19-c205-4271-9e5c-
example
```

Antwort

```
{
    "Key": "key1",
    "Value": "value1",
    "ItemCount": 1,
    "TotalSizeInBytes": 11
}
```

Beschreiben eines Schlüsselwertspeichers

Um einen Schlüsselwertspeicher zu beschreiben, führen Sie den folgenden Befehl aus.

```
aws cloudfront-keyvaluestore describe-key-value-store \
    --kvs-arn=arn:aws:cloudfront::123456789012:key-value-store/37435e19-c205-4271-9e5c-
example
```

Antwort

```
{
    "ETag": "KV1F83G8C2ARO7P",
    "ItemCount": 1,
    "TotalSizeInBytes": 11,
    "KvsARN": "arn:aws:cloudfront::123456789012:key-value-store/37435e19-
c205-4271-9e5c-example",
    "Created": "2024-05-08T07:48:45.381000-07:00",
    "LastModified": "2024-08-05T13:50:58.843000-07:00",
    "Status": "READY"
}
```

Erstellen eines Schlüssel-Wert-Paares

Um ein Schlüssel-Wert-Paar in Ihrem Schlüsselwertspeicher zu erstellen, führen Sie den folgenden Befehl aus.

```
aws cloudfront-keyvaluestore put-key \
    --if-match=KV1PA6795UKMFR9 \
    --key=key2 \
    --value=value2 \
    --kvs-arn=arn:aws:cloudfront::123456789012:key-value-store/37435e19-c205-4271-9e5c-
example
```

Antwort

```
{
    "ETag": "KV13V1IB3VIYZZH",
    "ItemCount": 3,
    "TotalSizeInBytes": 31
}
```

Löschen eines Schlüssel-Wert-Paares

Um ein Schlüssel-Wert-Paar zu löschen, führen Sie den folgenden Befehl aus.

```
aws cloudfront-keyvaluestore delete-key \
    --if-match=KV13V1IB3VIYZZH \
    --key=key1 \
    --kvs-arn=arn:aws:cloudfront::123456789012:key-value-store/37435e19-c205-4271-9e5c-
example
```

Ausgabe

```
{
    "ETag": "KV1VC38T7YXB528",
    "ItemCount": 2,
    "TotalSizeInBytes": 22
}
```

Aktualisieren eines Schlüssel-Wert-Paares

Sie können den `update-keys`-Befehl verwenden, um mehr als ein Schlüssel-Wert-Paar zu aktualisieren. Um beispielsweise ein vorhandenes Schlüssel-Wert-Paar zu löschen und ein anderes zu erstellen, führen Sie den folgenden Befehl aus.

```
aws cloudfront-keyvaluestore update-keys \
    --if-match=KV2EUQ1WTGCTBG2 \
    --kvs-arn=arn:aws:cloudfront::123456789012:key-value-store/37435e19-c205-4271-9e5c-
example \
    --deletes '[{"Key":"key2"}]' \
    --puts '[{"Key":"key3","Value":"value3"}]'
```

Antwort

```
{
```

```
    "ETag": "KV3AEGXETSR30VB",
    "ItemCount": 3,
    "TotalSizeInBytes": 28
}
```

Arbeiten mit Schlüssel-Wert-Paaren (API)

Folgen Sie diesem Abschnitt, um programmatisch mit Ihren Schlüssel-Wert-Paaren zu arbeiten.

Inhalt

- Abrufen eines Verweises auf einen Schlüsselwertspeicher
- Ändern von Schlüssel-Wert-Paaren in einem Schlüsselwertspeicher
- Beispielcode für CloudFront KeyValueStore

Abrufen eines Verweises auf einen Schlüsselwertspeicher

Wenn Sie die CloudFront KeyValueStore API verwenden, um einen Schreibvorgang aufzurufen, müssen Sie den ARN und den `ETag` des Schlüsselwertspeichers angeben. Gehen Sie wie folgt vor, um diese Daten abzurufen:

So rufen Sie einen Verweis auf einen Schlüsselwertspeicher ab

1. Verwenden Sie die API-Operation CloudFront ListKeyValueStores, um eine Liste von Schlüsselwertspeichern abzurufen. Suchen Sie den Schlüsselwertspeicher, den Sie bearbeiten möchten.
2. Verwenden Sie die API-Operation CloudFrontKeyValueStore DescribeKeyValueStore und geben Sie den Schlüsselwertspeicher aus dem vorherigen Schritt an.

   Die Antwort enthält den ARN und das `ETag` des Schlüsselwertspeichers.

   - Der ARN umfasst die AWS-Konto Zahl, die Konstante `key-value-store` und die UUID, wie im folgenden Beispiel:

     `arn:aws:cloudfront::123456789012:key-value-store/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111`
   - Ein `ETag` sieht z. B. folgendermaßen aus:

     `ETVABCEXAMPLE2`

Ändern von Schlüssel-Wert-Paaren in einem Schlüsselwertspeicher

Sie können den Schlüsselwertspeicher angeben, der das zu aktualisierende Schlüssel-Wert-Paar enthält.

Sehen Sie sich die folgenden CloudFront KeyValueStore API-Operationen an:

- CloudFrontKeyValueStore DeleteKey— Löscht ein Schlüssel-Wert-Paar
- CloudFrontKeyValueStore GetKey— Gibt ein Schlüssel-Wert-Paar zurück
- CloudFrontKeyValueStore ListKeys— Gibt eine Liste von Schlüssel-Wert-Paaren zurück
- CloudFrontKeyValueStore PutKey— Sie können die folgenden Aufgaben ausführen:
  - Erstellen Sie ein Schlüssel-Wert-Paar in einem Schlüsselwertspeicher, indem Sie einen neuen Schlüsselnamen und Wert angeben.
  - Legen Sie einen anderen Wert in einem vorhandenen Schlüssel-Wert-Paar fest, indem Sie einen vorhandenen Schlüsselnamen und einen neuen Schlüsselwert angeben.
- CloudFrontKeyValueStore UpdateKeys— Sie können eine oder mehrere der folgenden Aktionen in einem all-or-nothing Vorgang ausführen:
  - Löschen eines oder mehrerer Schlüssel-Wert-Paare
  - Erstellen eines oder mehrerer neuer Schlüssel-Wert-Paare
  - Festlegen eines anderen Werts in einem oder mehreren vorhandenen Schlüssel-Wert-Paaren

Beispielcode für CloudFront KeyValueStore

Example

Der folgende Code zeigt Ihnen, wie Sie die API-Operation `DescribeKeyValueStore` für einen Schlüsselwertspeicher aufrufen.

```
const {
  CloudFrontKeyValueStoreClient,
  DescribeKeyValueStoreCommand,
} = require("@aws-sdk/client-cloudfront-keyvaluestore");

require("@aws-sdk/signature-v4-crt");

(async () => {
  try {
    const client = new CloudFrontKeyValueStoreClient({
```

```
      region: "us-east-1"
    });
    const input = {
      KvsARN: "arn:aws:cloudfront::123456789012:key-value-store/a1b2c3d4-5678-90ab-
cdef-EXAMPLE11111",
    };
    const command = new DescribeKeyValueStoreCommand(input);

    const response = await client.send(command);
  } catch (e) {
    console.log(e);
  }
})();
```

Mit CloudFront Verbindungsfunktionen anpassen

CloudFront Mit den Verbindungsfunktionen können Sie einfache JavaScript Funktionen für die mTLS-Zertifikatsvalidierung und benutzerdefinierte Authentifizierungslogik schreiben. Ihre Verbindungsfunktionen werden während des mTLS-Verbindungsaufbaus ausgeführt, um Client-Zertifikate zu validieren, gerätespezifische Authentifizierungsregeln zu implementieren und Zertifikatssperrszenarien zu bewältigen. Die Connection Functions-Laufzeitumgebung bietet Startzeiten im Submillisekundenbereich, skaliert sofort, um Millionen von Verbindungen pro Sekunde zu verarbeiten, und ist äußerst sicher. Connection Functions sind eine native Funktion von CloudFront. Sie können Ihren Code also vollständig darin erstellen, testen und bereitstellen. CloudFront

Wenn Sie eine Verbindungsfunktion mit einer TLS-enabled CloudFront M-Distribution verknüpfen, CloudFront fängt es TLS-Verbindungsanfragen an CloudFront Edge-Standorten ab und leitet Zertifikatsinformationen an Ihre Funktion weiter. Sie können Verbindungsfunktionen aufrufen, wenn das folgende Ereignis eintritt:

- Während des TLS-Verbindungsaufbaus (Verbindungsanfrage) — für gegenseitige TLS-Verbindungen (mTLS)

Weitere Informationen zu Verbindungsfunktionen finden Sie in den folgenden Themen.

Themen

- Überblick und Arbeitsablauf
- Konfiguration und Grenzwerte

- Erstellen Sie CloudFront Verbindungsfunktionen für die gegenseitige TLS-Validierung (Viewer)
- Schreiben Sie den CloudFront Verbindungsfunktionscode für die gegenseitige TLS-Validierung (Viewer)
- Testen Sie die CloudFront Verbindungsfunktionen vor der Bereitstellung
- Ordnen Sie Verbindungsfunktionen Verteilungen zu
- Implementieren Sie den Widerruf von Zertifikaten für gegenseitiges TLS (Viewer) mit Functions und CloudFront KeyValueStore

Überblick und Arbeitsablauf

CloudFront Verbindungsfunktionen sind eine spezielle Art von CloudFront Funktionen, die während des TLS-Handshakes ausgeführt werden, wenn ein Client versucht, eine mTLS-Verbindung herzustellen. Ihre Verbindungsfunktion kann auf Client-Zertifikatsinformationen, mTLS-Konfigurationsparameter, Ergebnisse der Überprüfung der Zertifikatssperrung und die Client-IP-Adresse zugreifen.

Die Verbindungsfunktionen werden nach CloudFront der Standardvalidierung des Zertifikats (Vertrauenskette, Ablauf, Signaturüberprüfung) aufgerufen, können aber auch dann ausgeführt werden, wenn die Zertifikatssperrprüfungen fehlschlagen. Auf diese Weise können Sie eine benutzerdefinierte Logik für den Umgang mit gesperrten Zertifikaten implementieren oder zusätzliche Validierungskriterien hinzufügen.

Nachdem Sie eine Verbindungsfunktion erstellt und veröffentlicht haben, stellen Sie sicher, dass Sie eine Verknüpfung für den Ereignistyp der Verbindungsanforderung mit einer TLS-enabled M-Verteilung hinzufügen. Dadurch wird die Funktion jedes Mal ausgeführt, wenn ein Client versucht, eine mTLS-Verbindung mit CloudFront herzustellen.

CloudFront Verbindungsfunktionen folgen einem zweistufigen Lebenszyklus, mit dem Sie Funktionen entwickeln und testen können, bevor Sie sie in der Produktion einsetzen. Dieser Workflow stellt sicher, dass Ihre Verbindungsfunktionen ordnungsgemäß funktionieren, bevor sie sich auf den Live-Verkehr auswirken.

Themen

- Stufen der Funktionen
- Entwicklungs-Workflow
- Unterschiede zu anderen Funktionstypen

Stufen der Funktionen

Verbindungsfunktionen bestehen in einer von zwei Stufen:

- ENTWICKLUNG — Funktionen in dieser Phase können geändert, getestet und aktualisiert werden. Verwenden Sie diese Phase zum Schreiben und Debuggen Ihres Funktionscodes.
- LIVE — Die Funktionen in dieser Phase sind schreibgeschützt und verarbeiten den Produktionsdatenverkehr. Sie können Funktionen in der LIVE-Phase nicht direkt ändern.

Wenn Sie eine neue Verbindungsfunktion erstellen, beginnt sie in der DEVELOPMENT-Phase. Nach dem Testen und Validieren veröffentlichen Sie die Funktion, um sie in die LIVE-Phase zu verschieben.

Entwicklungs-Workflow

Folgen Sie diesem Arbeitsablauf, um Verbindungsfunktionen zu entwickeln und bereitzustellen:

1. Erstellen — Erstellen Sie in der ENTWICKLUNGSPHASE eine neue Verbindungsfunktion mit Ihrem ursprünglichen Code und Ihrer Konfiguration.
2. Test — Verwenden Sie die Testfunktion, um Ihre Funktion vor der Bereitstellung anhand von Beispielverbindungsereignissen zu validieren.
3. Update — Ändern Sie den Funktionscode und die Konfiguration je nach Bedarf auf der Grundlage der Testergebnisse.
4. Veröffentlichen — Wenn Sie bereit für die Produktion sind, veröffentlichen Sie die Funktion, um sie von der ENTWICKLUNGS- in die LIVE-Phase zu übertragen.
5. Zuordnen — Ordnen Sie die veröffentlichte Funktion Ihrer TLS-enabled M-Distribution zu, um Live-Verbindungen abzuwickeln.

Um Änderungen an einer LIVE-Funktion vorzunehmen, müssen Sie die DEVELOPMENT-Version aktualisieren und erneut veröffentlichen. Dadurch wird eine neue Version in der LIVE-Phase erstellt.

Unterschiede zu anderen Funktionstypen

Die Verbindungsfunktionen unterscheiden sich in mehreren wichtigen Punkten von den Funktionen für Zuschaueranfragen und Zuschauerantworten:

- Die Verbindungsfunktionen werden nach dem mTLS-Handshake ausgeführt, bevor eine HTTP-Verarbeitung stattfindet

- Verbindungsfunktionen haben Zugriff auf TLS-Zertifikatsinformationen anstelle von HTTP-Daten request/response
- Verbindungsfunktionen können die Verbindung nur zulassen oder verweigern, HTTP-Daten nicht ändern
- Verbindungsfunktionen werden nur für neue TLS-Verbindungen aufgerufen, nicht für die Wiederverwendung von Verbindungen
- Die Wiederaufnahme der TLS-Sitzung wird mit mTLS nicht unterstützt, um sicherzustellen, dass die Zertifikatsvalidierung bei jeder Verbindung erfolgt
- Verbindungsfunktionen werden zusätzlich zu den Standardfunktionen für Viewer-Anfragen und Viewer-Antworten ausgeführt
- Verbindungsfunktionen ordnen Sie auf der Verteilungsebene und nicht auf der Cache-Verhaltensebene zu.
- Verbindungsfunktionen unterstützen nur JavaScript Runtime 2.0.

Konfiguration und Grenzwerte

CloudFront Für Connection Functions gelten aufgrund ihrer speziellen Rolle bei der TLS-Verbindungsvalidierung und der Leistungsanforderungen des Edge-Computings spezifische Konfigurationsanforderungen und Dienstbeschränkungen.

Themen

- Anforderungen an den Funktionscode
- Service Limits
- Optionen zum Filtern von Funktionen

Anforderungen an den Funktionscode

Verbindungsfunktionen erfordern JavaScript Code, der TLS-Verbindungsereignisse verarbeitet. Der Funktionscode muss:

- Eingeschrieben werden JavaScript
- Verarbeiten Sie Verbindungsereignisse und treffen Sie allow/deny Entscheidungen
- Vollständige Ausführung innerhalb der Fristen
- Behandeln Sie die Logik zur Zertifikat- und Verbindungsvalidierung

Service Limits

Verbindungsfunktionen unterliegen den folgenden Beschränkungen:

- Funktionsgröße — Funktionscode und Konfiguration sind in ihrer Größe begrenzt
- Ausführungszeit — Funktionen haben strenge Ausführungszeitbeschränkungen für die TLS-Verbindungsverarbeitung
- Zuordnungsbeschränkungen — Jeder Distribution kann nur eine Verbindungsfunktion zugeordnet sein
- Einschränkungen für die Phase — Nur LIVE-Staging-Funktionen können Distributionen zugeordnet werden

Optionen zum Filtern von Funktionen

Beim Auflisten von Verbindungsfunktionen können Sie die folgenden Filter verwenden:

- Stufenfilter — Filtern Sie nach ENTWICKLUNGS- oder LIVE-Stufe
- Zuordnungsfilter — Filtern Sie nach Distributions-ID oder Schlüssel-Wert-Speicher-ID-Zuordnungen

Diese Filter helfen Ihnen dabei, Verbindungsfunktionen in verschiedenen Umgebungen und Anwendungsfällen zu organisieren und zu verwalten.

Erstellen Sie CloudFront Verbindungsfunktionen für die gegenseitige TLS-Validierung (Viewer)

Sie erstellen eine CloudFront Verbindungsfunktion in zwei Schritten:

1. Erstellen Sie den Funktionscode als JavaScript. Sie können das Standardbeispiel aus der CloudFront Konsole verwenden oder Ihr eigenes schreiben. Weitere Informationen finden Sie unter den folgenden Themen:
   - Schreiben Sie den CloudFront Verbindungsfunktionscode für die mTLS-Validierung
   - CloudFront Ereignisstruktur und Antwortformat der Verbindungsfunktion
   - Code-Beispiele für Verbindungsfunktionen
2. Verwenden Sie CloudFront , um die Verbindungsfunktion zu erstellen und Ihren Code einzufügen. Der Code ist innerhalb der Funktion vorhanden (nicht als Referenz).

Themen

- CloudFront Konsole
- AWS CLI

CloudFront Konsole

Um eine Verbindungsfunktion zu erstellen

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home.
2. Wählen Sie Funktion erstellen.
3. Geben Sie einen Funktionsnamen ein, der innerhalb von eindeutig ist AWS-Konto, wählen Sie Verbindungsfunktion als Funktionstyp aus, und klicken Sie dann auf Weiter.
4. Die Detailseite für die neue Verbindungsfunktion wird angezeigt.

   > **Note**
   >
   > Verbindungsfunktionen unterstützen nur JavaScript Runtime 2.0. Um die CloudFront KeyValueStore Verbindungsfunktionsintegration in der Funktion verwenden zu können, müssen Sie diese Runtime-Version verwenden.

5. Wählen Sie im Abschnitt Funktionscode die Registerkarte Build und geben Sie Ihren Verbindungsfunktionscode ein. Der Beispielcode, der auf der Registerkarte „Build“ enthalten ist, veranschaulicht die grundlegende Syntax für den Verbindungsfunktionscode.
6. Wählen Sie Änderungen speichern aus.
7. Wenn der Verbindungsfunktionscode KeyValueStore zur Überprüfung des Zertifikats oder zur Gerätevalidierung verwendet wird, müssen Sie eine zuordnen KeyValueStore.

   Sie können das verknüpfen KeyValueStore , wenn Sie die Funktion zum ersten Mal erstellen. Sie können sie auch später zuordnen, indem Sie Verbindungsfunktionen zuordnen.

   Gehen Sie folgendermaßen vor, KeyValueStore um einen Knoten zuzuordnen:

   - Gehen Sie zum KeyValueStore Abschnitt Associate und wählen Sie Associate existing aus KeyValueStore.
   - Wählen Sie KeyValueStore das aus, das die Zertifikatsdaten für Ihre Verbindungsfunktion enthält, und wählen Sie dann Associate aus KeyValueStore.

CloudFront ordnet den Speicher sofort der Funktion zu. Sie müssen die Funktion nicht speichern.

AWS CLI

Wenn Sie den verwenden AWS CLI, erstellen Sie in der Regel zuerst den Verbindungsfunktionscode in einer Datei und dann die Funktion mit der AWS CLI.

So erstellen Sie eine Verbindungsfunktion

1. Erstellen Sie den Verbindungsfunktionscode in einer Datei und speichern Sie ihn in einem Verzeichnis, mit dem Ihr Computer eine Verbindung herstellen kann.
2. Führen Sie den Befehl wie im Beispiel veranschaulicht aus. In diesem Beispiel wird die Notation `fileb://` verwendet, um die Datei zu übergeben. Es sind Zeilenumbrüche enthalten, um den Befehl lesbarer zu machen.

```
aws cloudfront create-connection-function \
    --name CertificateValidator \
    --connection-function-config '{
        "Comment":"Device certificate validation",
        "Runtime":"cloudfront-js-2.0",
        "KeyValueStoreAssociations":{
            "Quantity":1,
            "Items":[{
                "KeyValueStoreARN":"arn:aws:cloudfront::111122223333:key-value-
store/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
            }]
        }
    }' \
    --connection-function-code fileb://certificate-validator.js
```

> **Note**
>
> - Runtime — Verbindungsfunktionen unterstützen nur JavaScript Runtime 2.0 (cloudfront-js-2.0).
> - KeyValueStoreAssociations— Wenn Ihre Verbindungsfunktion die Zertifikatsvalidierung verwendet KeyValueStore , können Sie diese KeyValueStore bei der ersten Erstellung der Funktion zuordnen. Oder Sie können sie später mit der

Funktion update-connection-function verknüpfen. Die Menge ist immer 1, da jeder Verbindungsfunktion nur eine KeyValueStore zugeordnet werden kann.

3. Wenn der Befehl erfolgreich ausgeführt wurde, wird die Ausgabe folgendermaßen angezeigt.

```
ETag: ETVABCEXAMPLE
ConnectionFunctionSummary:
  ConnectionFunctionConfig:
    Comment: Device certificate validation
    Runtime: cloudfront-js-2.0
    KeyValueStoreAssociations:
      Quantity: 1
      Items:
        - KeyValueStoreARN: arn:aws:cloudfront::111122223333:key-value-store/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
  ConnectionFunctionMetadata:
    CreatedTime: '2024-09-04T16:32:54.292000+00:00'
    ConnectionFunctionARN: arn:aws:cloudfront::111122223333:connection-function/
CertificateValidator
    LastModifiedTime: '2024-09-04T16:32:54.292000+00:00'
    Stage: DEVELOPMENT
  Name: CertificateValidator
  Status: UNPUBLISHED
Location: https://cloudfront.amazonaws.com/2020-05-31/connection-function/
arn:aws:cloudfront:::connection-function/CertificateValidator
```

Die meisten Informationen werden aus der Anfrage wiederholt. Weitere Informationen werden von hinzugefügt CloudFront.

> **Note**
>
> - ETag — Dieser Wert ändert sich jedes Mal, wenn Sie die Verbindungsfunktion ändern. Sie benötigen diesen Wert, um die Funktion zu aktualisieren oder zu veröffentlichen.
> - Phase — Neue Verbindungsfunktionen beginnen in der ENTWICKLUNGSPHASE. Sie müssen die Funktion veröffentlichen, um sie in die LIVE-Phase zu verschieben, bevor Sie sie einer Distribution zuordnen.
> - Status — Der Funktionsstatus lautet UNPUBLISHED, bis Sie sie auf der LIVE-Bühne veröffentlichen.

Schreiben Sie den CloudFront Verbindungsfunktionscode für die gegenseitige TLS-Validierung (Viewer)

CloudFront Mit den Verbindungsfunktionen können Sie einfache JavaScript Funktionen für die mTLS-Zertifikatsvalidierung und benutzerdefinierte Authentifizierungslogik schreiben. Ihr Verbindungsfunktionscode kann Client-Zertifikate validieren, gerätespezifische Authentifizierungsregeln implementieren, Zertifikatssperrszenarien bewältigen und allow/deny Entscheidungen für TLS-Verbindungen an CloudFront Edge-Standorten weltweit treffen.

Verbindungsfunktionen bieten eine leistungsstarke Möglichkeit, die CloudFront integrierte Zertifikatsvalidierung um Ihre eigene Geschäftslogik zu erweitern. Im Gegensatz zu Viewer-Request- und Viewer-Response-Funktionen, die HTTP-Daten verarbeiten, arbeiten Verbindungsfunktionen auf der TLS-Ebene und haben Zugriff auf Zertifikatsinformationen, Client-IP-Adressen und TLS-Verbindungsdetails. Dadurch eignen sie sich ideal für die Implementierung von Zero-Trust-Sicherheitsmodellen, Geräteauthentifizierungssystemen und benutzerdefinierten Richtlinien zur Zertifikatsvalidierung, die über die standardmäßige PKI-Validierung hinausgehen.

Ihr Verbindungsfunktionscode wird in einer sicheren, isolierten Umgebung mit Startzeiten von unter einer Millisekunde ausgeführt und kann so skaliert werden, dass er Millionen von Verbindungen pro Sekunde verarbeitet. Die Runtime ist für Workloads zur Zertifikatsvalidierung optimiert und bietet eine integrierte Integration CloudFront KeyValueStore für Datenabfragen in Echtzeit. Dadurch sind ausgefeilte Authentifizierungsszenarien wie die Überprüfung von Zertifikatssperrlisten und die Validierung von Zulassungslisten für Geräte möglich.

Informationen zum Schreiben eines effektiven Verbindungsfunktionscodes finden Sie in den folgenden Themen. Vollständige Codebeispiele und schrittweise Anleitungen finden Sie in den Tutorial-Abschnitten dieses Handbuchs. Erkunden Sie auch die Beispiele für Verbindungsfunktionen, die in der CloudFront Konsole verfügbar sind.

Themen

- CloudFront Anwendungsfälle und Zwecke der Verbindungsfunktion
- CloudFront Verbindungsfunktion, Ereignisstruktur und Antwortformat
- CloudFront Verbindungsfunktionen, JavaScript Laufzeitfunktionen
- CloudFront Hilfsmethoden und APIs für Verbindungsfunktionen
- CloudFront Verbindung, Funktion KeyValueStore , Integration
- Verwenden Sie async und await

- Beispiele für Codes für Verbindungsfunktionen

CloudFront Anwendungsfälle und Zwecke der Verbindungsfunktion

Bevor Sie Ihre CloudFront Verbindungsfunktion schreiben, sollten Sie sorgfältig festlegen, welche Art von Zertifikatsvalidierungs- oder Authentifizierungslogik Sie implementieren müssen. Die Verbindungsfunktionen sind für bestimmte Anwendungsfälle konzipiert, die eine benutzerdefinierte Validierung erfordern, die über die standardmäßige PKI-Zertifikatsprüfung hinausgeht. Wenn Sie Ihren Anwendungsfall verstehen, können Sie effizienten Code entwerfen, der Ihre Sicherheitsanforderungen erfüllt und gleichzeitig eine optimale Leistung gewährleistet.

Zu den häufigsten Anwendungsfällen der Verbindungsfunktion gehören:

- Umgang mit Zertifikatssperrungen — Implementieren Sie benutzerdefinierte Richtlinien für den Umgang mit gesperrten Zertifikaten, einschließlich Übergangsfristen für die Zertifikatsrotation, Ausnahmen für vertrauenswürdige Netzwerke für interne Geräte oder Notfallzugriffsszenarien, in denen widerrufene Zertifikate möglicherweise vorübergehend abgerufen werden müssen.
- Optionale mTLS-Unterstützung — Behandeln Sie sowohl mTLS- als auch Nicht-MTLS-Verbindungen mit unterschiedlichen Authentifizierungsrichtlinien. So können Sie Clients, die Zertifikate unterstützen, mehr Sicherheit bieten und gleichzeitig die Kompatibilität mit älteren Clients aufrechterhalten.
- IP-based Authentifizierung — Kombinieren Sie die Zertifikatsvalidierung mit der Überprüfung der Client-IP-Adressen, um die Sicherheit zu erhöhen, z. B. indem Sie den Zugriff von bestimmten geografischen Regionen, Unternehmensnetzwerken oder bekannten bösartigen IP-Bereichen aus einschränken.
- Multi-tenant Zertifikatsvalidierung — Implementieren Sie mandantenspezifische Validierungsregeln, für die je nach den Attributen des Ausstellers oder des Antragstellers des Client-Zertifikats unterschiedliche Zertifizierungsstellen oder Validierungskriterien gelten.
- Time-based Zugriffskontrolle — Setzen Sie zeitabhängige Einschränkungen durch, sodass Zertifikate nur zu bestimmten Zeiten, Wartungsfenstern oder Geschäftsperioden gültig sind, auch wenn das Zertifikat selbst noch nicht abgelaufen ist.

Die Verbindungsfunktionen werden nach der CloudFront Standardüberprüfung des Zertifikats (Überprüfung der Vertrauenskette, Ablaufprüfungen und Signaturvalidierung) ausgeführt, jedoch bevor die TLS-Verbindung hergestellt wird. Dieses Timing gibt Ihnen die Flexibilität, benutzerdefinierte Validierungskriterien hinzuzufügen und gleichzeitig von der CloudFront integrierten

Zertifikatsvalidierung zu profitieren. Ihre Funktion erhält die Ergebnisse der Standardvalidierung und kann anhand von Standard- und benutzerdefinierten Kriterien fundierte Entscheidungen darüber treffen, ob die Verbindung zugelassen oder verweigert werden soll.

Berücksichtigen Sie beim Entwerfen Ihrer Verbindungsfunktion die Auswirkungen Ihrer Validierungslogik auf die Leistung. Funktionen haben ein Ausführungslimit von 5 Millisekunden, daher sollten komplexe Operationen hinsichtlich ihrer Geschwindigkeit optimiert werden. Verwenden Sie diese KeyValueStore Option eher für schnelle Datenabfragen als für komplexe Berechnungen und strukturieren Sie Ihre Validierungslogik so, dass sie bei ungültigen Zertifikaten schnell fehlschlägt.

CloudFront Verbindungsfunktion, Ereignisstruktur und Antwortformat

CloudFront Verbindungsfunktionen erhalten eine andere Ereignisstruktur als Viewer-Anforderungs- und Viewer-Antwortfunktionen. Anstelle von request/response HTTP-Daten erhalten Verbindungsfunktionen Zertifikat- und Verbindungsinformationen, anhand derer Sie Authentifizierungsentscheidungen treffen können.

Themen

- Ereignisstruktur für Verbindungsfunktionen
- Antwortformat für Verbindungsfunktionen

Ereignisstruktur für Verbindungsfunktionen

Verbindungsfunktionen erhalten ein Ereignisobjekt, das Zertifikat und Verbindungsinformationen enthält. Die Eventstruktur der Funktion ist unten dargestellt:

```
{
  "clientCertificate": {
    "certificates": {
      "leaf": {
        "serialNumber": "string",
        "issuer": "string",
        "subject": "string",
        "validity": {
          "notBefore": "string",
          "notAfter": "string"
        },
        "sha256Fingerprint": "string",
        "ocspEndpoint": "string",
        "san": "string",
```

```
        "spki": "string",
        "sanHex": "string"
      },
      "intermediates": [
        {
          "serialNumber": "string",
          "issuer": "string",
          "subject": "string",
          "validity": {
            "notBefore": "string",
            "notAfter": "string"
          },
          "sha256Fingerprint": "string",
          "ocspEndpoint": "string",
          "san": "string",
          "spki": "string",
          "sanHex": "string"
        }
      ]
    }
  },
  "clientIp": "string",
  "endpoint": "string",
  "distributionId": "string",
  "connectionId": "string"
}
```

Unten finden Sie ein Beispiel für die Event-Objektstruktur:

```
{
  "clientCertificate": {
    "certificates": {
      "leaf": {
        "serialNumber": "00:9e:2a:af:16:56:e5:47:25:7d:2e:38:c3:f9:9d:57:fa",
        "issuer": "C=US, O=Ram, OU=Edge, ST=WA, CN=mTLS-CA, L=Snoqualmie",
        "subject": "C=US, O=Ram, OU=Edge, ST=WA, CN=mTLS-CA, L=Snoqualmie",
        "validity": {
          "notBefore": "2025-09-10T23:43:10Z",
          "notAfter": "2055-09-11T00:43:02Z"
        },
        "sha256Fingerprint": "_w6bJ7aOAlGOj7NUhJxTfsfee-ONg_xop3_PTgTJpqs=",
        "ocspEndpoint": "http://ocsp.example.com",
```

```
        "san": "DNS:api.example.com,DNS:www.example.com,URI:urn:example:client:123,IP
 Address:192.168.1.100,email:alejandro_rosalez@example.com",
        "spki":
 "30820122300d06092a864886f70d0101010500038201 0f003082010a0282010100b8efe87db798500320936...",
        "sanHex":
 "3058820f6170692e6578616d706c652e636f6d820f7777772e6578616d706c652e636f6d8619..."
      },
      "intermediates": [
        {
          "serialNumber": "00:d4:04:a5:fd:cb:20:37:54",
          "issuer": "C=US, O=Example Corp, CN=Example Root CA",
          "subject": "C=US, O=Example Corp, CN=Example Intermediate CA",
          "validity": {
            "notBefore": "2024-01-01T00:00:00Z",
            "notAfter": "2034-01-01T23:59:59Z"
          },
          "sha256Fingerprint": "3a8c6b...",
          "ocspEndpoint": "http://ocsp.example.com/intermediate",
          "san": "DNS:ca.example.com,URI:urn:ca:intermediate:001",
          "spki":
 "30820122300d06092a864886f70d0101010500038201 0f003082010a0282010100b389dd513b0f1f14cac1...",
          "sanHex": "302b820e63612e6578616d706c652e636f6d..."
        }
      ]
    }
  },
  "clientIp": "192.0.2.1",
  "endpoint": "d3lch071jze0cb.cloudfront.net",
  "distributionId": "E1NXS4MQZH501R",
  "connectionId": "NpvTe1925xfj24a67sPQr7ae42BIq03FGhJJKfrQYWZcWZFp96SIIg=="
}
```

Antwortformat für Verbindungsfunktionen

Ihre Verbindungsfunktion muss ein Antwortobjekt zurückgeben, das angibt, ob die Verbindung zugelassen oder verweigert werden soll. Verwenden Sie die Hilfsmethoden, um Verbindungsentscheidungen zu treffen:

```
function connectionHandler(connection) {
    // Helper methods to allow or deny connections
    if (/* some logic to determine if function should allow connection */) {
        connection.allow();
    } else {
```

```
        connection.deny();
    }
}
```

Im Gegensatz zu den Funktionen für Zuschaueranfragen und Zuschauerantworten können Verbindungsfunktionen keine HTTP-Anfragen oder -Antworten ändern. Sie können die TLS-Verbindung nur zulassen oder verweigern.

CloudFront Verbindungsfunktionen, JavaScript Laufzeitfunktionen

CloudFront Connection Functions verwenden CloudFront Functions JavaScript Runtime 2.0, das eine sichere und leistungsstarke Umgebung bietet, die speziell für Workloads zur Zertifikatsvalidierung optimiert wurde. Die Laufzeit ist so konzipiert, dass sie in weniger als einer Millisekunde startet und Millionen von gleichzeitigen Ausführungen im globalen Edge-Netzwerk verarbeitet. CloudFront

Die Laufzeitumgebung umfasst umfassende Sprachunterstützung: JavaScript

- Unterstützung für ECMAScript 2020 (ES11) — Moderne JavaScript Funktionen, einschließlich optionaler Verkettung (?.) und Nullkoaleszenz (??)
- Built-in Objekte — JavaScript Standardobjekte wie Object, Array, JSON, Math und Date
- Konsolenprotokollierung — Verwenden Sie console.log () zum Debuggen und Überwachen von Entscheidungen zur Zertifikatsvalidierung. Protokolle sind während des Testens in Echtzeit verfügbar und können bei der Fehlerbehebung bei der Validierungslogik in der Entwicklung helfen
- KeyValueStore Integration — Nativer Zugriff auf CloudFront KeyValueStore für ultraschnelle Datenabfragen, der die Überprüfung von Zertifikatssperrungen in Echtzeit, die Validierung von Zulassungslisten für Geräte und den Abruf mandantenspezifischer Konfigurationen ermöglicht

Die Verbindungsfunktionen sind für eine hohe Leistung bei der Zertifikatsvalidierung optimiert. Die Runtime kümmert sich automatisch um die Speicherverwaltung, die Speicherbereinigung und die Bereinigung von Ressourcen, um eine konsistente Leistung über Millionen gleichzeitiger Verbindungen hinweg sicherzustellen. Alle Operationen sind so konzipiert, dass sie deterministisch und schnell sind, wobei KeyValueStore Suchvorgänge in der Regel in Mikrosekunden abgeschlossen werden.

Die Laufzeitumgebung ist zwischen den Funktionsausführungen vollständig isoliert, wodurch sichergestellt wird, dass keine Daten zwischen verschiedenen Client-Verbindungen verloren gehen. Jede Funktionsausführung beginnt in einem sauberen Zustand und hat keinen Zugriff auf frühere Ausführungsergebnisse oder Client-Daten von anderen Verbindungen.

CloudFront Hilfsmethoden und APIs für Verbindungsfunktionen

CloudFront Verbindungsfunktionen bieten spezielle Hilfsmethoden, die dazu dienen, Entscheidungen zur Zertifikatsvalidierung zu vereinfachen und die Beobachtbarkeit zu verbessern. Diese Methoden sind für den Arbeitsablauf bei der Verbindungsvalidierung optimiert und lassen sich nahtlos in CloudFront Verbindungsprotokollierungs- und Überwachungssysteme integrieren.

- connection.allow () — Erlaubt die Fortsetzung der TLS-Verbindung. Diese Methode signalisiert CloudFront , den TLS-Handshake abzuschließen und dem Client zu ermöglichen, die Verbindung herzustellen. Verwenden Sie diese Methode, wenn die Zertifikatsvalidierung erfolgreich ist und die benutzerdefinierte Authentifizierungslogik erfüllt ist
- connection.deny () — Lehnen Sie die TLS-Verbindung ab und beenden Sie den Handshake. Diese Methode schließt sofort die Verbindung und verhindert, dass HTTP-Verkehr fließt. Der Client erhält einen TLS-Verbindungsfehler. Verwenden Sie dies bei ungültigen Zertifikaten, fehlgeschlagener Authentifizierung oder Richtlinienverstößen
- connection.log CustomData () — Fügen Sie benutzerdefinierte Daten zu Verbindungsprotokollen hinzu (bis zu 800 Byte UTF-8 Text). Mit dieser Methode können Sie Validierungsergebnisse, Zertifikatsdetails oder Entscheidungsgründe zur Sicherheitsüberwachung, Konformitätsprüfung und Problembehandlung in CloudFront Verbindungsprotokolle aufnehmen

Diese Methoden bieten eine übersichtliche, deklarative Oberfläche, über die Verbindungsentscheidungen getroffen und relevante Informationen für die Überwachung und das Debuggen protokolliert werden können. Das allow/deny Muster stellt sicher, dass die Absicht Ihrer Funktion klar ist, und das CloudFront kann die Verbindungsbehandlung auf der Grundlage Ihrer Entscheidung optimieren. Benutzerdefinierte Protokollierungsdaten sind sofort in CloudFront Verbindungsprotokollen verfügbar und können zusammen mit Protokollanalysetools zur Sicherheitsüberwachung und für betriebliche Einblicke verwendet werden.

Rufen Sie immer entweder connection.allow () oder connection.deny () auf, bevor Ihre Funktion abgeschlossen ist. Wenn keine der Methoden aufgerufen wird, CloudFront wird die Verbindung aus Sicherheitsgründen standardmäßig verweigert.

CloudFront Verbindung, Funktion KeyValueStore , Integration

CloudFront Verbindungsfunktionen können verwendet werden CloudFront KeyValueStore , um ultraschnelle Datenabfragen für Zertifikatsvalidierungsszenarien durchzuführen. KeyValueStore ist besonders leistungsstark für Connection Functions, da es globalen, letztlich konsistenten Datenzugriff

mit Suchzeiten im Mikrosekundenbereich an allen Edge-Standorten ermöglicht. CloudFront Dadurch eignet es sich ideal für die Verwaltung von Zertifikatssperrlisten, Gerätezulassungslisten, Mandantenkonfigurationen und anderen Validierungsdaten, auf die bei TLS-Handshakes zugegriffen werden muss.

KeyValueStore Die Integration wurde speziell für leistungsstarke Workflows zur Verbindungsvalidierung entwickelt:

- kvs Handle.exists (key) — Prüft, ob ein Schlüssel in der existiert, KeyValueStore ohne den Wert abzurufen. Dies ist die effizienteste Methode für binäre Validierungsszenarien wie die Überprüfung von Zertifikatssperrungen, bei denen Sie nur wissen müssen, ob die Seriennummer eines Zertifikats in einer Sperrliste enthalten ist
- kvs Handle.get (key) — Ruft einen Wert aus dem KeyValueStore für komplexere Validierungsszenarien ab. Verwenden Sie diese Option, wenn Sie auf Konfigurationsdaten, Validierungsregeln oder Metadaten zugreifen müssen, die einem Zertifikat oder einer Gerätekennzeichnung zugeordnet sind

KeyValueStore Operationen sind asynchron und müssen mit async/await Syntax verwendet werden. Der KeyValueStore hat eine Gesamtgrößenbeschränkung von 10 MB und unterstützt bis zu 10 Millionen Schlüssel-Wert-Paare. KeyValueStore Die Daten sind letztendlich an allen Edge-Standorten konsistent, wobei Aktualisierungen in der Regel innerhalb von Sekunden übertragen werden.

Für eine optimale Leistung sollten Sie Ihre KeyValueStore Schlüssel so strukturieren, dass Suchvorgänge minimiert werden. Verwenden Sie die Seriennummern von Zertifikaten als Schlüssel für die einfache Sperrüberprüfung, oder erstellen Sie zusammengesetzte Schlüssel, die den Aussteller-Hash und die Seriennummer für Umgebungen mit mehreren Zertifizierungsstellen kombinieren. Berücksichtigen Sie beim Entwurf Ihrer Datenstruktur die Kompromisse zwischen Schlüsselkomplexität und KeyValueStore Kapazität.

Verwenden Sie async und await

Verbindungsfunktionen unterstützen asynchrone Operationen mithilfe der async/await Syntax, was bei der Arbeit mit KeyValueStore Vorgängen oder anderen asynchronen Aufgaben unerlässlich ist. Das async/await Muster stellt sicher, dass Ihre Funktion darauf wartet, dass KeyValueStore Suchvorgänge abgeschlossen sind, bevor Verbindungsentscheidungen getroffen werden, während gleichzeitig die für die TLS-Handshake-Verarbeitung erforderlichen Leistungsmerkmale beibehalten werden.

Die richtige async/await Verwendung ist für Verbindungsfunktionen von entscheidender Bedeutung, da die KeyValueStore Operationen zwar sehr schnell sind, aber dennoch Netzwerkvorgänge sind, die eine Koordination über die verteilte Infrastruktur hinweg CloudFront erfordern. Die Laufzeitumgebung übernimmt automatisch die Erfüllung von Versprechen und stellt sicher, dass Ihre Funktion innerhalb der Ausführungsgrenze von 5 Millisekunden abgeschlossen wird.

Example: Asynchrone Verbindungsfunktion mit KeyValueStore

```
import cf from 'cloudfront';

async function connectionHandler(connection) {
    const kvsHandle = cf.kvs();

    // Async operation to check KeyValueStore for certificate revocation
    const isRevoked = await
 kvsHandle.exists(connection.clientCertificate.certificates.leaf.serialNumber);

    if (isRevoked) {
        // Log the revocation decision with certificate details
        connection.logCustomData(`REVOKED_CERT:
${connection.clientCertificate.certificates.leaf.serialNumber}:
${connection.clientCertificate.certificates.leaf.issuer}`);
        console.log(`Denying connection for revoked certificate:
 ${connection.clientCertificate.certificates.leaf.serialNumber}`);
        return connection.deny();
    }

    // Log successful validation for monitoring
    connection.logCustomData(`VALID_CERT:
${connection.clientCertificate.certificates.leaf.serialNumber}`);
    console.log(`Allowing connection for valid certificate:
 ${connection.clientCertificate.certificates.leaf.serialNumber}`);
    return connection.allow();
}
```

Wird immer verwendet async/await , wenn KeyValueStore Methoden oder andere asynchrone Operationen aufgerufen werden. Die Laufzeit der Verbindungsfunktion verarbeitet die Zusagen automatisch und stellt sicher, dass die Ausführung innerhalb der strengen Zeitbeschränkungen der TLS-Handshake-Verarbeitung ordnungsgemäß abläuft. Vermeiden Sie die Verwendung von.then () oder Callback-Mustern, da dies async/await für eine sauberere Fehlerbehandlung und eine bessere Leistung in der Verbindungsfunktionsumgebung sorgt.

Strukturieren Sie Ihren Code beim Entwerfen asynchroner Verbindungsfunktionen so, dass die Anzahl der KeyValueStore Operationen minimiert wird, und führen Sie sie so früh wie möglich in Ihrer Validierungslogik aus. Dadurch wird eine maximale Leistung gewährleistet und das Risiko von Timeout-Problemen in Zeiten mit hohem Datenaufkommen verringert. Erwägen Sie, die zugehörigen Validierungsprüfungen stapelweise durchzuführen und die effizienteste KeyValueStore Methode (exists () oder get ()) für Ihren Anwendungsfall zu verwenden.

Beispiele für Codes für Verbindungsfunktionen

Die folgenden Beispiele veranschaulichen gängige Verbindungsfunktionsmuster für verschiedene Überprüfungsszenarien. Verwenden Sie diese Beispiele als Ausgangspunkt für Ihre eigenen Implementierungen von Verbindungsfunktionen.

Example: Validierung des Gerätezertifikats

In diesem Beispiel werden Geräteseriennummern und Betrefffelder für Zertifikate für IoT-Geräte, Spielekonsolen und andere Client-Authentifizierungsszenarien validiert:

```
async function connectionHandler(connection) {
    // Custom validation: check device serial number format
    const serialNumber = connection.clientCertificate.certificates.leaf.serialNumber;
    if (!serialNumber.startsWith("DEV")) {
        connection.logCustomData(`INVALID_SERIAL:${serialNumber}`);
        return connection.deny();
    }

    // Validate certificate subject contains required organizational unit
    const subject = connection.clientCertificate.certificates.leaf.subject;
    if (!subject.includes("OU=AuthorizedDevices")) {
        connection.logCustomData(`INVALID_OU:${subject}`);
        return connection.deny();
    }

    // Allow connection for valid devices
    connection.logCustomData(`VALID_DEVICE:${serialNumber}`);
    return connection.allow();
}
```

Diese Funktion führt über die standardmäßige Zertifikatsvalidierung hinaus mehrere Validierungsprüfungen durch, einschließlich der Überprüfung des Formats der Geräteseriennummer und der Überprüfung der Organisationseinheit.

Example: Optionale mTLs mit gemischter Authentifizierung

In diesem Beispiel werden sowohl mTLS- als auch Nicht-MTLS-Verbindungen mit unterschiedlichen Authentifizierungsrichtlinien behandelt:

```
async function connectionHandler(connection) {
    if (connection.clientCertificate) {
        // mTLS connection - enhanced validation for certificate holders
        const subject = connection.clientCertificate.certificates.leaf.subject;
        connection.logCustomData(`MTLS_SUCCESS:${subject}:${connection.clientIp}`);
        console.log(`mTLS connection from: ${subject}`);
        return connection.allow();
    } else {
        // Non-mTLS connection - apply IP-based restrictions
        const clientIp = connection.clientIp;

        // Only allow non-mTLS from specific IP ranges
        if (clientIp.startsWith("203.0.113.") || clientIp.startsWith("198.51.100.")) {
            connection.logCustomData(`NON_MTLS_ALLOWED:${clientIp}`);
            console.log(`Non-mTLS connection allowed from: ${clientIp}`);
            return connection.allow();
        }

        connection.logCustomData(`NON_MTLS_DENIED:${clientIp}`);
        return connection.deny();
    }
}
```

Diese Funktion bietet erhöhte Sicherheit für Clients mit Zertifikaten und gewährleistet gleichzeitig die Kompatibilität mit älteren Clients aus vertrauenswürdigen IP-Bereichen.

Testen Sie die CloudFront Verbindungsfunktionen vor der Bereitstellung

Sie können CloudFront Verbindungsfunktionen in der ENTWICKLUNGSPHASE mithilfe der TestConnectionFunction API-Operation testen. Beim Testen können Sie Ihre Funktionslogik anhand von Beispielverbindungsereignissen validieren, bevor Sie sie auf der LIVE-Phase veröffentlichen.

Themen

- Ablauf des Testens
- Testergebnisse
- Objekt zum Testen der Verbindung

Ablauf des Testens

Um eine Verbindungsfunktion zu testen:

1. Erstellen Sie eine Verbindungsfunktion in der ENTWICKLUNGSPHASE
2. Bereiten Sie ein Testverbindungsobjekt vor, das das TLS-Verbindungsereignis darstellt
3. Verwenden Sie die TestConnectionFunction API-Operation, um Ihre Funktion mit den Testdaten auszuführen
4. Überprüfen Sie die Testergebnisse, einschließlich der Funktionsausgabe, der Ausführungsprotokolle und aller Fehlermeldungen
5. Aktualisieren Sie Ihren Funktionscode nach Bedarf und wiederholen Sie den Testvorgang

Testergebnisse

Wenn Sie eine Verbindungsfunktion testen, beinhalten die Ergebnisse:

- Funktionszusammenfassung — Metadaten über die getestete Funktion
- Auslastung berechnen — Leistungskennzahlen, die die Ressourcennutzung anzeigen
- Ausführungsprotokolle — Die Konsolenausgabe Ihrer Funktion, einschließlich aller Protokollierungsanweisungen
- Funktionsausgabe — Das von Ihrer Funktion zurückgegebene Ergebnis
- Fehlermeldungen — Alle Laufzeitfehler oder Ausnahmen, die während der Ausführung aufgetreten sind

Objekt zum Testen der Verbindung

Das Verbindungstestobjekt ist ein binäres Blob (bis zu 40 KB), das das TLS-Verbindungsereignis darstellt, das Ihre Funktion verarbeiten wird. Dieses Objekt enthält das Zertifikat und die Verbindungsinformationen, die Ihre Funktion verwendet, um Authentifizierungsentscheidungen zu treffen.

> **Note**
>
> Die spezifische Struktur und das Format des Verbindungstestobjekts werden durch die Laufzeit der CloudFront Verbindungsfunktionen definiert. Einzelheiten AWS Support zur

Erstellung geeigneter Testobjekte für Ihren Anwendungsfall finden Sie in der Dokumentation zu den CloudFront Funktionen oder wenden Sie sich an uns.

Nachdem Sie Ihre Verbindungsfunktion erstellt haben, können Sie:

- Testen Sie die Funktion — Verwenden Sie die Testfunktionen in der Konsole oder CLI, um Ihre Funktion anhand von Beispielverbindungsereignissen zu validieren. Weitere Informationen finden Sie unter Testen der Verbindungsfunktion.
- Funktion aktualisieren — Ändern Sie den Funktionscode und die Konfiguration nach Bedarf. Verbindungsfunktionen, die sich in der ENTWICKLUNGSPHASE befinden, können jederzeit aktualisiert werden.
- Funktion veröffentlichen — Wenn Sie bereit für die Produktion sind, veröffentlichen Sie die Funktion, um sie von der ENTWICKLUNGS- in die LIVE-Phase zu verschieben. Weitere Informationen finden Sie unter Verbindungsfunktionen zuordnen.
- Einer Verteilung zuordnen — Ordnen Sie die veröffentlichte Funktion einer TLS-enabled M-Verteilung zu, um Direktverbindungen zu verarbeiten. Weitere Informationen finden Sie unter Verbindungsfunktionen zuordnen.

Ordnen Sie Verbindungsfunktionen Verteilungen zu

Nachdem Sie eine Verbindungsfunktion auf der LIVE-Stufe veröffentlicht haben, müssen Sie sie einer TLS-enabled M-Distribution zuordnen, um Live-Verbindungen verarbeiten zu können. Verbindungsfunktionen werden auf Verteilungsebene verknüpft, im Gegensatz zu Viewer-Request- und Viewer-Response-Funktionen, die mit Cache-Verhalten verknüpft sind.

Themen

- Anforderungen an die Zuordnung
- Funktionen mit Filtern organisieren
- Überlegungen zur Bereitstellung

Anforderungen an die Zuordnung

So ordnen Sie einer Verteilung eine Verbindungsfunktion zu:

- Die Funktion muss sich in der LIVE-Phase befinden

- Für die Distribution muss mTLS aktiviert sein
- Für die Verteilung muss ein gültiger Trust Store konfiguriert sein
- Sie können nur eine Verbindungsfunktion pro Verteilung zuordnen

Funktionen mit Filtern organisieren

CloudFront bietet Filterfunktionen, mit denen Sie Verbindungsfunktionen organisieren und verwalten können:

- Distributions-ID-Filter — Findet Funktionen, die bestimmten Verteilungen zugeordnet sind
- Key-value Speicherfilter — Suchen Sie nach Funktionen, die bestimmte Schlüsselwertspeicher für die Datensuche verwenden
- Stufenfilter — Listet Funktionen im DEVELOPMENT- oder LIVE-Stadium auf

Verwenden Sie diese Filter, wenn Sie mehrere Verbindungsfunktionen in verschiedenen Distributionen oder Entwicklungsumgebungen verwalten.

Überlegungen zur Bereitstellung

Berücksichtigen Sie beim Einsatz von Verbindungsfunktionen die folgenden Faktoren:

- Globaler Einsatz — Die Verbindungsfunktionen werden an allen CloudFront Edge-Standorten weltweit bereitgestellt. Dies kann mehrere Minuten dauern
- Versionsverwaltung — Jede veröffentlichte Version erstellt eine neue LIVE-Funktion, die die vorherige Version ersetzt
- Rollback-Strategie — Planen Sie das Rollback, indem Sie frühere Arbeitsversionen Ihres Funktionscodes beibehalten
- Testen in der Produktion — Erwägen Sie, separate Distributionen für Staging- und Produktionsumgebungen zu verwenden

Implementieren Sie den Widerruf von Zertifikaten für gegenseitiges TLS (Viewer) mit Functions und CloudFront KeyValueStore

Sie können CloudFront Connection Functions mit verwenden, KeyValueStore um die Zertifikatssperrüberprüfung zu implementieren. Auf diese Weise können Sie eine Liste der

Seriennummern für widerrufene Zertifikate verwalten und die Client-Zertifikate während des TLS-Handshakes anhand dieser Liste überprüfen.

Um die Sperrung von Zertifikaten zu implementieren, benötigen Sie diese Komponenten:

- Eine mit Viewer-mTLS konfigurierte Distribution
- A, KeyValueStore das die Seriennummern eines widerrufenen Zertifikats enthält
- Eine Verbindungsfunktion, die das abfragt, KeyValueStore um den Zertifikatsstatus zu überprüfen

Wenn ein Client eine Verbindung herstellt, CloudFront validiert er das Zertifikat anhand des Vertrauensspeichers und führt dann Ihre Verbindungsfunktion aus. Ihre Funktion vergleicht die Seriennummer des Zertifikats mit der KeyValueStore und erlaubt oder verweigert die Verbindung.

Themen

- Schritt 1: Erstellen Sie ein Formular KeyValueStore für widerrufene Zertifikate
- Schritt 2: Erstellen Sie die Sperrverbindungsfunktion
- Schritt 3: Testen Sie Ihre Sperrfunktion
- Schritt 4: Ordnen Sie die Funktion Ihrer Distribution zu
- Fortgeschrittene Sperrszenarien

Schritt 1: Erstellen Sie ein Formular KeyValueStore für widerrufene Zertifikate

Erstellen Sie eine KeyValueStore , um die Seriennummern widerrufener Zertifikate zu speichern, die Ihre Verbindungsfunktion bei mTLS-Verbindungen überprüfen kann.

Bereiten Sie zunächst die Seriennummern Ihres widerrufenen Zertifikats im JSON-Format vor:

```
{
  "data": [
    {
      "key": "ABC123DEF456",
      "value": ""
    },
    {
      "key": "789XYZ012GHI",
      "value": ""
    }
  ]
```

```
}
```

Laden Sie diese JSON-Datei in einen S3-Bucket hoch und erstellen Sie dann: KeyValueStore

```
aws s3 cp revoked-serials.json s3://your-bucket-name/revoked-serials.json
aws cloudfront create-key-value-store \
  --name revoked-serials-kvs \
  --import-source '{
    "SourceType": "S3",
    "SourceARN": "arn:aws:s3:::your-bucket-name/revoked-serials.json"
  }'
```

Warten Sie KeyValueStore , bis die Bereitstellung abgeschlossen ist. Überprüfen Sie den Status mit:

```
aws cloudfront get-key-value-store --name "revoked-serials-kvs"
```

Schritt 2: Erstellen Sie die Sperrverbindungsfunktion

Erstellen Sie eine Verbindungsfunktion, die die Seriennummern von Zertifikaten mit den vergleicht KeyValueStore , um festzustellen, ob Zertifikate gesperrt wurden.

Erstellen Sie eine Verbindungsfunktion, die die Seriennummern von Zertifikaten anhand der KeyValueStore folgenden Werte vergleicht:

```
aws cloudfront create-connection-function \
  --name "revocation-control" \
  --connection-function-config file://connection-function-config.json \
  --connection-function-code file://connection-function-code.txt
```

Die Konfigurationsdatei spezifiziert die KeyValueStore Zuordnung:

```
{
  "Runtime": "cloudfront-js-2.0",
  "Comment": "A function that implements revocation control via KVS",
  "KeyValueStoreAssociations": {
    "Quantity": 1,
    "Items": [
      {
        "KeyValueStoreArn": "arn:aws:cloudfront::account-id:key-value-store/kvs-id"
      }
```

```
    ]
  }
}
```

Der Verbindungsfunktionscode überprüft die folgenden Zertifikate KeyValueStore auf widerrufene Zertifikate:

```
import cf from 'cloudfront';

async function connectionHandler(connection) {
    const kvsHandle = cf.kvs();

    // Get parsed client serial number from client certificate
    const clientSerialNumber = connection.clientCertInfo.serialNumber;

    // Check KVS to see if serial number exists as a key
    const serialNumberExistsInKvs = await kvsHandle.exists(clientSerialNumber);

    // Deny connection if serial number exists in KVS
    if (serialNumberExistsInKvs) {
        console.log("Connection denied - certificate revoked");
        return connection.deny();
    }

    // Allow connections that don't exist in kvs
    console.log("Connection allowed");
    return connection.allow();
}
```

Schritt 3: Testen Sie Ihre Sperrfunktion

Verwenden Sie die CloudFront Konsole, um Ihre Verbindungsfunktion mit Beispielzertifikaten zu testen. Navigieren Sie in der Konsole zur Verbindungsfunktion und verwenden Sie die Registerkarte Test.

Testen Sie mit Musterzertifikaten

1. Fügen Sie ein Beispielzertifikat im PEM-Format in die Testoberfläche ein
2. Geben Sie optional eine Client-IP-Adresse zum Testen IP-based der Logik an
3. Wählen Sie Testfunktion, um die Ausführungsergebnisse zu sehen
4. Überprüfen Sie die Ausführungsprotokolle, um Ihre Funktionslogik zu überprüfen

Testen Sie sowohl mit gültigen als auch mit widerrufenen Zertifikaten, um sicherzustellen, dass Ihre Funktion beide Szenarien korrekt verarbeitet. Die Ausführungsprotokolle zeigen die Ausgabe von console.log und alle Fehler, die während der Funktionsausführung auftreten.

Schritt 4: Ordnen Sie die Funktion Ihrer Distribution zu

Nachdem Sie Ihre Verbindungsfunktion veröffentlicht haben, verknüpfen Sie sie mit Ihrer TLS-enabled M-Distribution, um die Zertifikatssperrüberprüfung zu aktivieren.

Sie können die Funktion auf der Seite mit den Verteilungseinstellungen oder in der Tabelle mit den zugehörigen Verteilungen der Verbindungsfunktion verknüpfen. Navigieren Sie zu Ihren Verteilungseinstellungen, scrollen Sie zum Abschnitt Viewer Mutual Authentication (mTLS), wählen Sie Ihre Verbindungsfunktion aus und speichern Sie die Änderungen.

Fortgeschrittene Sperrszenarien

Für komplexere Anforderungen an den Widerruf von Zertifikaten sollten Sie die folgenden zusätzlichen Konfigurationen in Betracht ziehen:

Themen

- Konvertiert Zertifikatssperrlisten (CRL) in ein KeyValueStore Format
- Behandeln Sie mehrere Zertifizierungsstellen
- Fügen Sie benutzerdefinierte Daten zu den Verbindungsprotokollen hinzu
- CRL-Updates verwalten
- Kapazität planen KeyValueStore

Konvertiert Zertifikatssperrlisten (CRL) in ein KeyValueStore Format

Wenn Sie eine CRL-Datei (Certificate Revocation List) haben, können Sie sie mit OpenSSL und jq in das KeyValueStore JSON-Format konvertieren:

CRL in ein Format konvertieren KeyValueStore

Extrahieren Sie Seriennummern aus der CRL-Datei:

```
openssl crl -text -noout -in rfc5280_CRL.crl | \
  awk '/Serial Number:/ {print $3}' | \
  cut -d'=' -f2 | \
```

```
  sed 's/../&:/g;s/:$//' >> serialnumbers.txt
```

Konvertieren Sie die Seriennummern in das KeyValueStore JSON-Format:

```
jq -R -s 'split("\n") | map(select(length > 0)) | {data: map({"key": ., "value": ""})}'
 \
  serialnumbers.txt >> serialnumbers_kvs.json
```

Laden Sie die formatierte Datei in S3 hoch und erstellen Sie KeyValueStore sie wie in Schritt 1 beschrieben.

Behandeln Sie mehrere Zertifizierungsstellen

Wenn Sie mehrere TrustStore Zertifizierungsstellen (CAs) haben, fügen Sie die Ausstellerinformationen in Ihre KeyValueStore Schlüssel ein, um Konflikte zwischen Zertifikaten verschiedener Zertifizierungsstellen zu vermeiden, die möglicherweise dieselbe Seriennummer haben.

Verwenden Sie für Szenarien mit mehreren Zertifizierungsstellen eine Kombination aus dem SHA1-Hash des Ausstellers und der Seriennummer als Schlüssel:

```
import cf from 'cloudfront';

async function connectionHandler(connection) {
    const kvsHandle = cf.kvs();
    const clientCert = connection.clientCertInfo;

    // Create composite key with issuer hash and serial number
    const issuer = clientCert.issuer.replace(/[^a-zA-Z0-9]/g, '').substring(0, 20);
    const serialno = clientCert.serialNumber;
    const compositeKey = `${issuer}_${serialno}`;

    const cert_revoked = await kvsHandle.exists(compositeKey);

    if (cert_revoked) {
        console.log(`Blocking revoked cert: ${serialno} from issuer: ${issuer}`);
        connection.deny();
    } else {
        connection.allow();
    }
}
```

> **Note**
>
> Wenn Sie die Emittentenkennung und die Seriennummer verwenden, werden längere Schlüssel erzeugt, wodurch sich die Gesamtzahl der Einträge, die Sie in der speichern können, verringern könnte. KeyValueStore

Fügen Sie benutzerdefinierte Daten zu den Verbindungsprotokollen hinzu

Verbindungsfunktionen können mithilfe der CustomData Protokollmethode benutzerdefinierte Daten zu CloudFront Verbindungsprotokollen hinzufügen. Auf diese Weise können Sie Ergebnisse der Sperrüberprüfung, Zertifikatsinformationen oder andere relevante Daten in Ihre Protokolle aufnehmen.

```
async function connectionHandler(connection) {
    const kvsHandle = cf.kvs();
    const clientSerialNumber = connection.clientCertInfo.serialNumber;
    const serialNumberExistsInKvs = await kvsHandle.exists(clientSerialNumber);

    if (serialNumberExistsInKvs) {
        // Log revocation details to connection logs
        connection.logCustomData(`REVOKED:${clientSerialNumber}:DENIED`);
        console.log("Connection denied - certificate revoked");
        return connection.deny();
    }

    // Log successful validation
    connection.logCustomData(`VALID:${clientSerialNumber}:ALLOWED`);
    console.log("Connection allowed");
    return connection.allow();
}
```

Benutzerdefinierte Daten sind auf 800 Byte gültigen UTF-8 Textes begrenzt. Wenn Sie diesen Grenzwert überschreiten, werden die Daten bis zur nächsten gültigen UTF-8 Grenze CloudFront gekürzt.

> **Note**
>
> Die benutzerdefinierte Datenprotokollierung funktioniert nur, wenn Verbindungsprotokolle für Ihre Verteilung aktiviert sind. Wenn Verbindungsprotokolle nicht konfiguriert sind, ist die CustomData Protokollmethode ein No-Op.

CRL-Updates verwalten

Zertifizierungsstellen können zwei Arten von CRLs ausstellen:

- Vollständige CRLs: Enthalten eine vollständige Liste aller widerrufenen Zertifikate
- Delta-CRLs: Listet nur Zertifikate auf, die seit der letzten vollständigen CRL gesperrt wurden

Für vollständige CRL-Aktualisierungen erstellen Sie eine neue KeyValueStore mit den aktualisierten Daten und leiten Sie die Verknüpfung der Verbindungsfunktion auf die neue um. KeyValueStore Dieser Ansatz ist einfacher als das Berechnen von Unterschieden und das Durchführen inkrementeller Aktualisierungen.

Verwenden Sie für Delta-CRL-Aktualisierungen den Befehl update-keys, um den vorhandenen Zertifikaten neue widerrufene Zertifikate hinzuzufügen: KeyValueStore

```
aws cloudfront update-key-value-store \
  --name "revoked-serials-kvs" \
  --if-match "current-etag" \
  --put file://delta-revoked-serials.json
```

Kapazität planen KeyValueStore

KeyValueStore hat eine Größenbeschränkung von 5 MB und unterstützt bis zu 10 Millionen Schlüssel-Wert-Paare. Planen Sie die Kapazität Ihrer Sperrliste auf der Grundlage Ihres Schlüsselformats und Ihrer Datengröße:

- Nur Seriennummer: Effizienter Speicher für eine einfache Widerrufsprüfung
- Aussteller-ID und Seriennummer: Längere Schlüssel für Umgebungen mit mehreren Zertifizierungsstellen

Bei großen Sperrlisten sollten Sie erwägen, einen abgestuften Ansatz zu implementieren, bei dem Sie KeyValueStores für verschiedene Zertifikatskategorien oder Zeiträume getrennte Zertifikate verwalten.

Anpassen am Edge mit Lambda@Edge

Lambda @Edge ist eine Erweiterung von. AWS Lambda Lambda @Edge ist ein Rechenservice, mit dem Sie Funktionen ausführen können, die den von Amazon bereitgestellten Inhalt anpassen CloudFront . Sie können Python-Funktionen in der Lambda-Konsole in einer AWS-Region, USA Ost (Nord-Virginia), erstellen. Node.js

Nachdem Sie die Funktion erstellt haben, können Sie mithilfe der Lambda-Konsole oder der CloudFront Lambda-Konsole Trigger hinzufügen, sodass die Funktionen AWS an Orten ausgeführt werden, die sich näher am Betrachter befinden, ohne Server bereitzustellen oder zu verwalten. Optional können Sie Lambda- und CloudFront API-Operationen verwenden, um Ihre Funktionen und Trigger programmgesteuert einzurichten.

Lambda@Edge skaliert automatisch zwischen einigen wenigen Anforderungen pro Tag und ein paar Tausend Anforderungen pro Sekunde. Die Verarbeitung von Anfragen an AWS Orten, die sich näher am Betrachter befinden, anstatt auf Originalservern, reduziert die Latenz erheblich und verbessert die Benutzererfahrung.

> **Note**
>
> Lambda@Edge wird für gRPC-Anforderungen nicht unterstützt. Weitere Informationen finden Sie unter gRPC mit CloudFront Distributionen verwenden.

Themen

- So arbeitet Lambda@Edge mit Anforderungen und Antworten
- Anwendungsmöglichkeiten von Lambda@Edge
- Erste Schritte mit Lambda@Edge-Funktionen (Konsole)
- Einrichten von IAM-Berechtigungen und -Rollen für Lambda@Edge
- Schreiben und Erstellen einer Lambda@Edge-Funktion
- Hinzufügen von Auslösern für eine Lambda@Edge-Funktion
- Testen und Debuggen von Lambda@Edge-Funktionen
- Löschen von Lambda@Edge-Funktionen und Replikaten

- Lambda@Edge-Ereignisstruktur
- Arbeiten mit Anforderungen und Antworten
- Beispielfunktionen für Lambda@Edge

So arbeitet Lambda@Edge mit Anforderungen und Antworten

Wenn Sie eine CloudFront Verteilung mit einer Lambda @Edge -Funktion verknüpfen, CloudFront fängt sie Anfragen und Antworten an CloudFront Edge-Standorten ab. Sie können Lambda-Funktionen ausführen, wenn die folgenden CloudFront Ereignisse eintreten:

- Wann CloudFront erhält er eine Anfrage von einem Zuschauer (Viewer-Anfrage)
- Bevor CloudFront eine Anfrage an den Ursprung weitergeleitet wird (ursprüngliche Anfrage)
- Wann CloudFront erhält er eine Antwort vom Ursprung (ursprüngliche Antwort)
- Before CloudFront gibt die Antwort an den Zuschauer zurück (Antwort des Zuschauers)

Wenn Sie verwenden AWS WAF, wird die Lambda @Edge Viewer-Anfrage ausgeführt, nachdem alle AWS WAF Regeln angewendet wurden.

Weitere Informationen erhalten Sie unter Arbeiten mit Anforderungen und Antworten und Lambda@Edge-Ereignisstruktur.

Anwendungsmöglichkeiten von Lambda@Edge

Es gibt viele Verwendungsmöglichkeiten für die Lambda @Edge -Verarbeitung mit Ihrer CloudFront Amazon-Distribution, wie z. B. die folgenden Beispiele:

- Eine Lambda-Funktion kann Cookies untersuchen und URLs so umschreiben, dass Benutzern verschiedene Versionen einer Website zum A/B Testen angezeigt werden.
- CloudFront kann den Zuschauern je nach verwendetem Gerät unterschiedliche Objekte zurückgeben, indem sie den `User-Agent` Header überprüft, der Informationen zu den Geräten enthält. CloudFront Kann beispielsweise je nach Bildschirmgröße des Geräts unterschiedliche Bilder zurückgeben. In ähnlicher Weise könnte die Funktion den Wert des `Referer` Headers berücksichtigen und veranlassen CloudFront , dass die Bilder an Bots mit der niedrigsten verfügbaren Auflösung zurückgegeben werden.
- Sie können Cookies auch auf andere Kriterien überprüfen. Wenn Sie beispielsweise auf einer Einzelhandels-Website, die Kleidung verkauft, Cookies verwenden, um anzugeben, welche Farbe

ein Benutzer für eine Jacke ausgewählt hat, kann eine Lambda-Funktion die Anfrage so ändern, dass das Bild einer Jacke in der ausgewählten Farbe CloudFront zurückgegeben wird.

- Eine Lambda-Funktion kann HTTP-Antworten generieren, wenn CloudFront Viewer-Anfragen oder Origin-Request-Ereignisse auftreten.
- Eine Funktion kann Header oder Autorisierungstoken überprüfen und einen Header einfügen, um den Zugriff auf Ihre Inhalte zu kontrollieren, bevor die Anfrage an Ihren Ursprung CloudFront weitergeleitet wird.
- Eine Lambda-Funktion kann auch Netzwerkaufrufe an externe Ressourcen durchführen, um die Anmeldeinformationen von Benutzern zu bestätigen oder weitere Inhalte zum Anpassen einer Antwort abzurufen.

Weitere Informationen hierzu einschließlich Beispielcode finden Sie unter Beispielfunktionen für Lambda@Edge.

Weitere Informationen zum Einrichten von Lambda@Edge in der Konsole finden Sie unter Tutorial: Erstellen einer grundlegenden Lambda@Edge-Funktion (Konsole).

Erste Schritte mit Lambda@Edge-Funktionen (Konsole)

Mit Lambda @Edge können Sie CloudFront Trigger verwenden, um eine Lambda-Funktion aufzurufen. Wenn Sie eine CloudFront Verteilung mit einer Lambda-Funktion verknüpfen, CloudFront fängt sie Anfragen und Antworten an CloudFront Edge-Standorten ab und führt die Funktion aus. Lambda-Funktionen können die Sicherheit erhöhen oder Informationen besser an Ihre Viewer anpassen, um die Leistung zu verbessern.

Die folgende Liste bietet einen grundlegenden Überblick darüber, wie Lambda-Funktionen mit CloudFront erstellt und verwendet werden.

Überblick: Lambda-Funktionen erstellen und verwenden mit CloudFront

1. Erstellen Sie eine Lambda-Funktion in der Region USA Ost (Nord-Virginia).
2. Speichern und veröffentlichen Sie eine nummerierte Version der Funktion.

   Wenn Sie die Funktion ändern möchten, müssen Sie die $LATEST-Version der Funktion in der Region USA Ost (Nord-Virginia) bearbeiten. Bevor Sie es für die Verwendung einrichten CloudFront, veröffentlichen Sie dann eine neue nummerierte Version.
3. Ordnen Sie der Funktion ein CloudFront Verteilungs- und Cache-Verhalten zu. Geben Sie dann ein oder mehrere CloudFront Ereignisse (Trigger) an, die die Ausführung der Funktion veranlassen.

Sie können beispielsweise einen Trigger erstellen, damit die Funktion ausgeführt wird, wenn sie eine Anfrage von einem Viewer CloudFront erhält.

4. Wenn Sie einen Auslöser erstellen, erstellt Lambda Replikate der Funktion an AWS -Standorten weltweit.

> **Tip**
>
> Weitere Informationen finden Sie unter Funktionen erstellen und aktualisieren, Ereignisstruktur und Hinzufügen von CloudFront Triggern. Weitere Ideen und Codebeispiele finden Sie auch in Beispielfunktionen für Lambda@Edge.

Eine schrittweise Anleitung finden Sie unter dem folgenden Thema:

Themen

- Tutorial: Erstellen einer grundlegenden Lambda@Edge-Funktion (Konsole)

Tutorial: Erstellen einer grundlegenden Lambda@Edge-Funktion (Konsole)

Dieses Tutorial zeigt Ihnen, wie Sie mit Lambda @Edge beginnen können, indem Sie eine Node.js Beispielfunktion erstellen und konfigurieren, die in CloudFront ausgeführt wird. In diesem Beispiel werden einer Antwort beim CloudFront Abrufen einer Datei HTTP-Sicherheitsheader hinzugefügt. (Dies kann die Sicherheit und den Datenschutz einer Website verbessern.)

Sie benötigen keine eigene Website für dieses Tutorial. Wenn Sie jedoch Ihre eigene Lambda@Edge-Lösung erstellen möchten, führen Sie ähnliche Schritte aus und wählen aus denselben Optionen aus.

Themen

- Schritt 1: Melden Sie sich für eine an AWS-Konto
- Schritt 2: Erstellen Sie eine CloudFront Distribution
- Schritt 3: Erstellen Ihrer Funktion
- Schritt 4: Fügen Sie einen CloudFront Trigger hinzu, um die Funktion auszuführen
- Schritt 5: Überprüfen, ob die Funktion funktioniert
- Schritt 6: Beheben von Problemen
- Schritt 7: Bereinigen der Ressourcen für Ihr Beispiel

- Ähnliche Informationen

Schritt 1: Melden Sie sich für eine an AWS-Konto

Registrieren Sie sich für ein AWS-Konto, falls Sie das noch nicht getan haben. Weitere Informationen finden Sie unter Melden Sie sich für eine an AWS-Konto.

Schritt 2: Erstellen Sie eine CloudFront Distribution

Bevor Sie die Lambda @Edge -Beispielfunktion erstellen, müssen Sie über eine CloudFront Umgebung verfügen, mit der Sie arbeiten können und die einen Ursprung für die Bereitstellung von Inhalten enthält.

In diesem Beispiel erstellen Sie eine CloudFront Distribution, die einen Amazon S3 S3-Bucket als Ursprung für die Verteilung verwendet. Wenn Sie bereits über eine Umgebung verfügen, die Sie benutzen können, können Sie diesen Schritt überspringen.

Um eine CloudFront Distribution mit einem Amazon S3 S3-Ursprung zu erstellen

1. Erstellen Sie einen Amazon S3-Bucket mit einer oder zwei Dateien, z. B. Image-Dateien, als Beispielinhalt. Für Hilfe folgen Sie den Schritten unter Hochladen Ihrer Inhalte zu Amazon S3. Stellen Sie sicher, dass Sie Berechtigungen erteilen, um öffentliche Lesezugriff auf die Objekte in Ihrem Bucket zu gewähren.
2. Erstellen Sie eine CloudFront Distribution und fügen Sie Ihren S3-Bucket als Ursprung hinzu, indem Sie die Schritte CloudFront unter Web-Distribution erstellen befolgen. Wenn Sie bereits eine Verteilung haben, können Sie stattdessen den Bucket als Ursprung für diese Verteilung hinzufügen.

   > **Tip**
   >
   > Notieren Sie sich die ID Ihrer Verteilung. Wenn Sie später in diesem Tutorial einen CloudFront Trigger für Ihre Funktion hinzufügen, müssen Sie die ID für Ihre Distribution in einer Dropdownliste auswählen — zum Beispiel. `E653W22221KDDL`

Schritt 3: Erstellen Ihrer Funktion

In diesem Schritt erstellen Sie eine Lambda-Funktion anhand einer Vorlage in der Lambda-Konsole. Die Funktion fügt Code hinzu, um Sicherheitsheader in Ihrer CloudFront-Verteilung zu aktualisieren.

So erstellen Sie eine Lambda-Funktion

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die AWS Lambda Konsole unter. https://console.aws.amazon.com/lambda/

   > **Important**
   >
   > Stellen Sie sicher, dass Sie sich in US-East-1 (Nord-Virginia) AWS-Region (US-East-1) befinden. Sie müssen sich in dieser Region befinden, um Lambda@Edge-Funktionen erstellen zu können.

2. Wählen Sie Funktion erstellen.
3. Wählen Sie auf der Seite Funktion erstellen die Option Blueprint verwenden aus, und filtern Sie dann nach den CloudFront Blueprints, indem Sie sie **`cloudfront`** in das Suchfeld eingeben.

   > **Note**
   >
   > CloudFront Blueprints sind nur in der Region US-East-1 (Nord-Virginia) (us-east-1) verfügbar.

4. Wählen Sie die Vorlage HTTP-Antwortheader ändern als Vorlage für Ihre Funktion.
5. Geben Sie folgende Informationen zu Ihrer Funktion ein:

   - Funktionsname – Geben Sie einen Namen für Ihre Funktion ein.
   - Ausführungsrolle – Wählen Sie, wie Sie die Berechtigungen für Ihre Funktion festlegen. Um die empfohlene Standardvorlage für Lambda @Edge -Berechtigungsrichtlinien zu verwenden, wählen Sie Neue Rolle aus AWS Richtlinienvorlagen erstellen aus.
   - Rollenname – Geben Sie einen Namen für die Rolle ein, die von der Richtlinienvorlage erstellt wird.
   - Richtlinienvorlagen — Lambda fügt automatisch die Richtlinienvorlage Basic Lambda @Edge -Berechtigungen hinzu, da Sie einen CloudFront Blueprint als Grundlage für Ihre Funktion ausgewählt haben. Diese Richtlinienvorlage fügt Ausführungsrollenberechtigungen hinzu, mit denen CloudFront Sie Ihre Lambda-Funktion CloudFront an Standorten auf der ganzen Welt für Sie ausführen können. Weitere Informationen finden Sie unter Einrichten von IAM-Berechtigungen und -Rollen für Lambda@Edge.

6. Wählen Sie unten auf der Seite Funktion erstellen aus.

7. Wählen Sie im daraufhin angezeigten Bereich In Lambda@Edge bereitstellen die Option Abbrechen aus. (Für dieses Tutorial müssen Sie den Funktionscode ändern, bevor Sie die Funktion in Lambda@Edge bereitstellen.)
8. Scrollen Sie nach unten bis zum Abschnitt Code-Quelle.
9. Ersetzen Sie den Vorlagencode durch eine Funktion, die die Sicherheits-Header ändert, die Ihren Ursprung zurückgibt. Beispielsweise könnten Sie Code wie den folgenden verwenden:

```
'use strict';
export const handler = (event, context, callback) => {

    //Get contents of response
    const response = event.Records[0].cf.response;
    const headers = response.headers;

    //Set new headers
    headers['strict-transport-security'] = [{key: 'Strict-Transport-Security',
 value: 'max-age= 63072000; includeSubdomains; preload'}];
    headers['content-security-policy'] = [{key: 'Content-Security-Policy', value:
 "default-src 'none'; img-src 'self'; script-src 'self'; style-src 'self'; object-
src 'none'"}];
    headers['x-content-type-options'] = [{key: 'X-Content-Type-Options', value:
 'nosniff'}];
    headers['x-frame-options'] = [{key: 'X-Frame-Options', value: 'DENY'}];
    headers['x-xss-protection'] = [{key: 'X-XSS-Protection', value: '1;
 mode=block'}];
    headers['referrer-policy'] = [{key: 'Referrer-Policy', value: 'same-origin'}];

    //Return modified response
    callback(null, response);
};
```

10. Wählen Sie Datei und dann Speichern aus, um den aktualisierten Code zu speichern.
11. Wählen Sie Bereitstellen.

Fahren Sie mit dem nächsten Abschnitt fort, um einen CloudFront Trigger zum Ausführen der Funktion hinzuzufügen.

Schritt 4: Fügen Sie einen CloudFront Trigger hinzu, um die Funktion auszuführen

Da Sie nun über eine Lambda-Funktion zum Aktualisieren von Sicherheitsheadern verfügen, konfigurieren Sie den CloudFront Trigger so, dass Ihre Funktion so ausgeführt wird, dass die Header

zu jeder Antwort hinzugefügt werden, die vom Ursprung Ihrer Distribution CloudFront empfangen wird.

Um den CloudFront Trigger für Ihre Funktion zu konfigurieren

1. Wählen Sie in der Lambda-Konsole auf der Seite Funktionsübersicht Ihrer Funktion die Option Auslöser hinzufügen aus.
2. Wählen Sie für die Trigger-Konfiguration CloudFront.
3. Wählen Sie In Lambda@Edge bereitstellen aus.
4. Geben Sie im Bereich Deploy to Lambda @Edge unter Configure CloudFront trigger die folgenden Informationen ein:

   - Distribution — Die CloudFront Distribution-ID, die Ihrer Funktion zugeordnet werden soll. Wählen Sie in der Dropdown-Liste die Distributions-ID aus.
   - Cacheverhalten – Das Cacheverhalten, das für den Auslöser verwendet werden soll. Behalten Sie in diesem Beispiel den Wert * bei, der das Standard-Cache-Verhalten Ihrer Verteilung bezeichnet. Weitere Informationen finden Sie unter Einstellungen für das Cache-Verhalten im Thema Referenz für alle Distributionseinstellungen.
   - CloudFront event — Der Trigger, der angibt, wann Ihre Funktion ausgeführt wird. Die Security-Header-Funktion sollte immer dann ausgeführt werden, wenn eine Antwort vom Ursprung CloudFront zurückgegeben wird. Wählen Sie in der Dropdown-Liste Ursprungsantwort aus. Weitere Informationen finden Sie unter Hinzufügen von Auslösern für eine Lambda@Edge-Funktion.
5. Aktivieren Sie das Kontrollkästchen Bereitstellung in Lambda@Edge bestätigen.
6. Wählen Sie Deploy, um den Trigger hinzuzufügen und die Funktion AWS an Standorten weltweit zu replizieren.
7. Warten Sie, bis die Funktion repliziert wurde. Dies dauert in der Regel mehrere Minuten.

   Sie können überprüfen, ob die Replikation abgeschlossen ist, indem Sie die CloudFront-Konsole öffnen und sich Ihre Verteilung ansehen. Warten Sie darauf, dass sich der Status der Distribution von Wird bereitgestellt in ein Datum und eine Uhrzeit ändert, was bedeutet, dass Ihre Funktion repliziert wurde. Zur Überprüfung der Funktionstätigkeit befolgen Sie die Schritte im nächsten Abschnitt.

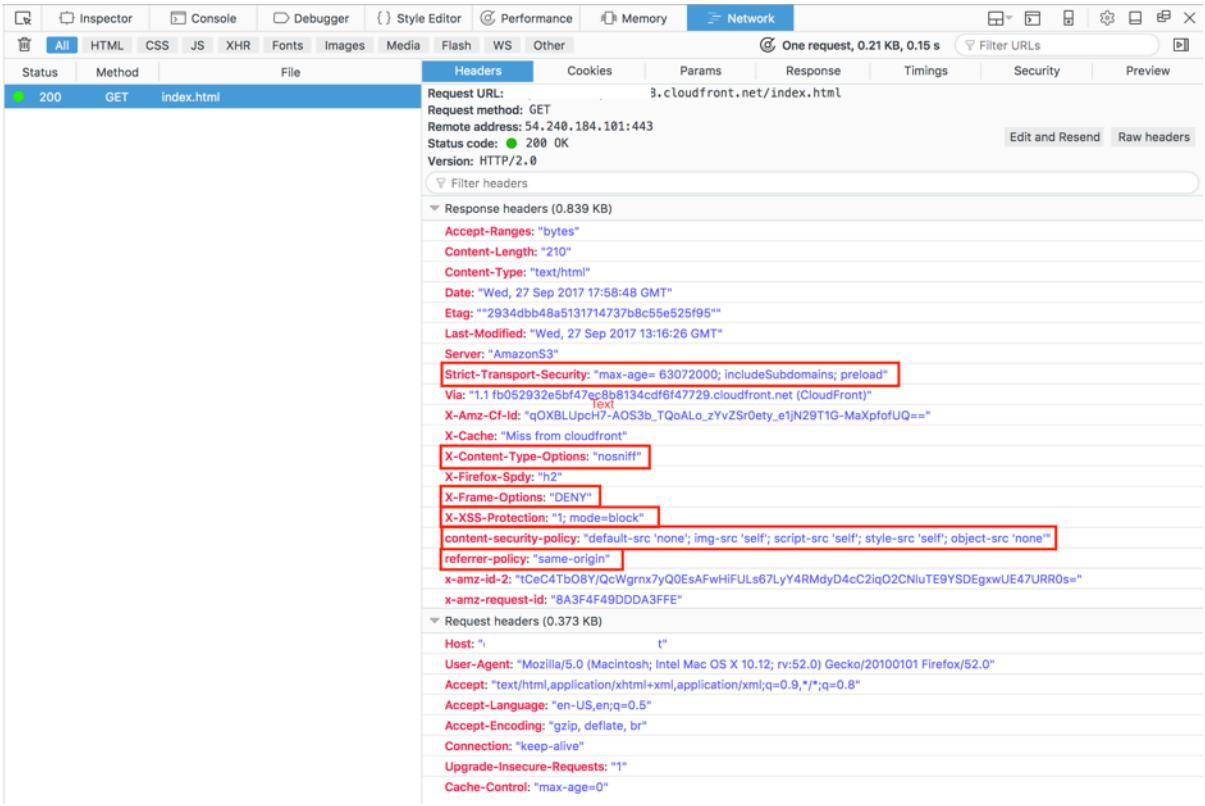
Schritt 5: Überprüfen, ob die Funktion funktioniert

Nachdem Sie Ihre Lambda-Funktion erstellt und einen Trigger konfiguriert haben, um sie für eine CloudFront Distribution auszuführen, überprüfen Sie, ob die Funktion das leistet, was Sie von ihr erwarten. In diesem Beispiel überprüfen Sie die HTTP-Header, die CloudFront zurückgegeben werden, um sicherzustellen, dass die Sicherheitsheader hinzugefügt wurden.

Überprüfen, ob Ihre Lambda@Edge-Funktion Sicherheitsheader hinzufügt

1. Geben Sie in einem Browser die URL für eine Datei in Ihrem S3-Bucket ein.
   Sie können beispielsweise eine URL wie die folgende Verwenden: `https://d111111abcdef8.cloudfront.net/image.jpg`.

   Weitere Informationen zum CloudFront Domainnamen, der in der Datei-URL verwendet werden soll, finden Sie unter. Passen Sie das URL-Format für Dateien in an CloudFront
2. Öffnen Sie die Web-Developer-Symbolleiste Ihres Browsers. Öffnen Sie beispielsweise in Ihrem Browserfenster in Chrome das Kontextmenü und wählen Sie dann Inspect.
3. Wählen Sie die Registerkarte Network (Netzwerk) aus.
4. Laden Sie die Seite, um Ihr Image anzuzeigen, und wählen Sie dann auf der linken Seite eine HTTP-Anforderung. Sie sehen die HTTP-Header in einem separaten Fenster.
5. Sehen Sie sich die Liste der HTTP-Header an, um zu überprüfen, ob die erwarteten Sicherheitsheader in der Liste enthalten sind. Beispielsweise könnten Sie Header wie im folgenden Screenshot gezeigt sehen.



Wenn die Sicherheitsheader in Ihrem Header-Liste enthalten sind, dann ist das hervorragend! Sie haben erfolgreich Ihre erste Lambda@Edge-Funktion erstellt. Wenn CloudFront Rückgabefehler auftreten oder es andere Probleme gibt, fahren Sie mit dem nächsten Schritt fort, um die Probleme zu beheben.

Schritt 6: Beheben von Problemen

Wenn Fehler CloudFront zurückgegeben werden oder die Sicherheitsheader nicht wie erwartet hinzugefügt werden, können Sie die Ausführung Ihrer Funktion anhand von CloudWatch Protokollen untersuchen. Achten Sie darauf, die Protokolle zu verwenden, die an dem AWS Ort gespeichert sind, der dem Ort, an dem die Funktion ausgeführt wird, am nächsten liegt.

Wenn Sie sich die Datei beispielsweise von London aus ansehen, versuchen Sie, die Region in der CloudWatch Konsole auf Europa (London) zu ändern.

Um CloudWatch Logs für Ihre Lambda @Edge -Funktion zu untersuchen

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.

2. Ändern Sie Region auf den Speicherort, der angezeigt wird, wenn Sie die Datei in Ihrem Browser anzeigen. Hier wird die Funktion ausgeführt.
3. Klicken Sie im linken Bereich auf Logs (Protokolle), um die Protokolle für Ihre Verteilung anzuzeigen.

Weitere Informationen finden Sie unter Überwachen Sie CloudFront Metriken mit Amazon CloudWatch.

Schritt 7: Bereinigen der Ressourcen für Ihr Beispiel

Wenn Sie einen Amazon S3 S3-Bucket und eine CloudFront Distribution nur für dieses Tutorial erstellt haben, löschen Sie die AWS Ressourcen, die Sie zugewiesen haben, sodass keine Gebühren mehr anfallen. Nachdem Sie Ihre AWS Ressourcen gelöscht haben, sind alle Inhalte, die Sie hinzugefügt haben, nicht mehr verfügbar.

Aufgaben

- Löschen des S3-Buckets
- Löschen Sie die Lambda-Funktion
- Löschen Sie die CloudFront Distribution

Löschen des S3-Buckets

Bevor Sie Ihren Amazon S3-Bucket löschen, stellen Sie sicher, dass die Protokollierung für den Bucket deaktiviert ist. Andernfalls werden AWS weiterhin Logs in Ihren Bucket geschrieben, während Sie ihn löschen.

Deaktivieren Sie die Protokollierung für einen Bucket wie folgt:

1. Öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie den Bucket aus, und wählen Sie dann Properties (Eigenschaften).
3. Wählen Sie unter Properties (Eigenschaften) Logging (Protokollierung) aus.
4. Deaktivieren Sie das Kontrollkästchen Enabled (Aktiviert).
5. Wählen Sie Save (Speichern) aus.

Jetzt können Sie Ihren Bucket löschen. Weitere Informationen erhalten Sie unter Löschen eines Buckets im Amazon Simple Storage Service-Konsolenbenutzerhandbuch.

Löschen Sie die Lambda-Funktion

Anleitungen zum Löschen der Zuweisung der Lambda-Funktion und optional der Funktion selbst finden Sie unter Löschen von Lambda@Edge-Funktionen und Replikaten.

Löschen Sie die CloudFront Distribution

Bevor Sie eine CloudFront Distribution löschen, müssen Sie sie deaktivieren. Eine deaktivierte Verteilung funktioniert nicht mehr und es fallen keine weiteren Kosten für sie an. Sie können eine deaktivierte Verteilung jederzeit wieder aktivieren. Nachdem Sie eine deaktivierte Verteilung gelöscht haben, ist sie nicht länger verfügbar.

Um eine CloudFront Distribution zu deaktivieren und zu löschen

1. Öffnen Sie die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home.
2. Klicken Sie mit der rechten Maustaste auf die Verteilung, die Sie deaktivieren möchten, und anschließend auf Disable (Deaktivieren).
3. Wenn Sie zur Bestätigung aufgefordert werden, wählen Sie Yes, Disable (Ja, deaktivieren).
4. Wählen Sie die deaktivierte Verteilung aus, und klicken Sie dann auf Delete (Löschen).
5. Wenn Sie zur Bestätigung aufgefordert werden, wählen Sie Yes, Delete.

Ähnliche Informationen

Nun, da Sie eine grundlegende Vorstellung davon haben, wie Lambda@Edge-Funktionen funktionieren, können Sie hier weitere Informationen erhalten:

- Beispielfunktionen für Lambda@Edge
- Bewährte Methoden für das Lambda @Edge -Design
- Reduzierung der Latenz und Verlagerung von Rechenleistung an den Edge mit Lambda @Edge

Einrichten von IAM-Berechtigungen und -Rollen für Lambda@Edge

Um Lambda@Edge zu konfigurieren, benötigen Sie die folgenden IAM-Berechtigungen und -Rollen für AWS Lambda:

- IAM-Berechtigungen — Mit diesen Berechtigungen können Sie Ihre Lambda-Funktion erstellen und sie Ihrer CloudFront Distribution zuordnen.

- Eine Ausführungsrolle für die Lambda-Funktion (IAM-Rolle) – Die Lambda-Service-Prinzipale übernehmen diese Rolle, um die Funktion auszuführen.
- Service-linked Rollen für Lambda @Edge — Die dienstverknüpften Rollen ermöglichen es bestimmten AWS-Services , Lambda-Funktionen in Protokolldateien zu replizieren AWS-Regionen und deren Verwendung CloudWatch zu ermöglichen. CloudFront

IAM-Berechtigungen sind erforderlich, um Lambda @Edge -Funktionen mit Distributionen zu verknüpfen CloudFront

Zusätzlich zu den IAM-Berechtigungen, die Sie für Lambda benötigen, benötigen Sie die folgenden Berechtigungen, um Lambda-Funktionen Distributionen zuzuordnen: CloudFront

- `lambda:GetFunction` – Gewährt die Berechtigung, Konfigurationsinformationen für die Lambda-Funktion und eine vorsignierte URL zum Herunterladen einer `.zip`-Datei abzurufen, die die Funktion enthält.
- `lambda:EnableReplication*` – Gewährt die Berechtigung für die Ressourcenrichtlinie, sodass der Lambda-Replikationsservice den Code und die Konfiguration der Funktion abrufen kann.
- `lambda:DisableReplication*` – Gewährt die Berechtigung für die Ressourcenrichtlinie, sodass der Lambda-Replikationsservice die Funktion löschen kann.

> **Important**
>
> Sie müssen das Sternchen (`*`) am Ende der Aktionen `lambda:EnableReplication*` und `lambda:DisableReplication*` hinzufügen.

- Geben Sie für die Ressource den ARN der Funktionsversion an, die Sie ausführen möchten, wenn ein CloudFront Ereignis eintritt, wie im folgenden Beispiel:

  `arn:aws:lambda:us-east-1:123456789012:function:TestFunction:2`

- `iam:CreateServiceLinkedRole`— Erteilt die Erlaubnis, eine serviceverknüpfte Rolle zu erstellen, in der Lambda @Edge Lambda-Funktionen repliziert. CloudFront Nachdem Sie Lambda@Edge zum ersten Mal konfiguriert haben, wird die serviceverknüpfte Rolle automatisch für Sie erstellt. Sie müssen diese Berechtigung nicht zu anderen Distributionen hinzufügen, die Lambda@Edge verwenden.

- `cloudfront:UpdateDistribution` oder `cloudfront:CreateDistribution` – Gewährt die Berechtigung, eine Distribution zu aktualisieren oder zu erstellen.

Weitere Informationen finden Sie unter den folgenden Themen:

- Identitäts- und Zugriffsmanagement für Amazon CloudFront
- Zugriffsberechtigungen für Lambda-Ressourcen im Entwicklerhandbuch für AWS Lambda

Funktionsausführungsrolle für Service-Prinzipale

Sie müssen eine IAM-Rolle erstellen, die die Service-Prinzipale `lambda.amazonaws.com` und `edgelambda.amazonaws.com` übernehmen können, wenn sie Ihre Funktion ausführen.

> **Tip**
>
> Wenn Sie Ihre Funktion in der Lambda-Konsole erstellen, können Sie wählen, ob Sie mithilfe einer AWS Richtlinienvorlage eine neue Ausführungsrolle erstellen möchten. Dieser Schritt fügt automatisch die erforderlichen Lambda@Edge-Berechtigungen hinzu, um Ihre Funktion auszuführen. Siehe Schritt 5 im Tutorial: Erstellen einer einfachen Lambda@Edge-Funktion.

Weitere Informationen zum manuellen Erstellen einer IAM-Rolle finden Sie unter Erstellen von Rollen und Anfügen von Richtlinien (Konsole) im IAM-Benutzerhandbuch.

Example Beispiel: Rollenvertrauensrichtlinie

Diese Rolle können Sie unter der Registerkarte Vertrauensstellung in der IAM-Konsole hinzufügen. Fügen Sie diese Richtlinie nicht unter der Registerkarte Berechtigungen hinzu.

JSON

```
{
   "Version":"2012-10-17",
   "Statement": [
      {
         "Effect": "Allow",
         "Principal": {
            "Service": [
               "lambda.amazonaws.com",
               "edgelambda.amazonaws.com"
            ]
         },
```

```
            "Action": "sts:AssumeRole"
        }
    ]
}
```

Weitere Informationen zu den Berechtigungen, die Sie der Ausführungsrolle erteilen müssen, finden Sie unter Zugriffsberechtigungen für Lambda-Ressourcen im Entwicklerhandbuch für AWS Lambda .

Hinweise

- Standardmäßig werden Daten in CloudWatch Logs geschrieben, wenn ein CloudFront Ereignis eine Lambda-Funktion auslöst. Wenn Sie diese Protokolle verwenden möchten, benötigt die Ausführungsrolle die Berechtigung, Daten in CloudWatch Logs zu schreiben. Sie können die vordefinierte AWSLambdaBasicExecutionRole verwenden, um der Ausführungsrolle die Berechtigung zu erteilen.

  Weitere Hinweise zu CloudWatch Protokollen finden Sie unterthe section called “Protokolle für Edge-Funktionen”.
- Wenn Ihr Lambda-Funktionscode auf andere AWS Ressourcen zugreift, z. B. auf das Lesen eines Objekts aus einem S3-Bucket, benötigt die Ausführungsrolle die Erlaubnis, diese Aktion auszuführen.

Service-linked Rollen für Lambda @Edge

Lambda@Edge verwendet serviceverknüpfte IAM-Rollen. Eine serviceverknüpfte Rolle ist eine einzigartige Art von IAM-Rolle, die direkt mit einem Dienst verknüpft ist. Service-linked Rollen sind vom Dienst vordefiniert und beinhalten alle Berechtigungen, die der Dienst benötigt, um andere AWS Dienste in Ihrem Namen aufzurufen.

Lambda@Edge verwendet die folgenden serviceverknüpften IAM-Rollen:

- AWSServiceRoleForLambdaReplicator – Lambda@Edge verwendet diese Rolle, um es Lambda@Edge zu ermöglichen, Funktionen in AWS-Regionen zu replizieren.

  Wenn Sie zum ersten Mal einen Lambda @Edge -Trigger hinzufügen CloudFront, AWSServiceRoleForLambdaReplicator wird automatisch eine Rolle mit dem Namen erstellt, damit Lambda @Edge Funktionen replizieren kann. AWS-Regionen Diese Rolle

ist für die Verwendung von Lambda@Edge-Funktionen erforderlich. Die ARN für die Rolle AWSServiceRoleForLambdaReplicator sieht wie im folgenden Beispiel aus:

```
arn:aws:iam::123456789012:role/aws-service-role/
replicator.lambda.amazonaws.com/AWSServiceRoleForLambdaReplicator
```

- AWSServiceRoleForCloudFrontLogger— CloudFront verwendet diese Rolle, um Protokolldateien zu übertragen. CloudWatch Sie können Protokolldateien verwenden, um Lambda@Edge-Validierungsfehler zu beheben.

  Die AWSServiceRoleForCloudFrontLogger Rolle wird automatisch erstellt, wenn Sie eine Lambda @Edge -Funktionsassoziation hinzufügen, an die Lambda @Edge -Fehlerprotokolldateien übertragen werden können CloudFront . CloudWatch Der ARN für die AWSServiceRoleForCloudFrontLogger-Rolle sieht so aus:

  ```
  arn:aws:iam::account_number:role/aws-service-role/
  logger.cloudfront.amazonaws.com/AWSServiceRoleForCloudFrontLogger
  ```

Eine serviceverknüpfte Rolle vereinfacht das Einrichten und Verwenden von Lambda@Edge, da Sie die erforderlichen Berechtigungen nicht manuell hinzufügen müssen. Lambda@Edge definiert die Berechtigungen seiner servicegebundenen Rollen. Nur Lambda@Edge kann die Rollen übernehmen. Die definierten Berechtigungen umfassen die Vertrauens- und Berechtigungsrichtlinie. Sie können die Berechtigungsrichtlinie keiner anderen IAM-Entität zuordnen.

Sie müssen alle zugehörigen Ressourcen CloudFront oder Lambda @Edge -Ressourcen entfernen, bevor Sie eine dienstverknüpfte Rolle löschen können. Dies trägt zum Schutz Ihrer Lambda@Edge-Ressourcen bei, sodass Sie keine serviceverknüpfte Rolle entfernen, die noch für den Zugriff auf aktive Ressourcen erforderlich ist.

Weitere Informationen zu serviceverknüpften Rollen finden Sie unter Service-linked Rollen für CloudFront.

Service-linked Rollenberechtigungen für Lambda @Edge

Lambda@Edge verwendet zwei servicegebundene Rollen. Diese heißen AWSServiceRoleForLambdaReplicator und AWSServiceRoleForCloudFrontLogger. In den folgenden Abschnitten werden die Berechtigungen für jede dieser Rollen beschrieben.

Inhalt

- Service-linked Rollenberechtigungen für Lambda-Replikator

- Service-linked Rollenberechtigungen für Logger CloudFront

Service-linked Rollenberechtigungen für Lambda-Replikator

Diese serviceverknüpfte Rolle ermöglicht Lambda das Replizieren von Lambda@Edge-Funktionen zu AWS-Regionen.

Die serviceverknüpfte Rolle AWSServiceRoleForLambdaReplicator vertraut dem Service `replicator.lambda.amazonaws.com`, sodass dieser die Rolle annehmen kann.

Die Rollenberechtigungsrichtlinie erlaubt Lambda@Edge die Durchführung der folgenden Aktionen für die angegebenen Ressourcen:

- `lambda:CreateFunction` auf `arn:aws:lambda:*:*:function:*`
- `lambda:DeleteFunction` auf `arn:aws:lambda:*:*:function:*`
- `lambda:DisableReplication` auf `arn:aws:lambda:*:*:function:*`
- `iam:PassRole` auf `all AWS resources`
- `cloudfront:ListDistributionsByLambdaFunction` auf `all AWS resources`

Service-linked Rollenberechtigungen für Logger CloudFront

Diese dienstbezogene Rolle ermöglicht das CloudFront Pushen von Protokolldateien, CloudWatch sodass Sie Lambda @Edge -Validierungsfehler debuggen können.

Die serviceverknüpfte Rolle AWSServiceRoleForCloudFrontLogger vertraut dem Service `logger.cloudfront.amazonaws.com`, sodass dieser die Rolle annehmen kann.

Die Rollenberechtigungsrichtlinie erlaubt Lambda@Edge die Durchführung der folgenden Aktionen für die angegebene `arn:aws:logs:*:*:log-group:/aws/cloudfront/*`-Ressource:

- `logs:CreateLogGroup`
- `logs:CreateLogStream`
- `logs:PutLogEvents`

Sie müssen Berechtigungen konfigurieren, damit eine IAM-Entität (z. B. ein Benutzer, eine Gruppe oder eine Rolle) die mit dem Lambda@Edge-Service verknüpften Rollen löschen kann. Weitere Informationen finden Sie unter Service-linked Rollenberechtigungen im IAM-Benutzerhandbuch.

Serviceverknüpfte Rollen für Lambda@Edge erstellen

Servicegebundene Rollen für Lambda@Edge werden in der Regel nicht manuell erstellt. In den folgenden Szenarien legt der Service die Rollen für Sie automatisch an:

- Wenn Sie zum ersten Mal einen Auslöser erstellen, erstellt der Service die Rolle AWSServiceRoleForLambdaReplicator (sofern sie nicht bereits vorhanden ist). Diese Rolle ermöglicht Lambda das Replizieren von Lambda@Edge-Funktionen in AWS-Regionen.

  Wenn Sie die serviceverknüpfte Rolle löschen, wird die Rolle erneut erstellt, wenn Sie einen neuen Auslöser für Lambda@Edge in einer Verteilung hinzufügen.
- Wenn Sie eine CloudFront Distribution aktualisieren oder erstellen, die über eine Lambda @Edge -Zuordnung verfügt, erstellt der Dienst die AWSServiceRoleForCloudFrontLogger Rolle (sofern die Rolle noch nicht vorhanden ist). Diese Rolle ermöglicht es CloudFront , Ihre Protokolldateien per Push zu CloudWatch übertragen.

  Wenn Sie die dienstverknüpfte Rolle löschen, wird die Rolle erneut erstellt, wenn Sie eine CloudFront Distribution aktualisieren oder erstellen, die über eine Lambda @Edge -Zuordnung verfügt.

Um diese dienstbezogenen Rollen manuell zu erstellen, können Sie die folgenden Befehle AWS Command Line Interface ()AWS CLI ausführen:

Um die zu erstellen AWSServiceRoleForLambdaReplicator role

- Führen Sie den folgenden Befehl aus.

```
aws iam create-service-linked-role --aws-service-name
 replicator.lambda.amazonaws.com
```

Um das zu erstellen AWSServiceRoleForCloudFrontLogger role

- Führen Sie den folgenden Befehl aus.

```
aws iam create-service-linked-role --aws-service-name
 logger.cloudfront.amazonaws.com
```

Bearbeiten von serviceverknüpften Lambda@Edge-Rollen

Lambda@Edge verhindert die Bearbeitung der serviceverknüpften Rollen AWSServiceRoleForLambdaReplicator und AWSServiceRoleForCloudFrontLogger. Da möglicherweise verschiedene Entitäten auf die Rolle verweisen, kann der Rollenname nach der Erstellung einer serviceverknüpften Rolle durch den Service nicht bearbeitet werden. Sie können mithilfe von IAM jedoch die Beschreibung der Rolle bearbeiten. Weitere Informationen finden Sie unter Bearbeiten einer serviceverknüpften Rolle im IAM-Benutzerhandbuch.

Unterstützt AWS-Regionen für dienstverknüpfte Lambda @Edge -Rollen

CloudFront unterstützt die Verwendung von dienstverknüpften Rollen für Lambda @Edge im Folgenden: AWS-Regionen

- USA Ost (Nord-Virginia) – `us-east-1`
- USA Ost (Ohio) – `us-east-2`
- USA West (Nordkalifornien) – `us-west-1`
- USA West (Oregon) – `us-west-2`
- Asia Pacific (Mumbai) – `ap-south-1`
- Asien-Pazifik (Seoul) – `ap-northeast-2`
- Asia Pacific (Singapore) – `ap-southeast-1`
- Asien-Pazifik (Sydney) – `ap-southeast-2`
- Asien-Pazifik (Tokio) – `ap-northeast-1`
- Europa (Frankfurt) – `eu-central-1`
- Europa (Ireland) – `eu-west-1`
- Europa (London) – `eu-west-2`
- South America (São Paulo) – `sa-east-1`

Schreiben und Erstellen einer Lambda@Edge-Funktion

Um Lambda@Edge zu verwenden, schreiben Sie den Code für Ihre AWS Lambda -Funktion. Weitere Informationen zum Schreiben von Lambda@Edge-Funktionen finden Sie in den folgenden Ressourcen:

- Lambda@Edge-Ereignisstruktur – Informationen zur Ereignisstruktur, die Sie für Lambda@Edge verwenden

- Beispielfunktionen für Lambda@Edge— Beispielfunktionen wie A/B Testen und Generieren einer HTTP-Umleitung.

Das Programmiermodell für die Verwendung von Node.js oder Python mit Lambda @Edge entspricht der Verwendung von Lambda in einem. AWS-Region Weitere Informationen finden Sie unter Erstellen von Lambda-Funktionen mit Node.js oder Erstellen von Lambda-Funktionen mit Python im AWS Lambda Entwicklerhandbuch.

Nehmen Sie in Ihre Lambda@Edge-Funktion den `callback`-Parameter auf und geben Sie das entsprechende Objekt für Anforderung- oder Antwortereignisse zurück:

- Request events (Anfrageereignisse) – Schließen Sie das `cf.request`-Objekt in die Antwort ein.

  Wenn Sie eine Antwort generieren, schließen Sie das Objekt `cf.response` in die Antwort ein. Weitere Informationen finden Sie unter Generieren von HTTP-Antworten in Anforderungsauslösern.
- Response events (Antwortereignisse): Schließen Sie das `cf.response`-Objekt in die Antwort ein.

Nachdem Sie Ihren eigenen Code geschrieben oder eines der Beispiele verwendet haben, erstellen Sie die Funktion in Lambda. Informationen zum Erstellen oder Bearbeiten einer vorhandenen Funktion finden Sie in den folgenden Themen:

Themen

- Erstellen einer Lambda@Edge-Funktion
- Bearbeiten einer Lambda-Funktion

Nachdem Sie die Funktion in Lambda erstellt haben, richten Sie Lambda so ein, dass die Funktion auf der Grundlage bestimmter CloudFront Ereignisse ausgeführt wird, die als Trigger bezeichnet werden. Weitere Informationen finden Sie unter Hinzufügen von Auslösern für eine Lambda@Edge-Funktion.

Erstellen einer Lambda@Edge-Funktion

Gehen Sie wie folgt vor AWS Lambda , um die Ausführung von Lambda-Funktionen einzurichten, die auf CloudFront Ereignissen basieren.

So erstellen Sie eine Lambda@Edge-Funktion

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die AWS Lambda Konsole unter https://console.aws.amazon.com/lambda/.
2. Wenn Sie bereits über eine oder mehrere Lambda-Funktionen verfügen, wählen Sie Create function.

   Wenn Sie nicht über Funktionen verfügen, wählen Sie Get Started Now.
3. Wählen Sie oben auf der Seite in der Liste „Region“ die Option US Ost (Nord-Virginia) aus.
4. Erstellen Sie eine Funktion mit Ihrem eigenen Code oder erstellen Sie eine Funktion, die mit einer CloudFront -Vorlage beginnt.

   - Um eine Funktion mit Ihrem eigenen Code zu erstellen, wählen Sie Author from scratch.
   - Um eine Liste mit Blueprints für anzuzeigen CloudFront, geben Sie Cloudfront in das Filterfeld ein und wählen Sie dann Enter.

     Wenn Sie eine dieser Vorlagen verwenden möchten, wählen Sie den Namen der entsprechenden Vorlage.
5. Geben Sie im Abschnitt Basic information folgende Werte ein:

   a. Name – Geben Sie einen Namen für die Funktion ein.

   b. Rolle – Um schnell loszulegen, wählen Sie Neue Rolle aus Vorlage(n) erstellen. Sie können auch Vorhandene Rolle auswählen oder Benutzerdefinierte Rolle erstellen auswählen und dann den Prompts folgen, um die Informationen für diesen Abschnitt zu vervollständigen.

   c. Rollenname – Geben Sie einen Namen für die Rolle ein.

   d. Richtlinienvorlagen – Wählen Sie Grundlegende Edge-Lambda-Berechtigungen aus.
6. Wenn Sie in Schritt 4 Author from scratch gewählt haben, fahren Sie mit Schritt 7 fort.

   Wenn Sie in Schritt 4 einen Blueprint ausgewählt haben, können Sie den Abschnitt Cloudfront verwenden, um einen Trigger zu erstellen, der diese Funktion einem Cache in einer CloudFront Verteilung und einem Ereignis zuordnet. CloudFront Wir empfehlen, an dieser Stelle Remove zu wählen, damit es bei der Erstellung der Funktionen keinen Auslöser gibt. Sie können Auslöser zu einem späteren Zeitpunkt hinzufügen.

> **Tip**
>
> Wir empfehlen Ihnen, die Funktion zu testen und zu debuggen, bevor Sie Auslöser hinzufügen. Wenn Sie jetzt einen Trigger hinzufügen, wird die Funktion ausgeführt, sobald Sie die Funktion erstellt haben. Die Replikation an AWS Standorte auf der ganzen Welt ist abgeschlossen und die entsprechende Distribution wird bereitgestellt.

7. Wählen Sie Funktion erstellen.

   Lambda erstellt zwei Versionen Ihrer Funktion: $LATEST und Version 1. Sie können nur die Version $LATEST bearbeiten, die Konsole zeit jedoch zunächst Version 1 an.

8. Um die Funktion zu bearbeiten, wählen Sie Version 1 oben auf der Seite, unter dem ARN für die Funktion. Wählen Sie anschließend auf der Registerkarte Versions die Option $LATEST. (Wenn Sie die Funktion verlassen haben und anschließend zurückgekehrt sind, lautet die Bezeichnung der Schaltfläche Qualifiers.)
9. Wählen Sie auf der Registerkarte Configuration den geeigneten Wert für Code entry type. Folgen Sie dann den Eingabeaufforderungen, um Ihren Code zu bearbeiten oder hochzuladen.
10. Wählen Sie den Wert für Runtime (Laufzeit) basierend auf dem Code der Funktion.
11. Fügen Sie im Bereich Tags geeignete Tags hinzu.
12. Wählen Sie Actions und dann Publish new version.
13. Geben Sie eine Beschreibung für die neue Version der Funktion ein.
14. Wählen Sie Publish.
15. Testen und debuggen Sie die Funktion. Weitere Informationen zu den Tests in der Lambda-Konsole finden Sie unter Aufrufen einer Lambda-Funktion mit der Konsole im Entwicklerhandbuch für AWS Lambda .
16. Wenn Sie bereit sind, die Funktion für CloudFront Ereignisse auszuführen, veröffentlichen Sie eine weitere Version und bearbeiten Sie die Funktion, um Trigger hinzuzufügen. Weitere Informationen finden Sie unter Hinzufügen von Auslösern für eine Lambda@Edge-Funktion.

Bearbeiten einer Lambda-Funktion

Nachdem Sie eine Lambda@Edge-Funktion erstellt haben, können Sie sie anhand der Lambda-Konsole bearbeiten.

Hinweise

- Die Originalversion ist mit $LATEST gekennzeichnet.
- Sie können nur die $LATEST-Version bearbeiten.
- Jedes Mal, wenn Sie die $LATEST-Version bearbeiten, müssen Sie eine neue nummerierte Version veröffentlichen.
- Sie können keine Auslöser für $LATEST erstellen.
- Wenn Sie eine neue Version einer Funktion veröffentlichen, kopiert Lambda nicht automatisch Auslöser von der vorherigen Version zur neuen Version. Sie müssen die Auslöser für die neue Version reproduzieren.
- Wenn Sie einer Funktion einen Trigger für ein CloudFront Ereignis hinzufügen und es bereits einen Trigger für dieselbe Verteilung, dasselbe Cache-Verhalten und dasselbe Ereignis für eine frühere Version derselben Funktion gibt, löscht Lambda den Trigger aus der früheren Version.
- Nachdem Sie Aktualisierungen an einer CloudFront Verteilung vorgenommen haben, z. B. Trigger hinzugefügt haben, müssen Sie warten, bis die Änderungen an den Edge-Standorten wirksam werden, bevor die Funktionen, die Sie in den Triggern angegeben haben, funktionieren.

So bearbeiten Sie eine Lambda-Funktion

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die AWS Lambda Konsole unter https://console.aws.amazon.com/lambda/.
2. Wählen Sie oben auf der Seite in der Liste „Region“ die Option US Ost (Nord-Virginia) aus.
3. Wählen Sie in der Liste der Funktionen den Namen der Funktion aus.

   Die Konsole zeigt standardmäßig die $LATEST-Version an. Sie können frühere Versionen anzeigen (wählen Sie Qualifiers), aber Sie können nur $LATEST bearbeiten.
4. Wählen Sie auf der Registerkarte Code für Code entry type (Code-Eingabetyp) die Bearbeitung des Codes im Browser, laden Sie eine .zip-Datei hoch oder laden Sie eine Datei aus Amazon S3 hoch.
5. Wählen Sie entweder Save oder Save and test.
6. Wählen Sie Actions und Publish new version.

7. Geben Sie im Dialogfeld Publish new version from $LATEST eine Beschreibung der neuen Version ein. Diese Beschreibung wird in der Liste der Versionen zusammen mit einer automatisch generierten Versionsnummer angezeigt.
8. Wählen Sie Publish.

   Die neue Version wird automatisch die aktuelle Version. Die Versionsnummer wird unter Version oben links auf der Seite angezeigt.

   > **Note**
   >
   > Wenn Sie noch keine Auslöser für die Funktion hinzugefügt haben, finden Sie Informationen hierzu unter Hinzufügen von Auslösern für eine Lambda@Edge-Funktion.

9. Wählen Sie die Registerkarte Triggers.
10. Wählen Sie Add trigger.
11. Wählen Sie im Dialogfeld Add trigger das Feld mit Punkten und dann CloudFront.

    > **Note**
    >
    > Wenn Sie bereits einen oder mehrere Auslöser für eine Funktion erstellt haben, CloudFront ist dies der Standarddienst.

12. Geben Sie die folgenden Werte an, um anzugeben, wann die Lambda-Funktion ausgeführt werden soll.

    a. Distributions-ID – Wählen Sie die ID der Distribution, der Sie den Auslöser hinzufügen möchten.

    b. Cacheverhalten – Wählen Sie das Cacheverhalten, das die Objekte angibt, für die Sie die Funktion ausführen möchten.

    c. CloudFront Ereignis — Wählen Sie das CloudFront Ereignis aus, das die Ausführung der Funktion veranlasst.

    d. Auslöser aktivieren und replizieren – Markieren Sie dieses Kontrollkästchen, sodass Lambda die Funktionen in AWS-Regionen weltweit repliziert.

13. Wählen Sie Absenden aus.
14. Um weitere Auslöser für diese Funktion hinzuzufügen, wiederholen Sie die Schritte 10 bis 13.

Weitere Informationen zum Testen und Debuggen der Funktion in der Lambda-Konsole finden Sie unter Aufrufen einer Lambda-Funktion mit der Konsole im Entwicklerhandbuch für AWS Lambda .

Wenn Sie bereit sind, die Funktion bei CloudFront Ereignissen ausführen zu lassen, veröffentlichen Sie eine weitere Version und bearbeiten Sie die Funktion, um Auslöser hinzuzufügen. Weitere Informationen finden Sie unter Hinzufügen von Auslösern für eine Lambda@Edge-Funktion.

Hinzufügen von Auslösern für eine Lambda@Edge-Funktion

Ein Lambda @Edge -Trigger ist eine Kombination aus einer CloudFront Verteilung, einem Cache-Verhalten und einem Ereignis, das die Ausführung einer Funktion bewirkt. Sie können beispielsweise einen Trigger erstellen, der bewirkt, dass die Funktion ausgeführt wird, wenn von einem Viewer eine Anfrage für ein bestimmtes Cache-Verhalten CloudFront empfangen wird, das Sie für Ihre Distribution eingerichtet haben. Sie können einen oder mehrere CloudFront Trigger angeben.

> **Tip**
>
> Wenn Sie eine CloudFront Verteilung erstellen, geben Sie Einstellungen an, die festlegen, CloudFront wie auf unterschiedliche Anfragen reagiert werden soll. Die Standardeinstellungen werden als Standard-Cacheverhalten für die Distribution bezeichnet. Sie können zusätzliche Cache-Verhaltensweisen einrichten, die definieren, wie unter bestimmten Umständen CloudFront reagiert wird, z. B. wenn eine Anfrage für einen bestimmten Dateityp empfangen wird. Weitere Informationen finden Sie unter Einstellungen für das Cache-Verhalten.

Wenn Sie eine Lambda-Funktion neu erstellen, können Sie nur einen Auslöser angeben. Sie können derselben Funktion später weitere Trigger hinzufügen, indem Sie die Lambda-Konsole verwenden oder die Verteilung in der CloudFront Konsole bearbeiten.

- Die Lambda-Konsole funktioniert gut, wenn Sie einer Funktion für dieselbe CloudFront Distribution weitere Trigger hinzufügen möchten.
- Die CloudFront Konsole kann besser sein, wenn Sie Trigger für mehrere Distributionen hinzufügen möchten, da es einfacher ist, die Distribution zu finden, die Sie aktualisieren möchten. Sie können auch andere CloudFront Einstellungen gleichzeitig aktualisieren.

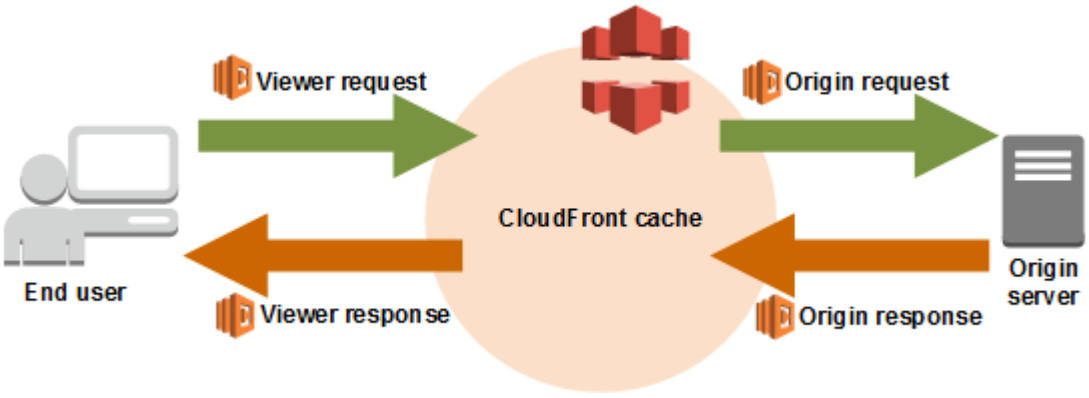
Themen

- CloudFront Ereignisse, die eine Lambda @Edge -Funktion auslösen können

- Auswählen des Ereignisses zur Auslösung der Funktion
- Hinzufügen von Auslösern zu einer Lambda@Edge-Funktion

CloudFront Ereignisse, die eine Lambda @Edge -Funktion auslösen können

Für jedes Cache-Verhalten in einer CloudFront Amazon-Distribution können Sie bis zu vier Trigger (Assoziationen) hinzufügen, die bewirken, dass eine Lambda-Funktion ausgeführt wird, wenn bestimmte CloudFront Ereignisse eintreten. CloudFront Trigger können auf einem von vier CloudFront Ereignissen basieren, wie in der folgenden Abbildung dargestellt.



Die folgenden CloudFront Ereignisse können verwendet werden, um Lambda @Edge -Funktionen auszulösen:

Viewer-Anforderung

Die Funktion wird ausgeführt, wenn sie eine Anfrage von einem Viewer CloudFront erhält, bevor sie überprüft, ob sich das angeforderte Objekt im CloudFront Cache befindet.

Die Funktion wird in den folgenden Fällen nicht ausgeführt:

- beim Abrufen einer benutzerdefinierten Fehlerseite
- Wenn eine HTTP-Anfrage CloudFront automatisch an HTTPS umgeleitet wird (wenn der Wert von Redirect HTTP to HTTPS Viewer-Protokollrichtlinien lautet).

Ursprungsanfrage

Die Funktion wird nur ausgeführt, wenn eine Anfrage an Ihren Ursprung CloudFront weitergeleitet wird. Wenn sich das angeforderte Objekt im CloudFront Cache befindet, wird die Funktion nicht ausgeführt.

Ursprungsantwort

Die Funktion wird ausgeführt, nachdem CloudFront sie eine Antwort vom Ursprung erhalten hat und bevor das Objekt in der Antwort zwischengespeichert wird. Beachten Sie, dass die Funktion auch dann ausgeführt wird, wenn ein Fehler vom Ursprung zurückgegeben wird.

Die Funktion wird in den folgenden Fällen nicht ausgeführt:

- Wenn sich die angeforderte Datei im CloudFront Cache befindet und nicht abgelaufen ist.
- Wenn die Antwort von einer Funktion generiert wird, die von einem Ursprungs-Anforderungsereignis ausgelöst wurde.

Viewer-Antwort

Diese Funktion wird ausgeführt, bevor die angeforderte Datei an den Viewer zurückgegeben wird. Beachten Sie, dass die Funktion unabhängig davon ausgeführt wird, ob sich die Datei bereits im CloudFront Cache befindet.

Die Funktion wird in den folgenden Fällen nicht ausgeführt:

- Wenn der Ursprung den HTTP-Statuscode „400“ oder höher zurückgibt.
- Wenn eine benutzerdefinierte Fehlerseite zurückgesendet wird.
- Wenn die Antwort von einer Funktion generiert wird, die von einem Viewer-Anforderungsereignis ausgelöst wurde.
- Wenn eine HTTP-Anfrage CloudFront automatisch an HTTPS umgeleitet wird (wenn der Wert von „HTTP zu HTTPS umleiten“ Viewer-Protokollrichtlinien lautet).

Wenn Sie einem Cache-Verhalten mehrere Auslöser hinzufügen, können diese jeweils dieselbe oder verschiedene Funktionen für jeden Auslöser ausführen. Sie können Funktionen auch in mehreren Verteilungen zuweisen.

> **Note**
>
> Wenn ein CloudFront Ereignis die Ausführung einer Lambda-Funktion auslöst, muss die Funktion beendet werden, bevor sie fortgesetzt CloudFront werden kann.
> Wenn beispielsweise eine Lambda-Funktion durch ein CloudFront Viewer-Anforderungsereignis ausgelöst CloudFront wird, wird keine Antwort an den Betrachter zurückgegeben oder die Anfrage an den Ursprung weitergeleitet, bis die Lambda-Funktion vollständig ausgeführt wurde.

Dies bedeutet, dass jede Anforderung, die eine Lambda-Funktion auslöst, die Latenz für die Anforderung erhöht. Daher sollte die Funktion so schnell wie möglich arbeiten.

Auswählen des Ereignisses zur Auslösung der Funktion

Wenn Sie entscheiden, welches CloudFront Ereignis Sie verwenden möchten, um eine Lambda-Funktion auszulösen, sollten Sie Folgendes berücksichtigen:

Ich CloudFront möchte Objekte zwischenspeichern, die durch eine Lambda-Funktion geändert wurden

Um ein Objekt zwischenzuspeichern, das durch eine Lambda-Funktion geändert wurde, CloudFront sodass es bei der nächsten Anforderung vom Edge-Standort aus bedient werden kann, verwenden Sie die Origin-Anfrage oder das Origin-Response-Ereignis.

Dadurch verringert sich die Verarbeitungslast für den Ursprung, die Latenz für nachfolgende Anforderungen wird verringert und die Kosten für den Aufruf von Lambda@Edge für nachfolgende Anforderungen reduziert.

Wenn Sie beispielsweise Header für Objekte hinzufügen, entfernen oder ändern möchten, die vom Ursprung zurückgegeben werden, und Sie das Ergebnis zwischenspeichern CloudFront möchten, verwenden Sie das Origin-Response-Ereignis.

Die Funktion soll für jede Anforderung ausgeführt werden

Um die Funktion für jede Anforderung auszuführen, die für die Verteilung CloudFront eingeht, verwenden Sie die Viewer-Anforderung oder die Viewer-Antwortereignisse.

Origin-Request- und Origin-Response-Ereignisse treten nur auf, wenn ein angefordertes Objekt nicht an einem Edge-Standort zwischengespeichert ist und eine Anfrage an den Ursprung CloudFront weiterleitet.

Die Funktion soll den Cache-Schlüssel ändern

Um einen Wert zu ändern, den Sie als Grundlage für das Zwischenspeichern verwenden, wählen Sie das Ereignis Viewer-Anforderung.

Wenn beispielsweise eine Funktion eine URL so ändert, dass Abkürzungen für die Sprachversionen in den Pfad eingebunden werden (zum Beispiel, weil der Benutzer seine Sprache aus einer Dropdown-Liste ausgewählt hat), verwenden Sie das Viewer-Anfrageereignis:

- URL in der Viewer-Anfrage — https://example.com/en/index.html
- URL, wenn die Anfrage von einer IP-Adresse in Deutschland kommt — https://example.com/de/index.html

Sie können Viewer-Anfrageereignisse auch verwenden, wenn die Zwischenspeicherung auf der Basis von Cookies oder Anfrageereignissen erfolgt.

> **Note**
>
> Wenn die Funktion Cookies oder Header ändert, konfigurieren Sie CloudFront sie so, dass der entsprechende Teil der Anfrage an den Ursprung weitergeleitet wird. Weitere Informationen finden Sie unter den folgenden Themen:
>
> - Zwischenspeichern von Inhalten auf der Grundlage von Cookies
> - Zwischenspeichern von Inhalten auf der Grundlage von Anforderungsheadern

Die Funktion wirkt sich auf die Antwort vom Ursprung aus

Um die Anforderung so zu verändern, dass sich die Antwort vom Ursprung ändert, verwenden Sie das Ereignis Ursprungsanforderung.

In der Regel werden die meisten Ereignisse mit Zuschaueranfragen nicht an den Absender weitergeleitet. CloudFront reagiert auf eine Anfrage mit einem Objekt, das sich bereits im Edge-Cache befindet. Wenn die Funktion die Anfrage auf der Grundlage eines ursprünglichen Anforderungsereignisses ändert, CloudFront speichert sie die Antwort auf die geänderte ursprüngliche Anfrage im Cache.

Hinzufügen von Auslösern zu einer Lambda@Edge-Funktion

Sie können die AWS Lambda Konsole oder die CloudFront Amazon-Konsole verwenden, um Ihrer Lambda @Edge -Funktion einen Trigger hinzuzufügen.

> **Important**
>
> Sie können Auslöser nur für nummerierte Versionen Ihrer Funktion erstellen (nicht für $LATEST).

Lambda console

Um Trigger für CloudFront Ereignisse zu einer Lambda @Edge -Funktion hinzuzufügen

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die AWS Lambda Konsole unter https://console.aws.amazon.com/lambda/.
2. Wählen Sie oben auf der Seite in der Liste „Region“ die Option US Ost (Nord-Virginia) aus.
3. Wählen Sie auf der Seite Functions den Namen der Funktion, für die Sie Auslöser hinzufügen möchten.
4. Wählen Sie auf der Seite Funktionsübersicht die Registerkarte Versionen aus.
5. Wählen Sie die Version, der Sie Auslöser hinzufügen möchten.

   Nach der Wahl einer Version ändert sich der Name der Schaltfläche in Version: $LATEST oder Version: Versionsnummer.
6. Wählen Sie die Registerkarte Triggers.
7. Wählen Sie Add trigger.
8. Wählen Sie unter Trigger-Konfiguration die Option Quelle auswählen**`cloudfront`**, geben Sie die Eingabetaste ein und wählen Sie dann CloudFront.

   > **Note**
   >
   > Wenn Sie bereits einen oder mehrere Auslöser erstellt haben, CloudFront ist dies der Standarddienst.
9. Geben Sie die folgenden Werte an, um anzugeben, wann die Lambda-Funktion ausgeführt werden soll.

   a. Distribution – Wählen Sie die Distribution aus, der Sie den Auslöser hinzufügen möchten.

   b. Cacheverhalten – Wählen Sie das Cacheverhalten, das die Objekte angibt, für die Sie die Funktion ausführen möchten.

   > **Note**
   >
   > Wenn Sie * für das Zwischenspeicher-Verhalten angeben, stellt die Lambda-Funktion das Standard-Zwischenspeicher-Verhalten bereit.

c. CloudFront Ereignis — Wählen Sie das CloudFront Ereignis aus, das die Ausführung der Funktion veranlasst.

d. Text einschließen – Aktivieren Sie dieses Kontrollkästchen, wenn Sie auf den Anforderungstext in Ihrer Funktion zugreifen möchten.

e. Bereitstellung in Lambda@Edge bestätigen – Aktivieren Sie dieses Kontrollkästchen, sodass AWS Lambda die Funktion in AWS-Regionen weltweit repliziert.

10. Wählen Sie Hinzufügen aus.

    Die Funktion beginnt mit der Verarbeitung von Anfragen für die angegebenen CloudFront Ereignisse, wenn die aktualisierte CloudFront Distribution bereitgestellt wird. Um zu ermitteln, ob eine Verteilung bereitgestellt ist, wählen Sie im Navigationsbereich die Option Distributions. Wenn eine Distribution bereitgestellt ist, ändert sich der Wert in der Spalte Status für die Distribution von Wird bereitgestellt in das Datum und die Uhrzeit der Bereitstellung.

CloudFront console

Um Trigger für CloudFront Ereignisse zu einer Lambda @Edge -Funktion hinzuzufügen

1. Rufen Sie den ARN der Lambda-Funktion ab, der Sie Auslöser hinzufügen möchten:

   a. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die AWS Lambda Konsole unter https://console.aws.amazon.com/lambda/.

   b. Wählen Sie in der Liste der Regionen oben auf der Seite die Option USA Ost (Nord-Virginia).

   c. Wählen Sie in der Liste der Funktionen den Namen der Funktion, der Sie Auslöser hinzufügen möchten.

   d. Wählen Sie auf der Seite Funktionsübersicht die Registerkarte Versionen und dann die nummerierte Version aus, der Sie Auslöser hinzufügen möchten.

   e. Wählen Sie die Schaltfläche ARN kopieren, um den ARN in Ihre Zwischenablage zu kopieren. Der ARN für die Lambda-Funktion sieht wie folgt aus:

      ```
      arn:aws:lambda:us-east-1:123456789012:function:TestFunction:2
      ```

      Die Nummer am Ende (2 in diesem Beispiel) ist die Versionsnummer der Funktion.

2. Öffnen Sie die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home.
3. Wählen Sie in der Liste der Verteilungen die ID der Verteilung aus, der Sie Auslöser hinzufügen möchten.
4. Wählen Sie die Registerkarte Behaviors aus.
5. Wählen Sie das Cacheverhalten, dem Sie Auslöser hinzufügen möchten, und dann Bearbeiten aus.
6. Wählen Sie unter Funktionszuordnungen in der Liste Funktionstyp die Option Lambda@Edge aus, wenn die Funktion bei Viewer-Anforderungen, Viewer-Antworten, Ursprungsanforderungen oder Ursprungsantworten ausgeführt werden soll.

   Weitere Informationen finden Sie unter Auswählen des Ereignisses zur Auslösung der Funktion.
7. Fügen Sie in das Textfeld Funktions-ARN/Name den ARN der Lambda-Funktion ein, die ausgeführt werden soll, wenn das ausgewählte Ereignis eintritt. Dies ist der Wert, den Sie aus der Lambda-Konsole kopiert haben.
8. Wählen Sie Text einschließen aus, wenn Sie auf den Anforderungstext in Ihrer Funktion zugreifen möchten.

   Wenn Sie nur den Anforderungstext ersetzen möchten, müssen Sie diese Option nicht auswählen.
9. Wenn dieselbe Funktion für mehrere Ereignistypen ausgeführt werden soll, wiederholen Sie die Schritte 6 und 7.
10. Wählen Sie Änderungen speichern aus.
11. Um Trigger zu weiteren Cache-Verhaltensweisen für diese Verteilung hinzuzufügen, wiederholen Sie die Schritte 5 bis 10.

    Die Funktion beginnt mit der Verarbeitung von Anfragen für die angegebenen CloudFront Ereignisse, wenn die aktualisierte CloudFront Distribution bereitgestellt wird. Um zu ermitteln, ob eine Verteilung bereitgestellt ist, wählen Sie im Navigationsbereich die Option Distributions. Wenn eine Distribution bereitgestellt ist, ändert sich der Wert in der Spalte Status für die Distribution von Wird bereitgestellt in das Datum und die Uhrzeit der Bereitstellung.

Testen und Debuggen von Lambda@Edge-Funktionen

Es ist wichtig, dass Sie Ihren Lambda @Edge -Funktionscode eigenständig testen, um sicherzustellen, dass er die beabsichtigte Aufgabe erfüllt, und Integrationstests durchzuführen, um sicherzustellen, dass die Funktion korrekt funktioniert. CloudFront

Während des Integrationstests oder nach der Bereitstellung Ihrer Funktion müssen Sie möglicherweise Fehler wie HTTP CloudFront 5xx-Fehler debuggen. Fehler können eine ungültige Antwort der Lambda-Funktion, Ausführungsfehler beim Auslösen der Funktion oder Fehler aufgrund einer Ablehnung der Ausführung durch den Lambda-Service sein. Die Abschnitte in diesem Thema enthalten Strategien, um festzustellen, welche Art von Fehler das Problem verursacht. Dazu finden Sie Schritte, die Sie unternehmen können, um das Problem zu beheben.

> **Note**
>
> Achten Sie bei der Überprüfung von CloudWatch Protokolldateien oder Metriken bei der Behebung von Fehlern darauf, dass diese an dem Ort angezeigt oder gespeichert werden, der dem Ort, an dem die Funktion ausgeführt wurde, AWS-Region am nächsten ist. Wenn Sie also eine Website oder Webanwendung mit Benutzern im Vereinigtes Königreich haben und Ihrer Distribution beispielsweise eine Lambda-Funktion zugeordnet ist, müssen Sie die Region ändern, um die CloudWatch Metriken oder Protokolldateien für London AWS-Region anzuzeigen. Weitere Informationen finden Sie unter the section called “Bestimmen der Lambda@Edge-Region”.

Themen

- Testen der Lambda@Edge-Funktionen
- Identifizieren Sie Lambda @Edge -Funktionsfehler in CloudFront
- Fehlerbehebung bei ungültigen Lambda@Edge-Funktionsantworten (Validierungsfehler)
- Fehlerbehebung bei Lambda@Edge-Funktionsausführungsfehlern
- Bestimmen der Lambda@Edge-Region
- Ermitteln Sie, ob Ihr Konto Logs an weiterleitet CloudWatch

Testen der Lambda@Edge-Funktionen

Der Test Ihrer Lambda-Funktion besteht aus zwei Schritten: eigenständiger Test und Integrationstest.

Eigenständiger Test der Funktionalität

Bevor Sie Ihre Lambda-Funktion hinzufügen CloudFront, stellen Sie sicher, dass Sie die Funktionalität zuerst testen, indem Sie die Testfunktionen in der Lambda-Konsole oder andere Methoden verwenden. Weitere Informationen zu den Tests in der Lambda-Konsole finden Sie unter Aufrufen einer Lambda-Funktion mit der Konsole im Entwicklerhandbuch für AWS Lambda .

Testen Sie den Betrieb Ihrer Funktion in CloudFront

Es ist wichtig, Integrationstests abzuschließen, bei denen Ihre Funktion einer Distribution zugeordnet ist und auf der Grundlage eines CloudFront Ereignisses ausgeführt wird. Stellen Sie sicher, dass die Funktion für das richtige Ereignis ausgelöst wird und eine für CloudFront gültige und korrekte Antwort zurückgibt. Achten Sie beispielsweise darauf, dass die Ereignisstruktur korrekt ist, dass nur gültige Header enthalten sind usw.

Wenn Sie die Integrationstests mit Ihrer Funktion in der Lambda-Konsole wiederholen, folgen Sie den Schritten im Lambda @Edge -Tutorial, wenn Sie Ihren Code oder den CloudFront Trigger ändern, der Ihre Funktion aufruft. Stellen Sie beispielsweise sicher, dass Sie mit einer nummerierten Version Ihrer Funktion arbeiten, wie in diesem Schritt des Tutorials beschrieben: Schritt 4: Fügen Sie einen CloudFront Trigger hinzu, um die Funktion auszuführen.

Beachten Sie beim Vornehmen von Änderungen und deren Bereitstellung, dass es mehrere Minuten dauern kann, bis Ihre aktualisierte Funktion und CloudFront Trigger in allen Regionen repliziert sind. Dies dauert in der Regel wenige Minuten, kann jedoch auch bis zu 15 Minuten dauern.

Sie können überprüfen, ob die Replikation abgeschlossen ist, indem Sie zur CloudFront Konsole gehen und sich Ihre Distribution ansehen.

So überprüfen Sie, ob die Bereitstellung Ihrer Replikation abgeschlossen ist

1. Öffnen Sie die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home.
2. Wählen Sie den Distributionsnamen aus.
3. Überprüfen Sie, ob sich der Status der Verteilung von InProgress (Läuft) zurück auf Deployed (Bereitgestellt) geändert hat, d.h. Ihre Funktion wurde repliziert. Anschließend befolgen Sie die Schritte im nächsten Abschnitt, um zu überprüfen, ob die Funktion funktioniert.

Beachten Sie, dass das Testen in der Konsole nur die Logik Ihrer Funktion validiert und keine Servicekontingente (früher als Limits bezeichnet) anwendet, die spezifisch für Lambda@Edge sind.

Identifizieren Sie Lambda @Edge -Funktionsfehler in CloudFront

Nachdem Sie überprüft haben, ob Ihre Funktionslogik korrekt funktioniert, werden möglicherweise immer noch HTTP 5xx-Fehler angezeigt, wenn Ihre Funktion ausgeführt wird. CloudFront HTTP 5xx-Fehler können aus einer Vielzahl von Gründen zurückgegeben werden, darunter Lambda-Funktionsfehler oder andere Probleme in. CloudFront

- Wenn Sie Lambda @Edge -Funktionen verwenden, können Sie Diagramme in der CloudFront Konsole verwenden, um herauszufinden, was den Fehler verursacht, und dann daran arbeiten, ihn zu beheben. Sie können beispielsweise sehen, ob HTTP 5xx-Fehler durch CloudFront oder durch Lambda-Funktionen verursacht werden, und dann können Sie für bestimmte Funktionen zugehörige Protokolldateien einsehen, um das Problem zu untersuchen.
- Informationen zur allgemeinen Behebung von HTTP-Fehlern finden Sie in CloudFront den Schritten zur Fehlerbehebung im folgenden Thema:. Problembehandlung bei Statuscodes für die Fehlerantwort in CloudFront

Was verursacht Lambda @Edge -Funktionsfehler in CloudFront

Es gibt mehrere Gründe, aus denen eine Lambda-Funktion einen HTTP 5xx-Fehler verursachen kann. Die Schritte zur Fehlerbehebung hängen von der Art des Fehlers ab. Fehler können wie folgt kategorisiert werden:

Ein Lambda-Funktionsausführungsfehler.

Ein Ausführungsfehler tritt auf, wenn Sie CloudFront keine Antwort von Lambda erhalten, weil die Funktion unbehandelte Ausnahmen enthält oder ein Fehler im Code vorliegt. Zum Beispiel, wenn der Code "callback(Error)" beinhaltet.

Eine ungültige Lambda-Funktionsantwort wird zurückgegeben an CloudFront

CloudFront Erhält nach der Ausführung der Funktion eine Antwort von Lambda. Ein Fehler wird zurückgegeben, wenn die Objektstruktur der Antwort nicht der Lambda@Edge-Ereignisstruktur entspricht oder die Antwort ungültige Header oder andere ungültige Felder enthält.

Die Ausführung in CloudFront wird aufgrund von Lambda-Servicequotas (früher als Limits bezeichnet) gedrosselt

Der Lambda-Service drosselt die Ausführung in jeder Region und gibt einen Fehler zurück, wenn Sie das Kontingent überschreiten. Weitere Informationen finden Sie unter Kontingente für Lambda@Edge.

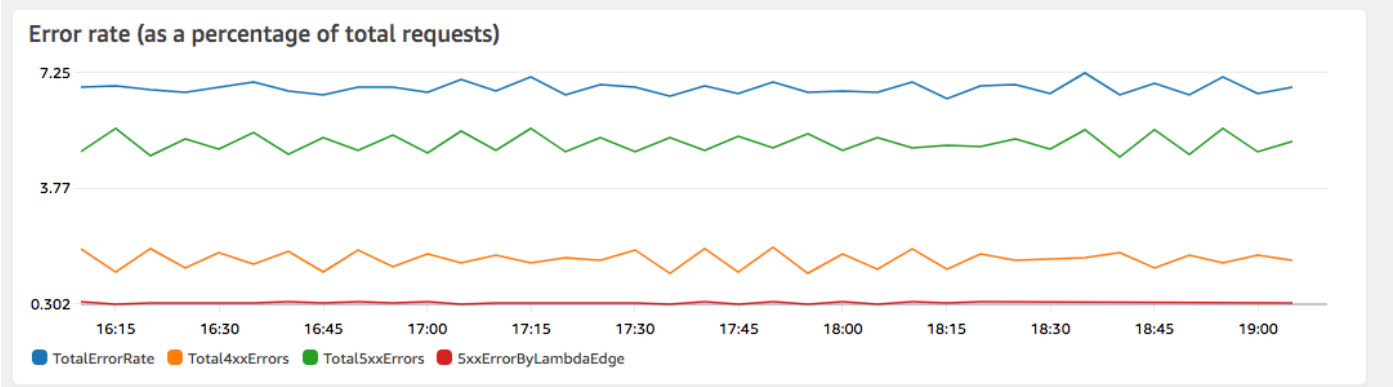
So bestimmten Sie den Typ des Fehlers

Um Ihnen bei der Entscheidung zu helfen, worauf Sie sich beim Debuggen und bei der Behebung von Fehlern konzentrieren sollten, ist es hilfreich CloudFront, herauszufinden, warum CloudFront ein HTTP-Fehler zurückgegeben wird. Zu Beginn können Sie die Grafiken verwenden, die im Abschnitt Überwachung der CloudFront Konsole auf dem AWS-Managementkonsole bereitgestellt werden. Weitere Informationen zum Anzeigen von Diagrammen im Bereich Überwachung der CloudFront Konsole finden Sie unterÜberwachen Sie CloudFront Metriken mit Amazon CloudWatch.

Die folgenden Diagramme sind besonders hilfreich, wenn Sie nachverfolgen möchten, ob Fehler von Ursprungs-Servern oder einer Lambda-Funktion zurückgegeben werden, und wenn Sie die Art des Problems eingrenzen möchten, wenn der Fehler aus einer Lambda-Funktion resultiert.

Fehlerratendiagramm

Eines der Diagramme, das Sie auf der Registerkarte Overview (Überblick) für Ihre Verteilungen anzeigen können, ist das Diagramm Error rates (Fehlerraten). Dieses Diagramm zeigt die Rate der Fehler als Prozentsatz aller Anforderungen an, die bei Ihren Verteilungen eingehen. Das Diagramm zeigt die Gesamtfehlerrate, die gesamten 4xx-Fehler, die gesamten 5xx-Fehler und die gesamten 5xx-Fehler an, die aus Lambda-Funktionen resultieren. Basierend auf Fehlertyp und Anzahl können Sie Schritte unternehmen, um diese Probleme zu untersuchen und zu beheben.



- Wenn Lambda-Fehler auftreten, können Sie eine genauere Untersuchung durchführen, indem Sie sich die spezifischen Arten von Fehlern ansehen, die von der Funktion zurückgegeben werden. Die Registerkarte Lambda@Edge errors (Lambda@Edge-Fehler) enthält Diagramme, die Funktionsfehler nach Typ kategorisieren, damit Sie das Problem für eine bestimmte Funktion herausfinden können.
- Wenn Sie CloudFront Fehler sehen, können Sie Fehler beheben und daran arbeiten, die ursprünglichen Fehler zu beheben oder Ihre CloudFront Konfiguration zu ändern. Weitere Informationen finden Sie unter Problembehandlung bei Statuscodes für die Fehlerantwort in CloudFront.

Diagramme für Ausführungsfehler und ungültige Funktionsanworten

Die Registerkarte Lambda@Edge errors (Lambda@Edge-Fehler) enthält Diagramme, die Lambda@Edge-Fehler für eine bestimmte Verteilung nach Typ kategorisieren. So zeigt beispielsweise ein Diagramm alle Ausführungsfehler unterteilt nach AWS-Region an.

Damit Fehler einfacher erkannt und behoben werden können, können Sie nach bestimmten Problemen suchen, indem Sie die Protokolldateien für spezifische Funktionen regionsweise öffnen und prüfen.

So zeigen Sie Protokolldateien für eine bestimmte Funktion nach Region an

1. Wählen Sie auf der Registerkarte Lambda@Edge-Fehler unter Zugeordnete Lambda@Edge-Funktionen den Funktionsnamen und dann Metriken anzeigen aus.
2. Klicken Sie anschließend auf der Seite mit dem Namen der Funktion rechts oben auf Funktionsprotokolle anzeigen und wählen Sie dann eine Region aus.

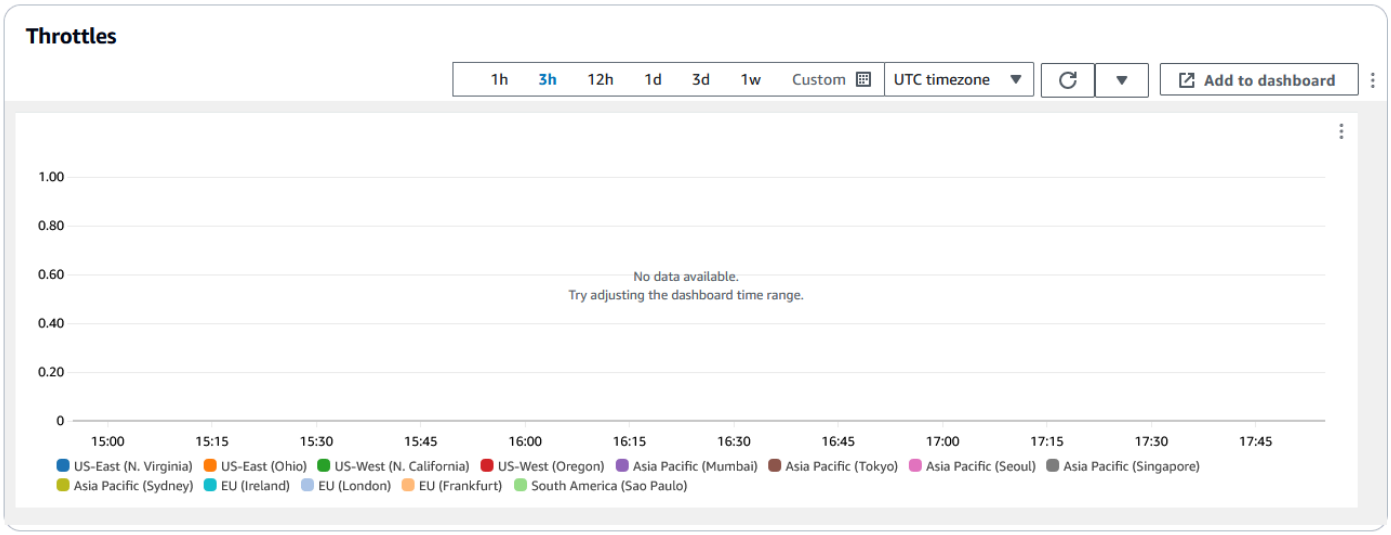
   Wenn Sie beispielsweise Probleme im Diagramm Fehler für die Region USA West (Oregon) sehen, wählen Sie diese Region aus der Dropdown-Liste aus. Dadurch wird die CloudWatch Amazon-Konsole geöffnet.
3. Wählen Sie in der CloudWatch Konsole für diese Region unter Protokollstreams einen Protokollstream aus, um die Ereignisse für die Funktion anzuzeigen.

Lesen Sie darüber hinaus die folgenden Abschnitte in diesem Kapitel, um weitere Empfehlungen für die Behebung von Fehlern zu erhalten.

Drosselungsdiagramm

Die Registerkarte Lambda@Edge errors (Lambda@Edge-Fehler) enthält auch ein Diagramm zu Drosselungen. In einigen Situationen drosselt der Lambda-Service Ihre Funktionsaufrufe auf

Regionsbasis, wenn Sie sich dem regionalen Kontingent (früher als Limit bezeichnet) für die Gleichzeitigkeit nähern. Wenn Sie einen limit exceeded (Grenzwert überschritten)-Fehler sehen, hat Ihre Funktion ein Kontingent erreicht, das der Lambda-Service für Ausführungen in einer Region nutzt. Weitere Informationen, u. a. auch zur Erhöhung des Kontingents, finden Sie unter Kontingente für Lambda@Edge.



Ein Beispiel zur Verwendung dieser Informationen beim Beheben von HTTP-Fehlern finden Sie unter Vier Schritte zum Debuggen der Bereitstelllung von Inhalten in AWS.

Fehlerbehebung bei ungültigen Lambda@Edge-Funktionsantworten (Validierungsfehler)

Wenn Sie feststellen, dass es sich bei Ihrem Problem um einen Lambda-Validierungsfehler handelt, bedeutet dies, dass Ihre Lambda-Funktion eine ungültige Antwort auf zurückgibt. CloudFront Folgen Sie den Anweisungen in diesem Abschnitt, um Maßnahmen zur Überprüfung Ihrer Funktion zu ergreifen und sicherzustellen, dass Ihre Antwort den Anforderungen entspricht. CloudFront

CloudFront validiert die Antwort einer Lambda-Funktion auf zwei Arten:

- Die Lambda-Antwort muss der gewünschten Objektstruktur entsprechen. Beispiele für eine fehlerhafte Objektstruktur sind: nicht interpretierbarer JSON-Code, fehlende Pflichtfelder und ein ungültiges Objekt in der Antwort. Weitere Informationen hierzu finden Sie unter Lambda@Edge-Ereignisstruktur.
- Die Antwort darf nur gültige Objektwerte beinhalten. Ein Fehler tritt auf, wenn die Antwort ein gültiges Objekt aber nicht unterstützte Werte enthält. Beispiele sind das Hinzufügen oder Aktualisieren von ungültigen Headern oder schreibgeschützten Headern (siehe Einschränkungen

für Edge-Funktionen), das Überschreiten der maximalen Body-Größe (siehe Beschränkungen für die Größe der generierten Antwort im Thema Lambda@Edge Fehler) und ungültige Zeichen oder Werte (siehe Lambda@Edge-Ereignisstruktur).

Wenn Lambda eine ungültige Antwort auf zurückgibt CloudFront, werden Fehlermeldungen in Protokolldateien geschrieben, die in die Region CloudFront übertragen werden, CloudWatch in der die Lambda-Funktion ausgeführt wurde. Dies ist das Standardverhalten, an das die Protokolldateien gesendet werden, CloudWatch wenn eine ungültige Antwort eingeht. Wenn Sie jedoch eine Lambda-Funktion mit verknüpft haben, CloudFront bevor die Funktionalität veröffentlicht wurde, ist sie möglicherweise nicht für Ihre Funktion aktiviert. Weitere Informationen finden Sie unter Feststellen, ob Ihr Konto Protokolle an CloudWatch überträgt weiter unten im Thema.

CloudFront überträgt Protokolldateien in die Region, die der Region entspricht, in der Ihre Funktion ausgeführt wurde, in der Protokollgruppe, die Ihrer Distribution zugeordnet ist. Protokollgruppen haben das folgende Format:`/aws/cloudfront/LambdaEdge/`*`DistributionId`*, wo *`DistributionId`* ist die ID Ihrer Distribution. Informationen zur Bestimmung der Region, in der Sie die CloudWatch Protokolldateien finden, finden Sie unter Bestimmung der Lambda @Edge -Region weiter unten in diesem Thema.

Wenn der Fehler reproduzierbar ist, können Sie eine neue Anfrage erstellen, die zu dem Fehler führt, und dann die Anforderungs-ID in einer fehlgeschlagenen CloudFront Antwort (`X-Amz-Cf-Id`Header) suchen, um einen einzelnen Fehler in den Protokolldateien zu lokalisieren. Der Eintrag in der Protokolldatei enthält Informationen, die Ihnen bei der Identifizierung Ursache für den Fehler helfen können. Er zeigt außerdem die entsprechende Lambda-Anforderungs-ID an, mit der Sie die Ursache im Kontext einer einzelnen Anforderung analysieren können.

Wenn ein Fehler nur sporadisch auftritt, können Sie mithilfe von CloudFront Zugriffsprotokollen die Anforderungs-ID für eine fehlgeschlagene Anfrage ermitteln und anschließend die CloudWatch Protokolle nach den entsprechenden Fehlermeldungen durchsuchen. Weitere Informationen finden Sie im vorherigen Abschnitt Bestimmung der Fehlerart.

Fehlerbehebung bei Lambda@Edge-Funktionsausführungsfehlern

Wenn es sich bei dem Problem um einen Lambda-Ausführungsfehler handelt, kann es hilfreich sein, Protokollierungsanweisungen für Lambda-Funktionen zu erstellen, Meldungen in CloudWatch Protokolldateien zu schreiben, die die Ausführung Ihrer Funktion überwachen CloudFront und feststellen, ob sie wie erwartet funktioniert. Anschließend können Sie in den CloudWatch Protokolldateien nach diesen Anweisungen suchen, um zu überprüfen, ob Ihre Funktion funktioniert.

> **Note**
>
> Auch wenn Sie Ihre Lambda@Edge-Funktion nicht geändert haben, kann diese durch Aktualisierungen der Lambda-Funktionsausführungsumgebung beeinträchtigt werden und einen Ausführungsfehler ausgeben. Informationen zum Testen und Migrieren auf eine neuere Version finden Sie unter Kommende Updates für die AWS Lambda- und AWS Lambda @Edge -Ausführungsumgebung.

Bestimmen der Lambda@Edge-Region

Um die Regionen zu sehen, in denen Ihre Lambda @Edge -Funktion Traffic empfängt, sehen Sie sich die Metriken für die Funktion auf der CloudFront Konsole auf der AWS-Managementkonsole an. Metriken werden für jede AWS Region angezeigt. Auf derselben Seite können Sie eine Region auswählen und Protokolldateien für diese anzeigen, um Probleme näher zu untersuchen. Sie müssen die CloudWatch Protokolldateien in der richtigen AWS Region überprüfen, um die Protokolldateien zu sehen, die bei der CloudFront Ausführung Ihrer Lambda-Funktion erstellt wurden.

Weitere Informationen zum Anzeigen von Diagrammen im Bereich Überwachung der CloudFront Konsole finden Sie unterÜberwachen Sie CloudFront Metriken mit Amazon CloudWatch.

Ermitteln Sie, ob Ihr Konto Logs an weiterleitet CloudWatch

CloudFront Aktiviert standardmäßig die Protokollierung ungültiger Lambda-Funktionsantworten und überträgt die Protokolldateien CloudWatch mithilfe einer der. Service-linked Rollen für Lambda @Edge Wenn Sie Lambda @Edge -Funktionen haben, die Sie hinzugefügt haben, CloudFront bevor die Funktion für das Protokoll der ungültigen Lambda-Funktionsantwort veröffentlicht wurde, wird die Protokollierung aktiviert, wenn Sie Ihre Lambda @Edge -Konfiguration das nächste Mal aktualisieren, z. B. indem Sie einen Trigger hinzufügen. CloudFront

Gehen Sie wie folgt vor, um zu überprüfen, ob die Übertragung der Protokolldateien an für Ihr Konto aktiviert CloudWatch ist:

- Prüfen Sie, ob die Protokolle in erscheinen CloudWatch — Achten Sie darauf, dass Sie in der Region suchen, in der die Lambda @Edge -Funktion ausgeführt wurde. Weitere Informationen finden Sie unter Bestimmen der Lambda@Edge-Region.
- Stellen Sie fest, ob die zugehörige serviceverknüpfte Rolle in Ihrem Konto in IAM vorhanden ist. Die IAM-Rolle `AWSServiceRoleForCloudFrontLogger` muss in Ihrem Konto vorhanden sein. Weitere Informationen über diese Rolle finden Sie unter Service-linked Rollen für Lambda @Edge.

Löschen von Lambda@Edge-Funktionen und Replikaten

Sie können eine Lambda@Edge-Funktion nur löschen, wenn die Replikate der Funktion von CloudFront gelöscht wurden. Replikate einer Lambda-Funktion werden in den folgenden Situationen automatisch gelöscht:

- Nachdem Sie die letzte Zuordnung für die Funktion aus allen Ihren CloudFront-Verteilungen entfernt haben. Wenn mehrere Verteilungen eine Funktion verwenden, werden die Replikate erst gelöscht, nachdem Sie die Funktionszuordnung aus der letzten Verteilung entfernt haben.
- Nachdem Sie die letzte Verteilung, der eine Funktion zugeordnet war, gelöscht haben.

Replicas werden in der Regel innerhalb von wenigen Stunden gelöscht. Lambda@Edge-Funktionsreplikate können nicht manuell gelöscht werden. Dadurch wird verhindert, dass ein Replikat gelöscht wird, das noch verwendet wird, was zu einem Fehler führen würde.

> **Warning**
>
> Erstellen Sie keine Anwendungen, die Lambda @Edge -Funktionsreplikate außerhalb von verwenden. CloudFront Diese Replikate werden gelöscht, wenn ihre Zuordnungen zu Verteilungen entfernt werden oder wenn die Verteilungen selbst gelöscht werden. Das Replikat, von dem eine externe Anwendung abhängt, könnte ohne Warnung entfernt werden, was zu einem Fehler führen würde.

Um eine Lambda @Edge -Funktionsassoziation aus einer CloudFront Distribution zu löschen

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home.
2. Wählen Sie die ID der Distribution mit der Lambda@Edge-Funktionszuordnung aus, die Sie löschen möchten.
3. Wählen Sie die Registerkarte Behaviors aus.
4. Wählen Sie das Cacheverhalten mit der Lambda@Edge-Funktionszuordnung, die Sie löschen möchten, und dann Bearbeiten aus.
5. Wählen Sie unter Funktionszuordnungen, Funktionstyp die Option Keine Zuordnung aus, um die Lambda@Edge-Funktionszuordnung zu löschen.
6. Wählen Sie Änderungen speichern aus.

Nachdem Sie eine Lambda @Edge -Funktionszuordnung aus einer CloudFront Distribution gelöscht haben, können Sie optional die Lambda-Funktion oder Funktionsversion aus löschen. AWS Lambda Warten Sie nach dem Löschen der Funktionszuordnung einige Stunden, damit die Lambda@Edge-Funktionsreplikate bereinigt werden können. Danach können Sie die Funktion mithilfe der Lambda-Konsole AWS CLI, der Lambda-API oder eines AWS SDK löschen.

Sie können auch eine bestimmte Version einer Lambda-Funktion löschen, wenn der Version keine CloudFront Distributionen zugeordnet sind. Warten Sie einige Stunden, nachdem Sie alle Zuordnungen für eine Lambda-Funktionsversion entfernt haben. Dann können Sie die Funktionsversion löschen.

Lambda@Edge-Ereignisstruktur

In den folgenden Themen werden die Anforderungs- und Antwortereignisobjekte beschrieben, die CloudFront an eine Lambda @Edge -Funktion übergeben werden, wenn sie ausgelöst wird.

Themen

- Dynamische Ursprungsauswahl
- Anforderungsereignisse
- Antwortereignisse

Dynamische Ursprungsauswahl

Sie können das Pfadmuster in einem Cache-Verhalten verwenden, um Anforderungen an einen Ursprung zu leiten, basierend auf dem Pfad und Namen des angeforderten Objekts, z. B. `images/*.jpg`. Mit Lambda@Edge können Sie Anforderungen auch auf der Grundlage anderer Merkmale, wie z. B. der Werte in Anforderungs-Headern, an einen Ursprung weiterleiten.

Es gibt eine Reihe von Möglichkeiten, wie diese dynamische Ursprungsauswahl genutzt werden kann. So können Sie z. B. Anforderungen über Ursprünge in verschiedenen geografischen Gebieten verteilen, um den globalen Lastausgleich zu unterstützen. Oder Sie können Anforderungen selektiv an verschiedene Ursprünge weiterleiten, die jeweils eine bestimmte Funktion erfüllen: Bot-Handling, SEO-Optimierung, Authentifizierung und so weiter. Code-Beispiele, die die Verwendung dieses Features demonstrieren, finden Sie unter Content-based dynamische Ursprungsauswahl — Beispiele.

Im ursprünglichen CloudFront Anforderungsereignis enthält das `origin` Objekt in der Ereignisstruktur Informationen über den Ursprung, an den die Anforderung weitergeleitet werden

würde, basierend auf dem Pfadmuster. Sie können die Werte im `origin`-Objekt aktualisieren, um eine Anforderung an eine andere Herkunft weiterzuleiten. Wenn Sie das `origin`-Objekt aktualisieren, müssen Sie den Ursprung in der Verteilung nicht definieren. Sie können ein Amazon S3-Ursprungsobjekt auch durch ein benutzerdefiniertes Ursprungsobjekt ersetzen und umgekehrt. Sie können jedoch nur einen einzigen Ursprung pro Anforderung angeben; entweder einen benutzerdefinierten Ursprung oder einen Amazon S3-Ursprung, aber nicht beide.

Anforderungsereignisse

Die folgenden Themen zeigen die Struktur des Objekts, das für Viewer- und CloudFront Origin-Request-Ereignisse an eine Lambda-Funktion übergeben wird. Diese Beispiele zeigen eine `GET`-Anfrage ohne Körper. Im Anschluss an die Beispiele finden Sie eine Liste aller möglichen Felder in Viewer- und Ursprungsanforderungsereignissen.

Themen

- Beispiel für Viewer-Anforderung
- Beispiel für Ursprungsanforderung
- Anforderungsereignisfelder

Beispiel für Viewer-Anforderung

Das folgende Beispiel zeigt ein Viewer-Anfrageereignisobjekt.

```
{
  "Records": [
    {
      "cf": {
        "config": {
          "distributionDomainName": "d111111abcdef8.cloudfront.net",
          "distributionId": "EDFDVBD6EXAMPLE",
          "eventType": "viewer-request",
          "requestId": "4TyzHTaYWb1GX1qTfsHhEqV6HUDd_BzoBZnwfnvQc_1oF26ClkoUSEQ=="
        },
        "request": {
          "clientIp": "203.0.113.178",
          "headers": {
            "host": [
              {
                "key": "Host",
                "value": "d111111abcdef8.cloudfront.net"
```

```
            }
          ],
          "user-agent": [
            {
              "key": "User-Agent",
              "value": "curl/7.66.0"
            }
          ],
          "accept": [
            {
              "key": "accept",
              "value": "*/*"
            }
          ]
        },
        "method": "GET",
        "querystring": "",
        "uri": "/"
      }
    }
  }
 ]
}
```

Beispiel für Ursprungsanforderung

Das folgende Beispiel zeigt ein Ursprungsanforderungsereignisobjekt.

```
{
  "Records": [
    {
      "cf": {
        "config": {
          "distributionDomainName": "d111111abcdef8.cloudfront.net",
          "distributionId": "EDFDVBD6EXAMPLE",
          "eventType": "origin-request",
          "requestId": "4TyzHTaYWb1GX1qTfsHhEqV6HUDd_BzoBZnwfnvQc_1oF26ClkoUSEQ=="
        },
        "request": {
          "clientIp": "203.0.113.178",
          "headers": {
            "x-forwarded-for": [
              {
                "key": "X-Forwarded-For",
```

```
                "value": "203.0.113.178"
              }
            ],
            "user-agent": [
              {
                "key": "User-Agent",
                "value": "Amazon CloudFront"
              }
            ],
            "via": [
              {
                "key": "Via",
                "value": "2.0 2afae0d44e2540f472c0635ab62c232b.cloudfront.net
(CloudFront)"
              }
            ],
            "host": [
              {
                "key": "Host",
                "value": "example.org"
              }
            ],
            "cache-control": [
              {
                "key": "Cache-Control",
                "value": "no-cache"
              }
            ]
          },
          "method": "GET",
          "origin": {
            "custom": {
              "customHeaders": {},
              "domainName": "example.org",
              "keepaliveTimeout": 5,
              "path": "",
              "port": 443,
              "protocol": "https",
              "readTimeout": 30,
              "responseCompletionTimeout": 30,
              "sslProtocols": [
                "TLSv1",
                "TLSv1.1",
                "TLSv1.2"
```

```
            ]
          }
        },
        "querystring": "",
        "uri": "/"
      }
    }
  }
 ]
}
```

Anforderungsereignisfelder

Anforderungsereignisobjektdaten sind in zwei Unterobjekten enthalten: `config` (`Records.cf.config`) und `request` (`Records.cf.request`). In den folgenden Listen werden die Felder jedes Unterobjekts beschrieben.

Felder im Config-Objekt

Die folgende Liste beschreibt die Felder im `config`-Objekt (`Records.cf.config`).

`distributionDomainName` (Schreibgeschützt)

Der Domänenname der Verteilung, die der Anforderung zugeordnet ist.

`distributionID` (Schreibgeschützt)

Die ID der Verteilung, die der Anforderung zugeordnet ist.

`eventType` (Schreibgeschützt)

Der Typ des Auslösers, der der Anforderung zugeordnet ist: `viewer-request` oder `origin-request`.

`requestId` (Schreibgeschützt)

Eine verschlüsselte Zeichenfolge, welche eine Betrachter-an-CloudFront-Anfrage eindeutig bezeichnet. Der `requestId`-Wert erscheint auch als CloudFront in `x-edge-request-id`-Zugriffsprotokollen. Weitere Informationen erhalten Sie unter Zugriffsprotokolle (Standardprotokolle) und Felder der Protokolldatei.

Felder im Anforderungsobjekt

Die folgende Liste beschreibt die Felder im `request`-Objekt (`Records.cf.request`).

`clientIp` (Schreibgeschützt)

Die IP-Adresse des Viewers, der die Anforderung gestellt hat. Wenn der Viewer einen HTTP-Proxy oder einen Load Balancer verwendet hat, um die Anfrage zu senden, entspricht der Wert der IP-Adresse des Proxys bzw. des Load Balancers.

Header () read/write

Die Header der Anforderung. Beachten Sie Folgendes:

- Die Schlüssel im `headers`-Objekt sind kleingeschriebene Versionen von Standard-HTTP-Header-Namen. Über diese Kleinbuchstaben-Schlüssel haben Sie Zugriff auf die Headerwerte (ohne Berücksichtigung von Groß-/Kleinschreibung).
- Jedes Header-Objekt (z. B. `headers["accept"]` oder `headers["host"]`) ist ein Array mit Schlüssel-Wert-Paaren. Für einen bestimmten Header enthält das Array ein Schlüssel-Wert-Paar für jeden Wert in der Anforderung.
- `key` enthält den Namen des Headers unter Beachtung der Groß-/Kleinschreibung, wie er in der HTTP-Anforderung angezeigt wurde, z. B. `Host`, `User-Agent`, `X-Forwarded-For`, `Cookie`, usw.
- `value` enthält den Header-Wert, wie er in der HTTP-Anforderung angezeigt wird.
- Wenn Ihre Lambda-Funktion Anfrage-Header hinzufügt oder ändert und Sie das `key`-Header-Feld nicht einschließen, fügt Lambda@Edge automatisch den Header `key` mit dem von Ihnen angegebenen Header-Namen ein. Unabhängig davon, wie Sie den Header-Namen formatiert haben, wird der automatisch eingefügte Header-Schlüssel mit einem großen Anfangsbuchstaben für jeden Teil formatiert, wobei die einzelnen Teile durch Bindestriche (-) getrennt werden.

  Beispielsweise können Sie einen Header wie den folgenden ohne Header hinzufügen: `key`

  ```
  "user-agent": [
    {
      "value": "ExampleCustomUserAgent/1.X.0"
    }
  ]
  ```

  In diesem Beispiel fügt Lambda@Edge automatisch `"key": "User-Agent"` ein.

Weitere Informationen zu Einschränkungen zur Verwendung von Headern finden Sie unter Einschränkungen für Edge-Funktionen.

`method` (Schreibgeschützt)

Die HTTP-Methode der Anforderung.

`querystring` (read/write)

Die Abfragezeichenfolge, falls vorhanden, in der Anforderung. Wenn die Anfrage keine Abfragezeichenfolge enthält, umfasst das Ereignisobjekt dennoch `querystring` mit einem leeren Wert. Weitere Informationen zu Abfragezeichenfolgen finden Sie unter Zwischenspeichern von Inhalten auf der Grundlage von Abfragezeichenfolgeparametern.

`uri` (read/write)

Der relative Pfad des angeforderten Objekts. Wenn Ihre Lambda-Funktion den `uri`-Wert ändert, beachten Sie Folgendes:

- Der neue `uri`-Wert muss mit einem Schrägstrich (/) beginnen.
- Wenn eine Funktion den `uri`-Wert ändert, ändert sich auch das Objekt, das die Betrachter anfordert.
- Wenn eine Funktion den `uri`-Wert ändert, ändert sich weder das Cache-Verhalten für die Anforderung noch der Ursprung, an den die Anforderung weitergeleitet wird.

`body` (read/write)

Der Hauptteil der HTTP-Anforderung. Die `body`-Struktur kann folgende Felder enthalten:

`inputTruncated` (Schreibgeschützt)

Ein boolesches Flag, das angibt, ob der Textkörper von Lambda@Edge abgeschnitten wurde. Weitere Informationen finden Sie unter Einschränkungen für Anforderungstext mit der Option „Text einschließen“.

`action` (read/write)

Die Aktion, die Sie für den Body durchführen möchten. Für `action` gibt es die folgenden Optionen:

- `read-only:` Dies ist die Standardeinstellung. Lambda@Edge ignoriert bei der Rückgaben der Antwort von der Lambda-Funktion alle Änderungen an `encoding` oder `data`. wenn `action` schreibgeschützt ist.
- `replace:` Geben Sie diese Option an, wenn Sie den an den Ursprung gesendeten Textkörper ersetzen möchten.

`encoding` (read/write)

Die Kodierung für den Body. Wenn Lambda@Edge den Textkörper der Lambda-Funktion bereitstellt, konvertiert es den Textkörper zuerst zu base64-encoding. Wenn Sie `replace` für die `action` auswählen, um den Textkörper zu ersetzen, können Sie die Kodierung `base64` (Standard) oder `text` verwenden. Wenn Sie `encoding` als `base64` festlegen, der Body jedoch kein gültiges base64 ist, gibt CloudFront einen Fehler zurück.

`data` (read/write)

Der Inhalt des Anforderungs-Bodys.

`origin`(read/write) (nur Ausgangsereignisse)

Der Ursprung, an den die Anfrage gesendet werden soll. Die `origin`-Struktur muss genau einen Ursprung enthalten, der ein benutzerdefinierter Ursprung oder ein Amazon S3-Ursprung sein kann.

Je nachdem, welchen Ursprungstyp Sie angeben (benutzerdefiniert oder Amazon-S3-Ursprünge), müssen Sie in Ihrer Anforderung die folgenden Felder angeben:

`customHeaders`(read/write) (Benutzerdefiniert und Amazon S3 S3-Ursprünge)

(Optional) Sie können benutzerdefinierte Header in die Anforderung aufnehmen, indem Sie für jeden benutzerdefinierten Header einen Header-Namens- und Wertpaar angeben. Sie können keine Header hinzufügen, die nicht erlaubt sind, und ein Header mit dem gleichen Namen kann in `Records.cf.request.headers` nicht vorhanden sein. Die Hinweise zu Anforderungs-Headern gelten auch für benutzerdefinierte Header. Weitere Informationen erhalten Sie unter Benutzerdefinierte Header, die nicht zu CloudFront Quellanfragen hinzugefügt werden können und Einschränkungen für Edge-Funktionen.

`domainName`(read/write) (Benutzerdefiniert und Amazon S3 S3-Ursprünge)

Der Domänenname des Ursprungs. Der Domänenname darf nicht leer sein.

- Für benutzerdefinierte Ursprünge – Geben Sie einen DNS-Domänennamen an, z. B. `www.example.com` Der Domänenname darf keinen Doppelpunkt (:) enthalten und keine IP-Adresse sein. Der Domänenname kann bis zu 253 Zeichen lang sein.
- Für Amazon S3-Ursprünge – Geben Sie den DNS-Domänennamen des Amazon S3-Buckets an, z. B. `amzn-s3-demo-bucket.s3.eu-west-1.amazonaws.com`. Der Name kann bis zu 128 Zeichen lang sein und muss in Kleinbuchstaben geschrieben werden.

`path`(read/write) (Benutzerdefiniert und Amazon S3 S3-Ursprünge)

Der Verzeichnispfad auf dem Ursprung, aus dem die Anforderung den Inhalt abrufen soll. Der Pfad sollte mit einem Schrägstrich (/) beginnen, aber nicht damit enden (z. B. nicht mit `example-path/`). Nur für benutzerdefinierte Ursprünge sollte der Pfad URL-codiert sein und eine Maximallänge von 255 Zeichen haben.

`keepaliveTimeout`(read/write) (nur benutzerdefinierte Ursprünge)

Wie lange (in Sekunden) CloudFront soll versucht werden, die Verbindung zum Ursprung aufrechtzuerhalten, nachdem das letzte Paket der Antwort empfangen wurde. Der Wert muss eine Zahl zwischen 1—120 (einschließlich) sein.

`port`(read/write) (nur benutzerdefinierte Ursprünge)

Der Port, zu dem eine Verbindung zu Ihrem benutzerdefinierten Ursprung hergestellt werden CloudFront soll. Der Port muss 80, 443 oder 1024 bis einschließlich 65535 sein.

`protocol`(read/write) (nur benutzerdefinierte Ursprünge)

Das Verbindungsprotokoll, das für die Verbindung mit Ihrem Origin verwendet werden CloudFront sollte. Dabei kann es sich um den Wert `http` oder `https` handeln.

`readTimeout`(read/write) (Benutzerdefiniert und Amazon S3 S3-Ursprünge)

Wie lange (in Sekunden) CloudFront sollten Sie auf eine Antwort warten, nachdem Sie eine Anfrage an Ihren Absender gesendet haben. Dies gibt auch an, wie lange CloudFront warten soll, nachdem das Paket einer Antwort empfangen wurde, bevor das nächste Paket empfangen wird. Der Wert muss eine Zahl zwischen 1—120 (einschließlich) sein.

Wenn Sie ein höheres Kontingent benötigen, finden Sie weitere Informationen unter Antwort-Timeout pro Ursprung.

`responseCompletionTimeout`(read/write) (Benutzerdefiniert und Amazon S3 S3-Ursprünge)

Die Zeit (in Sekunden), in der eine Anfrage vom CloudFront Ursprung geöffnet bleiben und auf eine Antwort warten kann. Wenn bis zu diesem Zeitpunkt noch keine vollständige Antwort vom Ursprung eingegangen ist, wird die Verbindung CloudFront beendet.

Der Wert für `responseCompletionTimeout` muss größer oder gleich dem für `readTimeout` angegebenen Wert sein. Wenn Sie diesen Wert auf 0 setzen, werden alle zuvor von Ihnen festgelegten Werte entfernt und der Standardwert wird wiederhergestellt. Sie können dies auch erreichen, indem Sie das Feld `responseCompletionTimeout` aus der Ereignisanforderung löschen.

`sslProtocols`(read/write) (nur benutzerdefinierte Ursprünge)

Das SSL/TLS Mindestprotokoll, das Sie verwenden CloudFront können, wenn Sie eine HTTPS-Verbindung mit Ihrem Ursprung herstellen. Werte können einer der folgenden sein: `TLSv1.2`, `TLSv1.1`, `TLSv1` oder `SSLv3`.

`authMethod`(read/write) (Nur Amazon S3 S3-Ursprünge)

Wenn Sie eine Ursprungszugriffsidentität (OAI) verwenden, setzen Sie dieses Feld auf `origin-access-identity`. Wenn Sie keine OAI verwenden, setzen Sie es auf `none`. Wenn Sie `authMethod` auf „`origin-access-identity`“ festlegen, gibt es mehrere Anforderungen:

- Sie müssen die `region` angeben (siehe das folgende Feld).
- Sie müssen dieselbe OAI verwenden, wenn Sie die Anforderung von einem Amazon S3-Ursprung zu einem anderen ändern.
- Sie können keine OAI verwenden, wenn Sie die Anforderung von einem benutzerdefinierten Ursprung zu einem Amazon-S3-Ursprung ändern.

> **Note**
>
> Dieses Feld unterstützt keine Ursprungszugriffssteuerung (OAC).

`region`(read/write) (Nur Amazon S3 S3-Ursprünge)

Die AWS Region Ihres Amazon S3 S3-Buckets. Dies ist nur erforderlich, wenn Sie `authMethod` auf „`origin-access-identity`“ festlegen.

Antwortereignisse

Die folgenden Themen zeigen die Struktur des Objekts, das für Viewer- und Origin-Response-Ereignisse an eine Lambda-Funktion CloudFront übergeben wird. Im Anschluss an die Beispiele finden Sie eine Liste aller möglichen Felder in Viewer- und Ursprungantwortereignissen.

Themen

- Beispiel für Ursprungsantwort
- Beispiel für Viewer-Antwort
- Antwortereignisfelder

Beispiel für Ursprungsantwort

Das folgende Beispiel zeigt ein Ursprungsantwortereignisobjekt.

```
{
  "Records": [
    {
      "cf": {
        "config": {
          "distributionDomainName": "d111111abcdef8.cloudfront.net",
          "distributionId": "EDFDVBD6EXAMPLE",
          "eventType": "origin-response",
          "requestId": "4TyzHTaYWb1GX1qTfsHhEqV6HUDd_BzoBZnwfnvQc_1oF26ClkoUSEQ=="
        },
        "request": {
          "clientIp": "203.0.113.178",
          "headers": {
            "x-forwarded-for": [
              {
                "key": "X-Forwarded-For",
                "value": "203.0.113.178"
              }
            ],
            "user-agent": [
              {
                "key": "User-Agent",
                "value": "Amazon CloudFront"
              }
            ],
            "via": [
              {
                "key": "Via",
                "value": "2.0 8f22423015641505b8c857a37450d6c0.cloudfront.net
 (CloudFront)"
              }
            ],
            "host": [
              {
                "key": "Host",
                "value": "example.org"
              }
            ],
            "cache-control": [
              {
```

```
            "key": "Cache-Control",
            "value": "no-cache"
          }
        ]
      },
      "method": "GET",
      "origin": {
        "custom": {
          "customHeaders": {},
          "domainName": "example.org",
          "keepaliveTimeout": 5,
          "path": "",
          "port": 443,
          "protocol": "https",
          "readTimeout": 30,
          "responseCompletionTimeout": 30,
          "sslProtocols": [
            "TLSv1",
            "TLSv1.1",
            "TLSv1.2"
          ]
        }
      },
      "querystring": "",
      "uri": "/"
    },
    "response": {
      "headers": {
        "access-control-allow-credentials": [
          {
            "key": "Access-Control-Allow-Credentials",
            "value": "true"
          }
        ],
        "access-control-allow-origin": [
          {
            "key": "Access-Control-Allow-Origin",
            "value": "*"
          }
        ],
        "date": [
          {
            "key": "Date",
            "value": "Mon, 13 Jan 2020 20:12:38 GMT"
```

```
        }
      ],
      "referrer-policy": [
        {
          "key": "Referrer-Policy",
          "value": "no-referrer-when-downgrade"
        }
      ],
      "server": [
        {
          "key": "Server",
          "value": "ExampleCustomOriginServer"
        }
      ],
      "x-content-type-options": [
        {
          "key": "X-Content-Type-Options",
          "value": "nosniff"
        }
      ],
      "x-frame-options": [
        {
          "key": "X-Frame-Options",
          "value": "DENY"
        }
      ],
      "x-xss-protection": [
        {
          "key": "X-XSS-Protection",
          "value": "1; mode=block"
        }
      ],
      "content-type": [
        {
          "key": "Content-Type",
          "value": "text/html; charset=utf-8"
        }
      ],
      "content-length": [
        {
          "key": "Content-Length",
          "value": "9593"
        }
      ]
```

```
          },
          "status": "200",
          "statusDescription": "OK"
        }
      }
    }
  ]
}
```

Beispiel für Viewer-Antwort

Das folgende Beispiel zeigt ein Viewer-Antwortereignisobjekt.

```
{
  "Records": [
    {
      "cf": {
        "config": {
          "distributionDomainName": "d111111abcdef8.cloudfront.net",
          "distributionId": "EDFDVBD6EXAMPLE",
          "eventType": "viewer-response",
          "requestId": "4TyzHTaYWb1GX1qTfsHhEqV6HUDd_BzoBZnwfnvQc_1oF26ClkoUSEQ=="
        },
        "request": {
          "clientIp": "203.0.113.178",
          "headers": {
            "host": [
              {
                "key": "Host",
                "value": "d111111abcdef8.cloudfront.net"
              }
            ],
            "user-agent": [
              {
                "key": "User-Agent",
                "value": "curl/7.66.0"
              }
            ],
            "accept": [
              {
                "key": "accept",
                "value": "*/*"
              }
            ]
```

```
        },
        "method": "GET",
        "querystring": "",
        "uri": "/"
      },
      "response": {
        "headers": {
          "access-control-allow-credentials": [
            {
              "key": "Access-Control-Allow-Credentials",
              "value": "true"
            }
          ],
          "access-control-allow-origin": [
            {
              "key": "Access-Control-Allow-Origin",
              "value": "*"
            }
          ],
          "date": [
            {
              "key": "Date",
              "value": "Mon, 13 Jan 2020 20:14:56 GMT"
            }
          ],
          "referrer-policy": [
            {
              "key": "Referrer-Policy",
              "value": "no-referrer-when-downgrade"
            }
          ],
          "server": [
            {
              "key": "Server",
              "value": "ExampleCustomOriginServer"
            }
          ],
          "x-content-type-options": [
            {
              "key": "X-Content-Type-Options",
              "value": "nosniff"
            }
          ],
          "x-frame-options": [
```

```
            {
              "key": "X-Frame-Options",
              "value": "DENY"
            }
          ],
          "x-xss-protection": [
            {
              "key": "X-XSS-Protection",
              "value": "1; mode=block"
            }
          ],
          "age": [
            {
              "key": "Age",
              "value": "2402"
            }
          ],
          "content-type": [
            {
              "key": "Content-Type",
              "value": "text/html; charset=utf-8"
            }
          ],
          "content-length": [
            {
              "key": "Content-Length",
              "value": "9593"
            }
          ]
        },
        "status": "200",
        "statusDescription": "OK"
      }
    }
  }
]
}
```

Antwortereignisfelder

Die Daten des Antwortereignisobjekts sind in drei Unterobjekten enthalten: `config` (`Records.cf.config`), `request` (`Records.cf.request`) und `response` (`Records.cf.response`). Weitere Hinweise zu den Feldern im Anforderungsobjekt finden Sie

unter Felder im Anforderungsobjekt. In den folgenden Listen werden die Felder in den Unterobjekten `config` und `response` beschrieben.

Felder im Config-Objekt

Die folgende Liste beschreibt die Felder im `config`-Objekt (`Records.cf.config`).

`distributionDomainName` (Schreibgeschützt)

Der Domänenname der Verteilung, die der Antwort zugeordnet ist.

`distributionID` (Schreibgeschützt)

Die ID der Verteilung, die der Antwort zugeordnet ist.

`eventType` (Schreibgeschützt)

Der Typ des Auslösers, der der Antwort zugeordnet ist: `origin-response` oder `viewer-response`.

`requestId` (Schreibgeschützt)

Eine verschlüsselte Zeichenfolge, die eindeutig die CloudFront Anfrage des Betrachters identifiziert, der diese Antwort zugeordnet ist. Der `requestId` Wert erscheint auch in den CloudFront Zugriffsprotokollen als. `x-edge-request-id` Weitere Informationen erhalten Sie unter Zugriffsprotokolle (Standardprotokolle) und Felder der Protokolldatei.

Felder im Antwortobjekt

Die folgende Liste beschreibt die Felder im `response`-Objekt (`Records.cf.response`). Hinweise zum Generieren einer HTTP-Antwort mithilfe einer Lambda@Edge-Funktion finden Sie unter Generieren von HTTP-Antworten in Anforderungsauslösern.

`headers` (read/write)

Die Header der Antwort. Beachten Sie Folgendes:

- Die Schlüssel im `headers`-Objekt sind kleingeschriebene Versionen von Standard-HTTP-Header-Namen. Über diese Kleinbuchstaben-Schlüssel haben Sie Zugriff auf die Headerwerte (ohne Berücksichtigung von Groß-/Kleinschreibung).
- Jedes Header-Objekt (z. B. `headers["content-type"]` oder `headers["content-length"]`) ist ein Array mit Schlüssel-Wert-Paaren. Für einen bestimmten Header enthält das Array ein Schlüssel-Wert-Paar für jeden Wert in der generierten Antwort.

- `key` enthält den Namen des Headers unter Beachtung der Groß-/Kleinschreibung, wie er in der HTTP-Antwort angezeigt wird, z. B. `Content-Type`, `Content-Length`, `Cookie` usw.
- `value` enthält den Header-Wert, wie er in der HTTP-Antwort angezeigt wird.
- Wenn Ihre Lambda-Funktion Antwort-Header hinzufügt oder ändert und Sie das `key`-Header-Feld nicht einschließen, fügt Lambda@Edge automatisch den Header `key` mit dem von Ihnen angegebenen Header-Namen ein. Unabhängig davon, wie Sie den Header-Namen formatiert haben, wird der automatisch eingefügte Header-Schlüssel mit einem großen Anfangsbuchstaben für jeden Teil formatiert, wobei die einzelnen Teile durch Bindestriche (-) getrennt werden.

  Beispielsweise können Sie einen Header wie den folgenden ohne Header hinzufügen: `key`

  ```
  "content-type": [
    {
      "value": "text/html;charset=UTF-8"
    }
  ]
  ```

  In diesem Beispiel fügt Lambda@Edge automatisch `"key": "Content-Type"` ein.

Weitere Informationen zu Einschränkungen zur Verwendung von Headern finden Sie unter Einschränkungen für Edge-Funktionen.

`status`

Den HTTP-Statuscode der Antwort.

`statusDescription`

Die HTTP-Statusbeschreibung der Antwort.

Arbeiten mit Anforderungen und Antworten

Informationen zur Verwendung von Lambda@Edge-Anforderungen und -Antworten finden Sie in den folgenden Themen:

Themen

- Verwenden von Lambda@Edge-Funktionen mit Origin Failover
- Generieren von HTTP-Antworten in Anforderungsauslösern
- Aktualisieren von HTTP-Antworten in Ursprungsantwortauslösern

- Zugriff auf den Anforderungstext über die Option „Text einschließen“

Verwenden von Lambda@Edge-Funktionen mit Origin Failover

Sie können Lambda @Edge -Funktionen mit CloudFront Distributionen verwenden, die Sie mit Ursprungsgruppen eingerichtet haben, z. B. für Origin-Failover, das Sie konfigurieren, um eine hohe Verfügbarkeit sicherzustellen. Um eine Lambda-Funktion mit einer Ursprungsgruppe zu verwenden, geben Sie die Funktion in einem Ursprungsanfragen- oder Ursprungsantwort-Auslöser für eine Ursprungsgruppe an, wenn Sie das Zwischenspeicher-Verhalten erstellen.

Weitere Informationen finden Sie hier:

- Erstellen von Ursprungsgruppen: Erstellen einer Ursprungsgruppe
- Funktionsweise von Origin Failover mit Lambda@Edge: Verwenden von Origin Failover mit Lambda@Edge-Funktionen

Generieren von HTTP-Antworten in Anforderungsauslösern

Wenn Sie CloudFront eine Anfrage erhalten, können Sie eine Lambda-Funktion verwenden, um eine HTTP-Antwort zu generieren, die direkt an den Betrachter CloudFront zurückkehrt, ohne die Antwort an den Ursprung weiterzuleiten. Die Generierung von HTTP-Antworten reduziert die Auslastung des Ursprungs und reduziert in der Regel auch die Latenzzeit für den Viewer.

Einige häufige Szenarien für die Generierung von HTTP-Antworten sind die folgenden:

- Rückgabe einer kleinen Website an den Viewer.
- Rückgabe eines HTTP 301- oder 302-Statuscodes, um den Benutzer auf eine andere Webseite umzuleiten.
- Rückgabe eines HTTP 401-Statuscodes an den Viewer, wenn sich der Benutzer nicht authentifiziert hat.

Eine Lambda@Edge-Funktion kann eine HTTP-Antwort generieren, wenn die folgenden CloudFront-Ereignisse auftreten:

Viewer-Anforderungsereignisse

> Wenn eine Funktion durch ein Viewer-Anforderungsereignis ausgelöst wird, CloudFront gibt sie die Antwort an den Betrachter zurück und speichert sie nicht im Cache.

Ursprungsanforderungsereignisse

Wenn eine Funktion durch ein Origin-Anforderungsereignis ausgelöst wird, CloudFront wird im Edge-Cache nach einer Antwort gesucht, die zuvor von der Funktion generiert wurde.

- Wenn sich die Antwort im Cache befindet, wird die Funktion nicht ausgeführt und CloudFront gibt die zwischengespeicherte Antwort an den Viewer zurück.
- Wenn sich die Antwort nicht im Cache befindet, wird die Funktion ausgeführt, CloudFront gibt die Antwort an den Betrachter zurück und speichert sie ebenfalls im Cache.

Beispiel-Code zum Generieren von HTTP-Antworten finden Sie unter Beispielfunktionen für Lambda@Edge. Sie können auch die HTTP-Antworten in Antwortauslösern ersetzen. Weitere Informationen finden Sie unter Aktualisieren von HTTP-Antworten in Ursprungsantwortauslösern.

Programmiermodell

Dieser Abschnitt enthält Informationen über das Programmiermodell für die Nutzung von Lambda@Edge zum Generieren von HTTP-Antworten.

Themen

- Antwortobjekt
- Fehler
- Pflichtfelder

Antwortobjekt

Die Antwort, die Sie als `result`-Parameter der `callback`-Methode zurückgeben, sollte folgenden Aufbau haben (beachten Sie, dass nur das `status`-Feld erforderlich ist).

```
const response = {
    body: 'content',
    bodyEncoding: 'text' | 'base64',
    headers: {
        'header name in lowercase': [{
            key: 'header name in standard case',
            value: 'header value'
         }],
         ...
    },
    status: 'HTTP status code (string)',
```

```
    statusDescription: 'status description'
};
```

Das Antwortobjekt kann die folgenden Werte enthalten:

`body`

Der Text, falls vorhanden, den Sie in der generierten Antwort zurückgeben CloudFront möchten.

`bodyEncoding`

Die Kodierung für den Wert, den Sie in `body` festgelegt haben. Die einzigen gültigen Codierungen sind `text` und `base64`. Wenn Sie `body` in das `response` Objekt aufnehmen, aber weglassen`bodyEncoding`, wird der CloudFront Hauptteil als Text behandelt.

Wenn Sie `bodyEncoding` als `base64` festlegen, der Body jedoch kein gültiges base64 ist, gibt CloudFront einen Fehler zurück.

`headers`

Header, die Sie in der generierten Antwort zurückgeben möchten CloudFront . Beachten Sie Folgendes:

- Die Schlüssel im `headers`-Objekt sind kleingeschriebene Versionen von Standard-HTTP-Header-Namen. Über diese Kleinbuchstaben-Schlüssel haben Sie Zugriff auf die Headerwerte (ohne Berücksichtigung von Groß-/Kleinschreibung).
- Jeder Header (z. B. `headers["accept"]` oder `headers["host"]`) ist ein Array mit Schlüssel-Wert-Paaren. Für einen bestimmten Header enthält das Array ein Schlüssel-Wert-Paar für jeden Wert in der generierten Antwort.
- `key` (optional) ist der von Groß- und Kleinschreibung unabhängige Name des Headers, wie er in einer HTTP-Anforderung erscheint (z. B. `accept` oder `host`).
- Geben Sie `value` als Header-Wert an.
- Wenn Sie den Header-Schlüsselteil des Schlüssel-Wert-Paares nicht aufnehmen, fügt Lambda@Edge automatisch einen Header-Schlüssel mit dem von Ihnen angegebenen Header-Namen ein. Unabhängig davon, wie Sie den Header-Namen formatiert haben, wird der automatisch eingefügte Header-Schlüssel mit einem großen Anfangsbuchstaben für jeden Teil formatiert, wobei die einzelnen Teile durch Bindestriche (-) getrennt werden.

  Beispielsweise können Sie einen Header wie den folgenden ohne Header-Schlüssel hinzufügen: `'content-type': [{ value: 'text/html;charset=UTF-8' }]`

In diesem Beispiel erstellt Lambda@Edge den folgenden Header-Schlüssel: `Content-Type`.

Weitere Informationen zu Einschränkungen zur Verwendung von Headern finden Sie unter Einschränkungen für Edge-Funktionen.

`status`

Den HTTP-Statuscode . Geben Sie den Statuscode als Zeichenfolge an. CloudFront verwendet den bereitgestellten Statuscode für Folgendes:

- Rückgabe in der Antwort
- Cache im CloudFront Edge-Cache, wenn die Antwort durch eine Funktion generiert wurde, die durch ein Origin-Request-Ereignis ausgelöst wurde
- Loggen Sie sich ein CloudFront Zugriffsprotokolle (Standardprotokolle)

Wenn der `status`-Wert nicht zwischen 200 und 599 liegt, gibt CloudFront einen Fehler an den Betrachter zurück.

`statusDescription`

Die Beschreibung, die CloudFront Sie zusammen mit dem HTTP-Statuscode in der Antwort zurückgeben möchten. Sie müssen keine Standardbeschreibungen verwenden (z. B. `OK` für einen HTTP-Statuscode 200). Die Aktualisierung dieses Felds funktioniert nur, wenn der Client eine Verbindung herstellt HTTP/1.1. Wenn der Client eine Verbindung herstelltHTTP/2, `statusDescription` werden alle Änderungen an der ignoriert und die Standardbeschreibung wird an den Client zurückgegeben.

Fehler

Es folgen mögliche Fehler für generierte HTTP-Antworten.

Antwort enthält einen Body und legt 204 (No Content) als Status fest

Wenn eine Funktion durch eine Viewer-Anfrage ausgelöst wird, wird ein HTTP-502-Statuscode (Bad Gateway) an den Viewer CloudFront zurückgegeben, wenn beide der folgenden Bedingungen zutreffen:

- Der Wert von `status` ist 204 (No Content)
- Die Antwort enthält einen Wert für `body`

Der Grund hierfür ist, dass Lambda@Edge eine optionale Einschränkung aus RFC 2616 umsetzt, die besagt, dass eine `HTTP 204`-Antwort keinen Nachrichtentext zu enthalten braucht.

Beschränkungen für die Größe der generierten Antwort

Die maximale Größe einer durch eine Lambda-Funktion generierten Antwort hängt von dem Ereignis ab, das die Funktion ausgelöst hat:

- Viewer-Anfrage-Ereignisse – 40 KB
- Ursprungsanfrageereignisse – 1 MB

Wenn die Antwort größer als die zulässige Größe ist, wird ein HTTP 502-Statuscode (Bad Gateway) an den Betrachter CloudFront zurückgegeben.

Pflichtfelder

Das Feld `status` ist ein Pflichtfeld.

Alle anderen Felder sind optional.

Aktualisieren von HTTP-Antworten in Ursprungsantwortauslösern

Wenn eine HTTP-Antwort vom Ursprungsserver CloudFront empfangen wird und ein Origin-Response-Trigger mit dem Cache-Verhalten verknüpft ist, können Sie die HTTP-Antwort so ändern, dass sie überschreibt, was vom Ursprung zurückgegeben wurde.

Einige häufige Szenarien für die Aktualisierung von HTTP-Antworten sind die folgenden:

- Ändern des Status, um einen HTTP 200-Statuscode festzulegen, und Erstellen statischer Body-Inhalte für die Rückgabe an den Viewer, wenn ein Ursprung einen Fehlerstatuscode (4xx oder 5xx) zurückgibt. Einen Beispiel-Code finden Sie unter Beispiel: Mithilfe eines Ursprungsantwortauslösers den Fehlerstatuscode auf 200 aktualisieren.
- Ändern des Status, um einen HTTP 301- oder HTTP 302-Statuscode festzulegen, um den Benutzer auf eine andere Website umzuleiten, wenn ein Ursprung einen Fehlerstatuscode zurückgibt (4xx oder 5xx). Einen Beispiel-Code finden Sie unter Beispiel: Mithilfe eines Ursprungsantwortauslösers den Fehlerstatuscode auf 302 aktualisieren.

> **Note**
>
> Die Funktion muss einen Statuswert zwischen `200` und `599` (einschließlich) zurückgeben, andernfalls CloudFront gibt sie einen Fehler an den Viewer zurück.

Sie können auch die HTTP-Antworten in Viewer- und Ursprungsanfrageereignissen ersetzen. Weitere Informationen finden Sie unter Generieren von HTTP-Antworten in Anforderungsauslösern.

Wenn Sie mit der HTTP-Antwort arbeiten, stellt Lambda@Edge den Textkörper, der vom Ursprungsserver zurückgegeben wird, nicht für den Ursprungsantwortauslöser bereit. Sie können einen statischen Inhaltstext erzeugen, indem Sie ihn auf den gewünschten Wert setzen, oder den Text innerhalb der Funktion entfernen, indem Sie den Wert auf leer setzen. Wenn Sie das Textkörperfeld in Ihrer Funktion nicht aktualisieren, wird der ursprüngliche Textkörper, der vom Ursprungsserver zurückgegeben wird, an den Viewer zurückgegeben.

Zugriff auf den Anforderungstext über die Option „Text einschließen“

Sie können jetzt auswählen, ob Lambda@Edge den Textkörper in einer Anforderung für eine nicht schreibgeschützte HTTP-Methode (POST, PUT, DELETE etc.) weitergeben soll, sodass Sie in Ihrer Lambda-Funktion darauf zugreifen können. Sie können den Lesezugriff wählen oder angeben, dass Sie den Textkörper ersetzen möchten.

Um diese Option zu aktivieren, wählen Sie Include Body (Body einbeziehen) aus, wenn Sie einen CloudFront -Auslöser für Ihre für ein Betrachteranforderungs- oder Ursprungsanforderungsereignis vorgesehene Funktion erstellen. Weitere Informationen finden Sie unter Hinzufügen von Auslösern für eine Lambda@Edge-Funktion. Mehr zur Nutzung von Include Body (Textkörper einbeziehen) mit Ihrer Funktion finden Sie unter Lambda@Edge-Ereignisstruktur.

Sie können dieses Feature in den folgenden Szenarien nutzen:

- Verarbeitung von Webformularen (z. B. "Kontakt"-Formulare) ohne Rückgabe von Kundeneingabedaten an den Ursprungs-Server
- Erfassen von Web-Beacon-Daten, die von Viewer-Browsern gesendet werden, und Verarbeiten am Edge

Einen Beispiel-Code finden Sie unter Beispielfunktionen für Lambda@Edge.

> **Note**
>
> Wenn der Anforderungs-Body groß ist, wird er von Lambda@Edge abgeschnitten. Ausführliche Informationen zur maximalen Größe und Kürzung finden Sie unter Einschränkungen für Anforderungstext mit der Option „Text einschließen“.

Beispielfunktionen für Lambda@Edge

Sehen Sie sich die folgenden Beispiele für die Verwendung von Lambda-Funktionen mit Amazon CloudFront an.

> **Note**
>
> Wenn Sie Runtime Node.js 18 oder höher für Ihre Lambda @Edge -Funktion wählen, wird automatisch eine `index.mjs` Datei für Sie erstellt. Um die folgenden Codebeispiele zu verwenden, benennen Sie die `index.mjs`-Datei in `index.js` um.

Themen

- Allgemeine Beispiele
- Generieren von Antworten – Beispiele
- Abfragezeichenfolgen – Beispiele
- Personalisieren von Inhalten nach Land oder Gerätetyp-Header – Beispiele
- Content-based dynamische Ursprungsauswahl — Beispiele
- Aktualisieren von Fehlerstatusangaben – Beispiele
- Zugriff auf den Anforderungstext – Beispiele

Allgemeine Beispiele

Die folgenden Beispiele zeigen gängige Verwendungsmöglichkeiten von Lambda @Edge in CloudFront.

Themen

- Beispiel: Testen A/B
- Beispiel: Überschreiben eines Antwortheaders

Beispiel: Testen A/B

Sie können das folgende Beispiel verwenden, um zwei verschiedene Versionen eines Images zu testen, ohne Weiterleitungen zu erstellen oder die URL zu ändern. In diesem Beispiel werden die Cookies in der Viewer-Anfrage gelesen und die Anforderungs-URL wird entsprechend geändert.

Wenn der Viewer kein Cookie mit einem der erwarteten Werte sendet, weist das Beispiel den Viewer zufällig einer der URLs zu.

Node.js

```
'use strict';

exports.handler = (event, context, callback) => {
    const request = event.Records[0].cf.request;
    const headers = request.headers;

    if (request.uri !== '/experiment-pixel.jpg') {
        // do not process if this is not an A-B test request
        callback(null, request);
        return;
    }

    const cookieExperimentA = 'X-Experiment-Name=A';
    const cookieExperimentB = 'X-Experiment-Name=B';
    const pathExperimentA = '/experiment-group/control-pixel.jpg';
    const pathExperimentB = '/experiment-group/treatment-pixel.jpg';

    /*
     * Lambda at the Edge headers are array objects.
     *
     * When a client sends multiple headers with the same name, each
     * header is a separate entry in the array. For example:
     * > GET /viewerRes/test HTTP/1.1
     * > X-Custom: valueA
     * > X-Custom: valueB
     *
     * You can access the first at headers["x-custom"][0].value
     * and the second at headers["x-custom"][1].value.
     *
     * CloudFront concatenates multiple Cookie headers into a single entry,
 separated by "; ".
     * For example:
     * > Cookie: First=1; Second=2
     * > Cookie: ClientCode=abc
     *
     * becomes headers["cookie"][0].value = "First=1; Second=2; ClientCode=abc"
     * This behavior is consistent across HTTP/1.1, HTTP/2, and HTTP/3.
     *
```

```
     * Header values are not parsed.
     */
    let experimentUri;
    if (headers.cookie) {
        for (let i = 0; i < headers.cookie.length; i++) {
            if (headers.cookie[i].value.indexOf(cookieExperimentA) >= 0) {
                console.log('Experiment A cookie found');
                experimentUri = pathExperimentA;
                break;
            } else if (headers.cookie[i].value.indexOf(cookieExperimentB) >= 0) {
                console.log('Experiment B cookie found');
                experimentUri = pathExperimentB;
                break;
            }
        }
    }

    if (!experimentUri) {
        console.log('Experiment cookie has not been found. Throwing dice...');
        if (Math.random() < 0.75) {
            experimentUri = pathExperimentA;
        } else {
            experimentUri = pathExperimentB;
        }
    }

    request.uri = experimentUri;
    console.log(`Request uri set to "${request.uri}"`);
    callback(null, request);
};
```

Python

```
import json
import random

def lambda_handler(event, context):
    request = event['Records'][0]['cf']['request']
    headers = request['headers']

    if request['uri'] != '/experiment-pixel.jpg':
        # Not an A/B Test
        return request
```

```
    cookieExperimentA, cookieExperimentB = 'X-Experiment-Name=A', 'X-Experiment-
Name=B'
    pathExperimentA, pathExperimentB = '/experiment-group/control-pixel.jpg', '/
experiment-group/treatment-pixel.jpg'

    '''
    Lambda at the Edge headers are array objects.

    When a client sends multiple headers with the same name, each
    header is a separate entry in the array. For example:
    > GET /viewerRes/test HTTP/1.1
    > X-Custom: valueA
    > X-Custom: valueB

    You can access the first at headers["x-custom"][0].value
    and the second at headers["x-custom"][1].value.

    CloudFront concatenates multiple Cookie headers into a single entry, separated
 by "; ".
    For example:
    > Cookie: First=1; Second=2
    > Cookie: ClientCode=abc

    becomes headers["cookie"][0].value = "First=1; Second=2; ClientCode=abc"
    This behavior is consistent across HTTP/1.1, HTTP/2, and HTTP/3.

    Header values are not parsed.
    '''

    experimentUri = ""

    for cookie in headers.get('cookie', []):
        if cookieExperimentA in cookie['value']:
            print("Experiment A cookie found")
            experimentUri = pathExperimentA
            break
        elif cookieExperimentB in cookie['value']:
            print("Experiment B cookie found")
            experimentUri = pathExperimentB
            break

    if not experimentUri:
        print("Experiment cookie has not been found. Throwing dice...")
```

```
        if random.random() < 0.75:
            experimentUri = pathExperimentA
        else:
            experimentUri = pathExperimentB

    request['uri'] = experimentUri
    print(f"Request uri set to {experimentUri}")
    return request
```

Beispiel: Überschreiben eines Antwortheaders

Das folgende Beispiel zeigt, wie Sie den Wert eines Antwortheaders basierend auf dem Wert eines anderen Headers modifizieren:

Node.js

```
export const handler = async (event) => {
    const response = event.Records[0].cf.response;
    const headers = response.headers;

    const headerNameSrc = 'X-Amz-Meta-Last-Modified';
    const headerNameDst = 'Last-Modified';

    if (headers[headerNameSrc.toLowerCase()]) {
        headers[headerNameDst.toLowerCase()] = [{
            key: headerNameDst,
            value: headers[headerNameSrc.toLowerCase()][0].value,
        }];
        console.log(`Response header "${headerNameDst}" was set to ` +
                    `"${headers[headerNameDst.toLowerCase()][0].value}"`);
    }

    return response;
};
```

Python

```
import json

def lambda_handler(event, context):
    response = event['Records'][0]['cf']['response']
    headers = response['headers']
```

```
    header_name_src = 'X-Amz-Meta-Last-Modified'
    header_name_dst = 'Last-Modified'

    if headers.get(header_name_src.lower()):
        headers[header_name_dst.lower()] = [{
            'key': header_name_dst,
            'value': headers[header_name_src.lower()][0]['value']
        }]
        print(f'Response header "{header_name_dst}" was set to '
              f'"{headers[header_name_dst.lower()][0]["value"]}"')

    return response
```

Generieren von Antworten – Beispiele

Die folgenden Beispiele zeigen, wie Sie Lambda@Edge zum Generieren von Antworten verwenden können.

Themen

- Beispiel: Bereitstellen von statischen Inhalten (generierte Antwort)
- Beispiel: Generieren einer HTTP-Umleitung (generierte Antwort)

Beispiel: Bereitstellen von statischen Inhalten (generierte Antwort)

Das folgende Beispiel zeigt, wie Sie mit einer Lambda-Funktion Inhalte von statischen Websites bereitstellen können, wodurch sich die Verarbeitungslast auf dem Ursprungs-Server und die Latenz insgesamt verringert.

> **Note**
>
> Sie können HTTP-Antworten für Viewer-Anfrage- und Ursprungsanfrageereignisse generieren. Weitere Informationen finden Sie unter the section called “Generieren von HTTP-Antworten in Anforderungsauslösern”.
> Sie können auch den Text der HTTP-Antwort in Ursprungsantwortereignissen ersetzen oder entfernen. Weitere Informationen finden Sie unter the section called “Aktualisieren von HTTP-Antworten in Ursprungsantwortauslösern”.

Node.js

```
'use strict';

const content = `
<\!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>Simple Lambda@Edge Static Content Response</title>
  </head>
  <body>
    <p>Hello from Lambda@Edge!</p>
  </body>
</html>
`;

exports.handler = (event, context, callback) => {
    /*
     * Generate HTTP OK response using 200 status code with HTML body.
     */
    const response = {
        status: '200',
        statusDescription: 'OK',
        headers: {
            'cache-control': [{
                key: 'Cache-Control',
                value: 'max-age=100'
            }],
            'content-type': [{
                key: 'Content-Type',
                value: 'text/html'
            }]
        },
        body: content,
    };
    callback(null, response);
};
```

Python

```
import json
```

```
CONTENT = """
<\!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>Simple Lambda@Edge Static Content Response</title>
</head>
<body>
    <p>Hello from Lambda@Edge!</p>
</body>
</html>
"""

def lambda_handler(event, context):
    # Generate HTTP OK response using 200 status code with HTML body.
    response = {
        'status': '200',
        'statusDescription': 'OK',
        'headers': {
            'cache-control': [
                {
                    'key': 'Cache-Control',
                    'value': 'max-age=100'
                }
            ],
            "content-type": [
                {
                    'key': 'Content-Type',
                    'value': 'text/html'
                }
            ]
        },
        'body': CONTENT
    }
    return response
```

Beispiel: Generieren einer HTTP-Umleitung (generierte Antwort)

Das folgende Beispiel zeigt, wie Sie eine HTTP-Umleitung generieren.

> **Note**
>
> Sie können HTTP-Antworten für Viewer-Anfrage- und Ursprungsanfrageereignisse generieren. Weitere Informationen finden Sie unter Generieren von HTTP-Antworten in Anforderungsauslösern.

Node.js

```
'use strict';

exports.handler = (event, context, callback) => {
    /*
     * Generate HTTP redirect response with 302 status code and Location header.
     */
    const response = {
        status: '302',
        statusDescription: 'Found',
        headers: {
            location: [{
                key: 'Location',
                value: 'https://docs.aws.amazon.com/lambda/latest/dg/lambda-
edge.html',
            }],
        },
    };
    callback(null, response);
};
```

Python

```
def lambda_handler(event, context):

    # Generate HTTP redirect response with 302 status code and Location header.

    response = {
        'status': '302',
        'statusDescription': 'Found',
        'headers': {
            'location': [{
                'key': 'Location',
```

```
                'value': 'https://docs.aws.amazon.com/lambda/latest/dg/lambda-
edge.html'
            }]
        }
    }

    return response
```

Abfragezeichenfolgen – Beispiele

Die folgenden Beispiele zeigen Möglichkeiten zur Verwendung von Lambda@Edge mit Abfragezeichenfolgen.

Themen

- Beispiel: Hinzufügen eines Headers auf Grundlage eines Abfragezeichenfolgeparameters
- Beispiel: Normalisieren von Abfragezeichenfolgeparametern zum Verbessern der Cache-Trefferrate
- Beispiel: Umleiten von nicht authentifizierten Benutzern auf eine Anmeldeseite

Beispiel: Hinzufügen eines Headers auf Grundlage eines Abfragezeichenfolgeparameters

Im folgenden Beispiel wird gezeigt, wie Sie das Schlüssel-Wert-Paar eines Abfragezeichenfolgeparameter abrufen und auf Grundlage dieser Werte einen Header hinzufügen können.

Node.js

```
'use strict';

const querystring = require('querystring');
exports.handler = (event, context, callback) => {
    const request = event.Records[0].cf.request;

    /* When a request contains a query string key-value pair but the origin server
     * expects the value in a header, you can use this Lambda function to
     * convert the key-value pair to a header. Here's what the function does:
     * 1. Parses the query string and gets the key-value pair.
     * 2. Adds a header to the request using the key-value pair that the function
 got in step 1.
```

```
     */

    /* Parse request querystring to get javascript object */
    const params = querystring.parse(request.querystring);

    /* Move auth param from querystring to headers */
    const headerName = 'Auth-Header';
    request.headers[headerName.toLowerCase()] = [{ key: headerName, value:
 params.auth }];
    delete params.auth;

    /* Update request querystring */
    request.querystring = querystring.stringify(params);

    callback(null, request);
};
```

Python

```
from urllib.parse import parse_qs, urlencode

def lambda_handler(event, context):
    request = event['Records'][0]['cf']['request']

    '''
    When a request contains a query string key-value pair but the origin server
    expects the value in a header, you can use this Lambda function to
    convert the key-value pair to a header. Here's what the function does:
        1. Parses the query string and gets the key-value pair.
        2. Adds a header to the request using the key-value pair that the function
 got in step 1.
    '''

    # Parse request querystring to get dictionary/json
    params = {k : v[0] for k, v in parse_qs(request['querystring']).items()}

    # Move auth param from querystring to headers
    headerName = 'Auth-Header'
    request['headers'][headerName.lower()] = [{'key': headerName, 'value':
 params['auth']}]
    del params['auth']

    # Update request querystring
```

```
    request['querystring'] = urlencode(params)

    return request
```

Beispiel: Normalisieren von Abfragezeichenfolgeparametern zum Verbessern der Cache-Trefferrate

Das folgende Beispiel zeigt, wie Sie Ihre Cache-Trefferquote verbessern können, indem Sie die folgenden Änderungen an den Abfragezeichenfolgen vornehmen, CloudFront bevor Anfragen an Ihren Ursprung weitergeleitet werden:

- Alphabetisches Anordnen von Schlüssel-Wert-Paaren nach dem Parametername.
- Ändern der Schreibung von Schlüssel-Wert-Paaren in Kleinschreibung.

Weitere Informationen finden Sie unter Zwischenspeichern von Inhalten auf der Grundlage von Abfragezeichenfolgeparametern.

Node.js

```
'use strict';

const querystring = require('querystring');

exports.handler = (event, context, callback) => {
    const request = event.Records[0].cf.request;
    /* When you configure a distribution to forward query strings to the origin and
     * to cache based on an allowlist of query string parameters, we recommend
     * the following to improve the cache-hit ratio:
     * - Always list parameters in the same order.
     * - Use the same case for parameter names and values.
     *
     * This function normalizes query strings so that parameter names and values
     * are lowercase and parameter names are in alphabetical order.
     *
     * For more information, see:
     * https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/
QueryStringParameters.html
     */

    console.log('Query String: ', request.querystring);

    /* Parse request query string to get javascript object */
```

```
    const params = querystring.parse(request.querystring.toLowerCase());
    const sortedParams = {};

    /* Sort param keys */
    Object.keys(params).sort().forEach(key => {
        sortedParams[key] = params[key];
    });

    /* Update request querystring with normalized  */
    request.querystring = querystring.stringify(sortedParams);

    callback(null, request);
};
```

Python

```
from urllib.parse import parse_qs, urlencode

def lambda_handler(event, context):
    request = event['Records'][0]['cf']['request']
    '''
    When you configure a distribution to forward query strings to the origin and
    to cache based on an allowlist of query string parameters, we recommend
    the following to improve the cache-hit ratio:
    Always list parameters in the same order.
    - Use the same case for parameter names and values.

    This function normalizes query strings so that parameter names and values
    are lowercase and parameter names are in alphabetical order.

    For more information, see:
    https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/
QueryStringParameters.html
    '''
    print("Query string: ", request["querystring"])

    # Parse request query string to get js object
    params = {k : v[0] for k, v in parse_qs(request['querystring'].lower()).items()}

    # Sort param keys
    sortedParams = sorted(params.items(), key=lambda x: x[0])

    # Update request querystring with normalized
```

```
    request['querystring'] = urlencode(sortedParams)

    return request
```

Beispiel: Umleiten von nicht authentifizierten Benutzern auf eine Anmeldeseite

Im folgenden Beispiel wird gezeigt, wie Benutzer zu einer Anmeldeseite umgeleitet werden, wenn sie ihre Anmeldeinformationen nicht eingegeben haben.

Node.js

```
'use strict';

function parseCookies(headers) {
    const parsedCookie = {};
    if (headers.cookie) {
        headers.cookie[0].value.split(';').forEach((cookie) => {
            if (cookie) {
                const parts = cookie.split('=');
                parsedCookie[parts[0].trim()] = parts[1].trim();
            }
        });
    }
    return parsedCookie;
}

exports.handler = (event, context, callback) => {
    const request = event.Records[0].cf.request;
    const headers = request.headers;

    /* Check for session-id in request cookie in viewer-request event,
     * if session-id is absent, redirect the user to sign in page with original
     * request sent as redirect_url in query params.
     */

    /* Check for session-id in cookie, if present then proceed with request */
    const parsedCookies = parseCookies(headers);
    if (parsedCookies && parsedCookies['session-id']) {
        callback(null, request);
        return;
    }
```

```
    /* URI encode the original request to be sent as redirect_url in query params */
    const encodedRedirectUrl = encodeURIComponent(`https://
${headers.host[0].value}${request.uri}?${request.querystring}`);
    const response = {
        status: '302',
        statusDescription: 'Found',
        headers: {
            location: [{
                key: 'Location',
                value: `https://www.example.com/signin?redirect_url=
${encodedRedirectUrl}`,
            }],
        },
    };
    callback(null, response);
};
```

Python

```
import urllib

def parseCookies(headers):
    parsedCookie = {}
    if headers.get('cookie'):
        for cookie in headers['cookie'][0]['value'].split(';'):
            if cookie:
                parts = cookie.split('=')
                parsedCookie[parts[0].strip()] = parts[1].strip()
    return parsedCookie

def lambda_handler(event, context):
    request = event['Records'][0]['cf']['request']
    headers = request['headers']

    '''
    Check for session-id in request cookie in viewer-request event,
    if session-id is absent, redirect the user to sign in page with original
    request sent as redirect_url in query params.
    '''

    # Check for session-id in cookie, if present, then proceed with request
    parsedCookies = parseCookies(headers)
```

```
    if parsedCookies and parsedCookies['session-id']:
        return request

    # URI encode the original request to be sent as redirect_url in query params
    redirectUrl = "https://%s%s?%s" % (headers['host'][0]['value'], request['uri'],
request['querystring'])
    encodedRedirectUrl = urllib.parse.quote_plus(redirectUrl.encode('utf-8'))

    response = {
        'status': '302',
        'statusDescription': 'Found',
        'headers': {
            'location': [{
                'key': 'Location',
                'value': 'https://www.example.com/signin?redirect_url=%s' %
encodedRedirectUrl
            }]
        }
    }
    return response
```

Personalisieren von Inhalten nach Land oder Gerätetyp-Header – Beispiele

Die folgenden Beispiele zeigen, wie Sie mit Lambda@Edge das Verhalten basierend auf dem Standort oder dem Typ des Geräts anpassen können, das vom jeweiligen Viewer verwendet wird.

Themen

- Beispiel: Umleiten von Viewer-Anforderungen an eine länderspezifische URL
- Beispiel: Bereitstellen verschiedener Versionen eines Objekts auf Grundlage des Geräts

Beispiel: Umleiten von Viewer-Anforderungen an eine länderspezifische URL

Im folgenden Beispiel wird gezeigt, wie eine HTTP-Umleitungsantwort mit einer länderspezifischen URL erzeugt und die Antwort an den Viewer zurückgegeben wird. Dies ist nützlich, wenn Sie länderspezifische Antworten bereitstellen möchten. Beispiel:

- Wenn Sie über länderspezifischen Subdomänen verfügen, z. B. us.example.com und tw.example.com, können Sie eine Umleitungsantwort erzeugen, wenn ein Viewer example.com anfordert.

- Wenn Sie Videos streamen, aber keine Rechte für das Streamen von Inhalten in einem bestimmten Land besitzen, können Sie Benutzer in diesem Land auf eine Seite umleiten, auf der erklärt wird, warum sie das Video nicht ansehen können.

Beachten Sie Folgendes:

- Sie müssen Ihre Verteilung so konfigurieren, dass die Zwischenspeicherung auf Grundlage des `CloudFront-Viewer-Country`-Headers erfolgt. Weitere Informationen finden Sie unter Basierend auf den ausgewählten Anforderungsheadern.
- CloudFront fügt den `CloudFront-Viewer-Country` Header nach dem Viewer-Anforderungsereignis hinzu. Wenn Sie dieses Beispiel verwenden möchten, müssen Sie einen Auslöser für das ursprüngliche Anfrageereignis erstellen.

Node.js

```
'use strict';

/* This is an origin request function */
exports.handler = (event, context, callback) => {
    const request = event.Records[0].cf.request;
    const headers = request.headers;

    /*
     * Based on the value of the CloudFront-Viewer-Country header, generate an
     * HTTP status code 302 (Redirect) response, and return a country-specific
     * URL in the Location header.
     * NOTE: 1. You must configure your distribution to cache based on the
     *          CloudFront-Viewer-Country header. For more information, see
     *          https://docs.aws.amazon.com/console/cloudfront/cache-on-selected-
headers
     *       2. CloudFront adds the CloudFront-Viewer-Country header after the
 viewer
     *          request event. To use this example, you must create a trigger for
 the
     *          origin request event.
     */

    let url = 'https://example.com/';
    if (headers['cloudfront-viewer-country']) {
        const countryCode = headers['cloudfront-viewer-country'][0].value;
```

```
        if (countryCode === 'TW') {
            url = 'https://tw.example.com/';
        } else if (countryCode === 'US') {
            url = 'https://us.example.com/';
        }
    }

    const response = {
        status: '302',
        statusDescription: 'Found',
        headers: {
            location: [{
                key: 'Location',
                value: url,
            }],
        },
    };
    callback(null, response);
};
```

Python

```
# This is an origin request function

def lambda_handler(event, context):
    request = event['Records'][0]['cf']['request']
    headers = request['headers']

    '''
    Based on the value of the CloudFront-Viewer-Country header, generate an
    HTTP status code 302 (Redirect) response, and return a country-specific
    URL in the Location header.
    NOTE: 1. You must configure your distribution to cache based on the
            CloudFront-Viewer-Country header. For more information, see
            https://docs.aws.amazon.com/console/cloudfront/cache-on-selected-headers
          2. CloudFront adds the CloudFront-Viewer-Country header after the viewer
            request event. To use this example, you must create a trigger for the
            origin request event.
    '''

    url = 'https://example.com/'
    viewerCountry = headers.get('cloudfront-viewer-country')
    if viewerCountry:
```

```
        countryCode = viewerCountry[0]['value']
        if countryCode == 'TW':
            url = 'https://tw.example.com/'
        elif countryCode == 'US':
            url = 'https://us.example.com/'

    response = {
        'status': '302',
        'statusDescription': 'Found',
        'headers': {
            'location': [{
                'key': 'Location',
                'value': url
            }]
        }
    }

    return response
```

Beispiel: Bereitstellen verschiedener Versionen eines Objekts auf Grundlage des Geräts

Das folgende Beispiel zeigt, wie Sie für verschiedene Versionen eines Objekts auf Grundlage des Gerätetyps, den der Benutzer verwendet (z. B. ein mobiles Gerät oder ein Tablet), bereitstellen. Beachten Sie Folgendes:

- Sie müssen Ihre Verteilung so konfigurieren, dass die Zwischenspeicherung auf Grundlage der `CloudFront-Is-*-Viewer`-Header erfolgt. Weitere Informationen finden Sie unter Basierend auf den ausgewählten Anforderungsheadern.
- CloudFront fügt die `CloudFront-Is-*-Viewer` Header nach dem Viewer-Anforderungsereignis hinzu. Wenn Sie dieses Beispiel verwenden möchten, müssen Sie einen Auslöser für das ursprüngliche Anfrageereignis erstellen.

Node.js

```
'use strict';

/* This is an origin request function */
exports.handler = (event, context, callback) => {
    const request = event.Records[0].cf.request;
    const headers = request.headers;
```

```
    /*
     * Serve different versions of an object based on the device type.
     * NOTE: 1. You must configure your distribution to cache based on the
     *          CloudFront-Is-*-Viewer headers. For more information, see
     *          the following documentation:
     *          https://docs.aws.amazon.com/console/cloudfront/cache-on-selected-
headers
     *          https://docs.aws.amazon.com/console/cloudfront/cache-on-device-type
     *       2. CloudFront adds the CloudFront-Is-*-Viewer headers after the viewer
     *          request event. To use this example, you must create a trigger for
 the
     *          origin request event.
     */

    const desktopPath = '/desktop';
    const mobilePath = '/mobile';
    const tabletPath = '/tablet';
    const smarttvPath = '/smarttv';

    if (headers['cloudfront-is-desktop-viewer']
        && headers['cloudfront-is-desktop-viewer'][0].value === 'true') {
        request.uri = desktopPath + request.uri;
    } else if (headers['cloudfront-is-mobile-viewer']
               && headers['cloudfront-is-mobile-viewer'][0].value === 'true') {
        request.uri = mobilePath + request.uri;
    } else if (headers['cloudfront-is-tablet-viewer']
               && headers['cloudfront-is-tablet-viewer'][0].value === 'true') {
        request.uri = tabletPath + request.uri;
    } else if (headers['cloudfront-is-smarttv-viewer']
               && headers['cloudfront-is-smarttv-viewer'][0].value === 'true') {
        request.uri = smarttvPath + request.uri;
    }
    console.log(`Request uri set to "${request.uri}"`);

    callback(null, request);
};
```

Python

```
# This is an origin request function
def lambda_handler(event, context):
    request = event['Records'][0]['cf']['request']
```

```
    headers = request['headers']

    '''
    Serve different versions of an object based on the device type.
    NOTE: 1. You must configure your distribution to cache based on the
            CloudFront-Is-*-Viewer headers. For more information, see
            the following documentation:
            https://docs.aws.amazon.com/console/cloudfront/cache-on-selected-headers
            https://docs.aws.amazon.com/console/cloudfront/cache-on-device-type
          2. CloudFront adds the CloudFront-Is-*-Viewer headers after the viewer
            request event. To use this example, you must create a trigger for the
            origin request event.
    '''

    desktopPath = '/desktop';
    mobilePath = '/mobile';
    tabletPath = '/tablet';
    smarttvPath = '/smarttv';

    if 'cloudfront-is-desktop-viewer' in headers and headers['cloudfront-is-desktop-
viewer'][0]['value'] == 'true':
        request['uri'] = desktopPath + request['uri']
    elif 'cloudfront-is-mobile-viewer' in headers and headers['cloudfront-is-mobile-
viewer'][0]['value'] == 'true':
        request['uri'] = mobilePath + request['uri']
    elif 'cloudfront-is-tablet-viewer' in headers and headers['cloudfront-is-tablet-
viewer'][0]['value'] == 'true':
        request['uri'] = tabletPath + request['uri']
    elif 'cloudfront-is-smarttv-viewer' in headers and headers['cloudfront-is-
smarttv-viewer'][0]['value'] == 'true':
        request['uri'] = smarttvPath + request['uri']

    print("Request uri set to %s" % request['uri'])

    return request
```

Content-based dynamische Ursprungsauswahl — Beispiele

Die folgenden Beispiele zeigen, wie Sie Lambda@Edge zur Weiterleitung an verschiedene Ursprünge basierend auf Informationen in der Anforderung verwenden können.

Themen

- Beispiel: Mithilfe eines Ursprungsanforderungsauslösers von einem benutzerdefinierten Ursprung zu einem Amazon-S3-Ursprung wechseln
- Beispiel: Ändern der Amazon-S3-Ursprungsregion mithilfe eines Ursprungsanforderungsauslösers
- Beispiel: Mithilfe eines Ursprungsanforderungsauslösers von einem Amazon-S3-Ursprung zu einem benutzerdefinierten Ursprung wechseln
- Beispiel: Verwenden eines Ursprungsanforderungsauslösers zur schrittweisen Übertragung von Netzwerkverkehr von einem Amazon-S3-Bucket zu einem anderen
- Beispiel: Verwenden eines Ursprungsanforderungsauslösers zum Ändern des Ursprungsdomainnamens anhand des Landes-Headers

Beispiel: Mithilfe eines Ursprungsanforderungsauslösers von einem benutzerdefinierten Ursprung zu einem Amazon-S3-Ursprung wechseln

Diese Funktion demonstriert, wie mit Hilfe eines Ursprungsanforderungsauslösers von einem benutzerdefinierten Ursprung zu einem Amazon S3-Ursprung gewechselt werden kann, von dem der Inhalt auf Basis der Anforderungseigenschaften abgerufen wird.

Node.js

```
'use strict';

 const querystring = require('querystring');

 exports.handler = (event, context, callback) => {
     const request = event.Records[0].cf.request;

     /**
      * Reads query string to check if S3 origin should be used, and
      * if true, sets S3 origin properties.
      */

     const params = querystring.parse(request.querystring);

     if (params['useS3Origin']) {
         if (params['useS3Origin'] === 'true') {
             const s3DomainName = 'amzn-s3-demo-bucket.s3.amazonaws.com';

             /* Set S3 origin fields */
             request.origin = {
                 s3: {
```

```
                    domainName: s3DomainName,
                    region: '',
                    authMethod: 'origin-access-identity',
                    path: '',
                    customHeaders: {}
                }
            };
            request.headers['host'] = [{ key: 'host', value: s3DomainName}];
        }
    }

    callback(null, request);
};
```

Python

```
from urllib.parse import parse_qs

def lambda_handler(event, context):
    request = event['Records'][0]['cf']['request']
    '''
    Reads query string to check if S3 origin should be used, and
    if true, sets S3 origin properties
    '''
    params = {k: v[0] for k, v in parse_qs(request['querystring']).items()}
    if params.get('useS3Origin') == 'true':
        s3DomainName = 'amzn-s3-demo-bucket.s3.amazonaws.com'

        # Set S3 origin fields
        request['origin'] = {
            's3': {
                'domainName': s3DomainName,
                'region': '',
                'authMethod': 'origin-access-identity',
                'path': '',
                'customHeaders': {}
            }
        }
        request['headers']['host'] = [{'key': 'host', 'value': s3DomainName}]
    return request
```

Beispiel: Ändern der Amazon-S3-Ursprungsregion mithilfe eines Ursprungsanforderungsauslösers

Diese Funktion demonstriert, wie mit Hilfe eines Ursprungsanforderungsauslösers von einem Amazon S3-Ursprung gewechselt werden kann, von dem der Inhalt auf Basis der Anforderungseigenschaften abgerufen wird.

In diesem Beispiel verwenden Sie den Wert des `CloudFront-Viewer-Country` Headers, um den S3-Bucket-Domänennamen auf einen Bucket in einer Region zu aktualisieren, die näher am Betrachter liegt. Dies kann auf verschiedene Weise nützlich sein:

- Es werden Latenzen reduziert, wenn die angegebene Region näher am Land des Viewers liegt.
- Es sorgt für Datenhoheit, indem es sicherstellt, dass die Daten aus einem Ursprung stammen, der in dem Land liegt, aus dem die Anfrage stammt.

In diesem Beispiel gehen Sie wie folgt vor:

- Konfigurieren Sie Ihre Verteilung so, dass die Zwischenspeicherung auf Grundlage des `CloudFront-Viewer-Country`-Headers erfolgt. Weitere Informationen finden Sie unter Basierend auf den ausgewählten Anforderungsheadern.
- Erstellen Sie einen Trigger für diese Funktion im ursprünglichen Anforderungsereignis. CloudFrontfügt den `CloudFront-Viewer-Country` Header nach dem Viewer-Anforderungsereignis hinzu. Um dieses Beispiel zu verwenden, müssen Sie also sicherstellen, dass die Funktion für eine ursprüngliche Anfrage ausgeführt wird.

> **Note**
>
> Der folgende Beispielcode verwendet dieselbe Ursprungszugriffsidentität (OAI) für alle S3-Buckets, die Sie für Ihren Ursprung verwenden. Weitere Informationen finden Sie unter Ursprungszugriffsidentitäten.

Node.js

```
'use strict';

exports.handler = (event, context, callback) => {
    const request = event.Records[0].cf.request;
```

```
    /**
     * This blueprint demonstrates how an origin-request trigger can be used to
     * change the origin from which the content is fetched, based on request
 properties.
     * In this example, we use the value of the CloudFront-Viewer-Country header
     * to update the S3 bucket domain name to a bucket in a Region that is closer to
     * the viewer.
     *
     * This can be useful in several ways:
     *      1) Reduces latencies when the Region specified is nearer to the viewer's
     *         country.
     *      2) Provides data sovereignty by making sure that data is served from an
     *         origin that's in the same country that the request came from.
     *
     * NOTE: 1. You must configure your distribution to cache based on the
     *          CloudFront-Viewer-Country header. For more information, see
     *          https://docs.aws.amazon.com/console/cloudfront/cache-on-selected-
headers
     *       2. CloudFront adds the CloudFront-Viewer-Country header after the
 viewer
     *          request event. To use this example, you must create a trigger for
 the
     *          origin request event.
     */

    const countryToRegion = {
        'DE': 'eu-central-1',
        'IE': 'eu-west-1',
        'GB': 'eu-west-2',
        'FR': 'eu-west-3',
        'JP': 'ap-northeast-1',
        'IN': 'ap-south-1'
    };

    if (request.headers['cloudfront-viewer-country']) {
        const countryCode = request.headers['cloudfront-viewer-country'][0].value;
        const region = countryToRegion[countryCode];

        /**
         * If the viewer's country is not in the list you specify, the request
         * goes to the default S3 bucket you've configured.
         */
        if (region) {
            /**
```

```
             * If you've set up OAI, the bucket policy in the destination bucket
             * should allow the OAI GetObject operation, as configured by default
             * for an S3 origin with OAI. Another requirement with OAI is to provide
             * the Region so it can be used for the SIGV4 signature. Otherwise, the
             * Region is not required.
             */
            request.origin.s3.region = region;
            const domainName = `amzn-s3-demo-bucket-in-${region}.s3.
${region}.amazonaws.com`;
            request.origin.s3.domainName = domainName;
            request.headers['host'] = [{ key: 'host', value: domainName }];
        }
    }

    callback(null, request);
};
```

Python

```
def lambda_handler(event, context):
    request = event['Records'][0]['cf']['request']

    '''
    This blueprint demonstrates how an origin-request trigger can be used to
    change the origin from which the content is fetched, based on request
 properties.
    In this example, we use the value of the CloudFront-Viewer-Country header
    to update the S3 bucket domain name to a bucket in a Region that is closer to
    the viewer.

    This can be useful in several ways:
        1) Reduces latencies when the Region specified is nearer to the viewer's
            country.
        2) Provides data sovereignty by making sure that data is served from an
            origin that's in the same country that the request came from.

    NOTE: 1. You must configure your distribution to cache based on the
            CloudFront-Viewer-Country header. For more information, see
            https://docs.aws.amazon.com/console/cloudfront/cache-on-selected-headers
          2. CloudFront adds the CloudFront-Viewer-Country header after the viewer
            request event. To use this example, you must create a trigger for the
            origin request event.
    '''
```

```
    countryToRegion = {
        'DE': 'eu-central-1',
        'IE': 'eu-west-1',
        'GB': 'eu-west-2',
        'FR': 'eu-west-3',
        'JP': 'ap-northeast-1',
        'IN': 'ap-south-1'
    }

    viewerCountry = request['headers'].get('cloudfront-viewer-country')
    if viewerCountry:
        countryCode = viewerCountry[0]['value']
        region = countryToRegion.get(countryCode)

        # If the viewer's country in not in the list you specify, the request
        # goes to the default S3 bucket you've configured
        if region:
            '''
            If you've set up OAI, the bucket policy in the destination bucket
            should allow the OAI GetObject operation, as configured by default
            for an S3 origin with OAI. Another requirement with OAI is to provide
            the Region so it can be used for the SIGV4 signature. Otherwise, the
            Region is not required.
            '''
            request['origin']['s3']['region'] = region
            domainName = 'amzn-s3-demo-bucket-in-{0}.s3.
{0}.amazonaws.com'.format(region)
            request['origin']['s3']['domainName'] = domainName
            request['headers']['host'] = [{'key': 'host', 'value': domainName}]

    return request
```

Beispiel: Mithilfe eines Ursprungsanforderungsauslösers von einem Amazon-S3-Ursprung zu einem benutzerdefinierten Ursprung wechseln

Diese Funktion demonstriert, wie mit Hilfe eines Ursprungsanforderungsauslösers von einem benutzerdefinierten Ursprung gewechselt werden kann, von dem der Inhalt auf Basis der Anforderungseigenschaften abgerufen wird.

Node.js

```
'use strict';

const querystring = require('querystring');

 exports.handler = (event, context, callback) => {
     const request = event.Records[0].cf.request;

     /**
      * Reads query string to check if custom origin should be used, and
      * if true, sets custom origin properties.
      */

     const params = querystring.parse(request.querystring);

     if (params['useCustomOrigin']) {
         if (params['useCustomOrigin'] === 'true') {

             /* Set custom origin fields*/
             request.origin = {
                 custom: {
                     domainName: 'www.example.com',
                     port: 443,
                     protocol: 'https',
                     path: '',
                     sslProtocols: ['TLSv1', 'TLSv1.1'],
                     readTimeout: 5,
                     keepaliveTimeout: 5,
                     customHeaders: {}
                 }
             };
             request.headers['host'] = [{ key: 'host', value: 'www.example.com'}];
         }
     }
    callback(null, request);
};
```

Python

```
from urllib.parse import parse_qs

def lambda_handler(event, context):
```

```
    request = event['Records'][0]['cf']['request']

    # Reads query string to check if custom origin should be used, and
    # if true, sets custom origin properties

    params = {k: v[0] for k, v in parse_qs(request['querystring']).items()}

    if params.get('useCustomOrigin') == 'true':
            # Set custom origin fields
            request['origin'] = {
                'custom': {
                    'domainName': 'www.example.com',
                    'port': 443,
                    'protocol': 'https',
                    'path': '',
                    'sslProtocols': ['TLSv1', 'TLSv1.1'],
                    'readTimeout': 5,
                    'keepaliveTimeout': 5,
                    'customHeaders': {}
                }
            }
            request['headers']['host'] = [{'key': 'host', 'value':
 'www.example.com'}]

    return request
```

Beispiel: Verwenden eines Ursprungsanforderungsauslösers zur schrittweisen Übertragung von Netzwerkverkehr von einem Amazon-S3-Bucket zu einem anderen

Diese Funktion demonstriert, wie Sie schrittweise und kontrolliert den Datenverkehr von einem Amazon-S3-Bucket zu einem anderen übertragen können.

Node.js

```
'use strict';

    function getRandomInt(min, max) {
        /* Random number is inclusive of min and max*/
        return Math.floor(Math.random() * (max - min + 1)) + min;
 }

exports.handler = (event, context, callback) => {
```

```
    const request = event.Records[0].cf.request;
    const BLUE_TRAFFIC_PERCENTAGE = 80;

    /**
      * This Lambda function demonstrates how to gradually transfer traffic from
      * one S3 bucket to another in a controlled way.
      * We define a variable BLUE_TRAFFIC_PERCENTAGE which can take values from
      * 1 to 100. If the generated randomNumber less than or equal to
 BLUE_TRAFFIC_PERCENTAGE, traffic
      * is re-directed to blue-bucket. If not, the default bucket that we've
 configured
      * is used.
      */

    const randomNumber = getRandomInt(1, 100);

if (randomNumber <= BLUE_TRAFFIC_PERCENTAGE) {
         const domainName = 'blue-bucket.s3.amazonaws.com';
         request.origin.s3.domainName = domainName;
         request.headers['host'] = [{ key: 'host', value: domainName}];
     }
    callback(null, request);
};
```

Python

```
import math
import random

def getRandomInt(min, max):
    # Random number is inclusive of min and max
    return math.floor(random.random() * (max - min + 1)) + min

def lambda_handler(event, context):
    request = event['Records'][0]['cf']['request']
    BLUE_TRAFFIC_PERCENTAGE = 80

    '''
    This Lambda function demonstrates how to gradually transfer traffic from
    one S3 bucket to another in a controlled way.
    We define a variable BLUE_TRAFFIC_PERCENTAGE which can take values from
    1 to 100. If the generated randomNumber less than or equal to
 BLUE_TRAFFIC_PERCENTAGE, traffic
```

```
    is re-directed to blue-bucket. If not, the default bucket that we've configured
    is used.
    '''

    randomNumber = getRandomInt(1, 100)

    if randomNumber <= BLUE_TRAFFIC_PERCENTAGE:
        domainName = 'blue-bucket.s3.amazonaws.com'
        request['origin']['s3']['domainName'] = domainName
        request['headers']['host'] = [{'key': 'host', 'value': domainName}]

    return request
```

Beispiel: Verwenden eines Ursprungsanforderungsauslösers zum Ändern des Ursprungsdomainnamens anhand des Landes-Headers

Diese Funktion demonstriert, wie Sie den Ursprungsdomänennamen, basierend auf dem `CloudFront-Viewer-Country`-Header, ändern können, sodass der Inhalt von einem Ursprungsland aus bereitgestellt wird, das näher am Land des Viewers liegt.

Die Implementierung dieser Funktionalität für Ihre Verteilung bietet u. a. ggf. folgende Vorteile:

- Es werden Latenzen reduziert, wenn die angegebene Region näher am Land des Viewers liegt.
- Es wird für Datenhoheit gesorgt, indem sichergestellt wird, dass die Daten aus einem Ursprung stammen, der in dem Land liegt, aus dem die Anfrage stammt

Beachten Sie, dass Sie Ihre Verteilung für die Zwischenspeicherung auf Basis des `CloudFront-Viewer-Country`-Headers konfigurieren müssen, um diese Funktionalität zu aktivieren. Weitere Informationen finden Sie unter the section called “Basierend auf den ausgewählten Anforderungsheadern”.

Node.js

```
'use strict';

exports.handler = (event, context, callback) => {
     const request = event.Records[0].cf.request;

  if (request.headers['cloudfront-viewer-country']) {
        const countryCode = request.headers['cloudfront-viewer-country'][0].value;
```

```
        if (countryCode === 'GB' || countryCode === 'DE' || countryCode === 'IE' )
 {
            const domainName = 'eu.example.com';
            request.origin.custom.domainName = domainName;
            request.headers['host'] = [{key: 'host', value: domainName}];
        }
     }

    callback(null, request);
};
```

Python

```
def lambda_handler(event, context):
    request = event['Records'][0]['cf']['request']

    viewerCountry = request['headers'].get('cloudfront-viewer-country')
    if viewerCountry:
        countryCode = viewerCountry[0]['value']
        if countryCode == 'GB' or countryCode == 'DE' or countryCode == 'IE':
            domainName = 'eu.example.com'
            request['origin']['custom']['domainName'] = domainName
            request['headers']['host'] = [{'key': 'host', 'value': domainName}]
    return request
```

Aktualisieren von Fehlerstatusangaben – Beispiele

Die folgenden Beispiele leiten Sie bei der Verwendung von Lambda@Edge zum Ändern des an Benutzer zurückgegebenen Fehlerstatus an.

Themen

- Beispiel: Mithilfe eines Ursprungsantwortauslösers den Fehlerstatuscode auf 200 aktualisieren
- Beispiel: Mithilfe eines Ursprungsantwortauslösers den Fehlerstatuscode auf 302 aktualisieren

Beispiel: Mithilfe eines Ursprungsantwortauslösers den Fehlerstatuscode auf 200 aktualisieren

Diese Funktion demonstriert, wie Sie den Antwortstatus auf 200 aktualisieren und statischen Body-Content für die Rückgabe an den Viewer generieren können:

- Die Funktion wird in einer Ursprungsantwort ausgelöst.

- Der Antwortstatus vom Ursprungs-Server ist ein Fehlerstatuscode (4xx oder 5xx).

Node.js

```
'use strict';

exports.handler = (event, context, callback) => {
    const response = event.Records[0].cf.response;

    /**
     * This function updates the response status to 200 and generates static
     * body content to return to the viewer in the following scenario:
     * 1. The function is triggered in an origin response
     * 2. The response status from the origin server is an error status code (4xx or
 5xx)
     */

    if (response.status >= 400 && response.status <= 599) {
        response.status = 200;
        response.statusDescription = 'OK';
        response.body = 'Body generation example';
    }

    callback(null, response);
};
```

Python

```
def lambda_handler(event, context):
    response = event['Records'][0]['cf']['response']

    '''
    This function updates the response status to 200 and generates static
    body content to return to the viewer in the following scenario:
    1. The function is triggered in an origin response
    2. The response status from the origin server is an error status code (4xx or
 5xx)
    '''

    if int(response['status']) >= 400 and int(response['status']) <= 599:
        response['status'] = 200
        response['statusDescription'] = 'OK'
```

```
        response['body'] = 'Body generation example'
    return response
```

Beispiel: Mithilfe eines Ursprungsantwortauslösers den Fehlerstatuscode auf 302 aktualisieren

Diese Funktion demonstriert, wie Sie den HTTP-Statuscode auf 302 aktualisieren können, um ihn auf einen anderen Pfad (Cache-Verhalten) umzuleiten, der einen anderen Ursprung hat. Beachten Sie Folgendes:

- Die Funktion wird in einer Ursprungsantwort ausgelöst.
- Der Antwortstatus vom Ursprungs-Server ist ein Fehlerstatuscode (4xx oder 5xx).

Node.js

```
'use strict';

exports.handler = (event, context, callback) => {
    const response = event.Records[0].cf.response;
    const request = event.Records[0].cf.request;

    /**
     * This function updates the HTTP status code in the response to 302, to
 redirect to another
     * path (cache behavior) that has a different origin configured. Note the
 following:
     * 1. The function is triggered in an origin response
     * 2. The response status from the origin server is an error status code (4xx or
 5xx)
     */

    if (response.status >= 400 && response.status <= 599) {
        const redirect_path = `/plan-b/path?${request.querystring}`;

        response.status = 302;
        response.statusDescription = 'Found';

        /* Drop the body, as it is not required for redirects */
        response.body = '';
        response.headers['location'] = [{ key: 'Location', value: redirect_path }];
    }
```

```
    callback(null, response);
};
```

Python

```
def lambda_handler(event, context):
    response = event['Records'][0]['cf']['response']
    request = event['Records'][0]['cf']['request']

    '''
    This function updates the HTTP status code in the response to 302, to redirect
 to another
    path (cache behavior) that has a different origin configured. Note the
 following:
    1. The function is triggered in an origin response
    2. The response status from the origin server is an error status code (4xx or
 5xx)
    '''

    if int(response['status']) >= 400 and int(response['status']) <= 599:
        redirect_path = '/plan-b/path?%s' % request['querystring']

        response['status'] = 302
        response['statusDescription'] = 'Found'

        # Drop the body as it is not required for redirects
        response['body'] = ''
        response['headers']['location'] = [{'key': 'Location', 'value':
 redirect_path}]

    return response
```

Zugriff auf den Anforderungstext – Beispiele

Die folgenden Beispiele veranschaulichen, wie Sie Lambda@Edge für die Arbeit mit POST-Anforderungen verwenden können.

> **Note**
>
> Um diese Beispiele zu verwenden, müssen Sie die Option Textkörper einschließen in der Lambda-Funktionszuordnung der Verteilung aktivieren. Sie ist standardmäßig nicht aktiviert.

- Um diese Einstellung in der CloudFront Konsole zu aktivieren, aktivieren Sie das Kontrollkästchen Body in the Lambda Function Association einbeziehen.
- Um diese Einstellung in der CloudFront API oder mit zu aktivieren CloudFormation, setzen Sie das `IncludeBody` Feld auf `true` in`LambdaFunctionAssociation`.

Themen

- Beispiel: Verwenden eines Anforderungsauslösers zum Lesen eines HTML-Formulars
- Beispiel: Verwenden eines Anforderungsauslösers zum Bearbeiten eines HTML-Formulars

Beispiel: Verwenden eines Anforderungsauslösers zum Lesen eines HTML-Formulars

Diese Funktion zeigt, wie Sie den Body einer POST-Anforderung verarbeiten können, die durch ein HTML-Formular (Webformular) erzeugt wird (z. B. ein "Kontaktformular"). Beispielsweise könnten Sie ein HTML-Formular wie das Folgende nutzen:

```
<html>
  <form action="https://example.com" method="post">
    Param 1: <input type="text" name="name1"><br>
    Param 2: <input type="text" name="name2"><br>
    input type="submit" value="Submit">
  </form>
</html>
```

Für die folgende Beispielfunktion muss die Funktion in einer CloudFront -Betrachteranforderung oder -Ursprungsanforderung ausgelöst werden.

Node.js

```
'use strict';

const querystring = require('querystring');

/**
 * This function demonstrates how you can read the body of a POST request
 * generated by an HTML form (web form). The function is triggered in a
 * CloudFront viewer request or origin request event type.
 */
```

```
exports.handler = (event, context, callback) => {
    const request = event.Records[0].cf.request;

    if (request.method === 'POST') {
        /* HTTP body is always passed as base64-encoded string. Decode it. */
        const body = Buffer.from(request.body.data, 'base64').toString();

        /* HTML forms send the data in query string format. Parse it. */
        const params = querystring.parse(body);

        /* For demonstration purposes, we only log the form fields here.
         * You can put your custom logic here. For example, you can store the
         * fields in a database, such as Amazon DynamoDB, and generate a response
         * right from your Lambda@Edge function.
         */
        for (let param in params) {
            console.log(`For "${param}" user submitted "${params[param]}".\n`);
        }
    }
    return callback(null, request);
};
```

Python

```
import base64
from urllib.parse import parse_qs

'''
Say there is a POST request body generated by an HTML such as:

<html>
<form action="https://example.com" method="post">
    Param 1: <input type="text" name="name1"><br>
    Param 2: <input type="text" name="name2"><br>
    input type="submit" value="Submit">
</form>
</html>

'''

'''
This function demonstrates how you can read the body of a POST request
```

```
generated by an HTML form (web form). The function is triggered in a
CloudFront viewer request or origin request event type.
'''

def lambda_handler(event, context):
    request = event['Records'][0]['cf']['request']

    if request['method'] == 'POST':
        # HTTP body is always passed as base64-encoded string. Decode it
        body = base64.b64decode(request['body']['data'])

        # HTML forms send the data in query string format. Parse it
        params = {k: v[0] for k, v in parse_qs(body).items()}

        '''
        For demonstration purposes, we only log the form fields here.
        You can put your custom logic here. For example, you can store the
        fields in a database, such as Amazon DynamoDB, and generate a response
        right from your Lambda@Edge function.
        '''
        for key, value in params.items():
            print("For %s use submitted %s" % (key, value))

    return request
```

Beispiel: Verwenden eines Anforderungsauslösers zum Bearbeiten eines HTML-Formulars

Diese Funktion zeigt, wie Sie den Body einer POST-Anforderung bearbeiten können, die durch ein HTML-Formular (Webformular) erzeugt wird. Die Funktion wird in einer CloudFront Viewer-Anfrage oder einer Origin-Anfrage ausgelöst.

Node.js

```
'use strict';

const querystring = require('querystring');

exports.handler = (event, context, callback) => {
    var request = event.Records[0].cf.request;
    if (request.method === 'POST') {
        /* Request body is being replaced. To do this, update the following
        /* three fields:
```

```
         *    1) body.action to 'replace'
         *    2) body.encoding to the encoding of the new data.
         *
         *       Set to one of the following values:
         *
         *           text - denotes that the generated body is in text format.
         *               Lambda@Edge will propagate this as is.
         *           base64 - denotes that the generated body is base64 encoded.
         *               Lambda@Edge will base64 decode the data before sending
         *               it to the origin.
         *    3) body.data to the new body.
         */
        request.body.action = 'replace';
        request.body.encoding = 'text';
        request.body.data = getUpdatedBody(request);
    }
    callback(null, request);
};

function getUpdatedBody(request) {
    /* HTTP body is always passed as base64-encoded string. Decode it. */
    const body = Buffer.from(request.body.data, 'base64').toString();

    /* HTML forms send data in query string format. Parse it. */
    const params = querystring.parse(body);

    /* For demonstration purposes, we're adding one more param.
     *
     * You can put your custom logic here. For example, you can truncate long
     * bodies from malicious requests.
     */
    params['new-param-name'] = 'new-param-value';
    return querystring.stringify(params);
}
```

Python

```
import base64
from urllib.parse import parse_qs, urlencode

def lambda_handler(event, context):
    request = event['Records'][0]['cf']['request']
    if request['method'] == 'POST':
```

```
        '''
        Request body is being replaced. To do this, update the following
        three fields:
            1) body.action to 'replace'
            2) body.encoding to the encoding of the new data.

            Set to one of the following values:

                text - denotes that the generated body is in text format.
                    Lambda@Edge will propagate this as is.
                base64 - denotes that the generated body is base64 encoded.
                    Lambda@Edge will base64 decode the data before sending
                    it to the origin.
            3) body.data to the new body.
        '''
        request['body']['action'] = 'replace'
        request['body']['encoding'] = 'text'
        request['body']['data'] = getUpdatedBody(request)
    return request

def getUpdatedBody(request):
    # HTTP body is always passed as base64-encoded string. Decode it
    body = base64.b64decode(request['body']['data'])

    # HTML forms send data in query string format. Parse it
    params = {k: v[0] for k, v in parse_qs(body).items()}

    # For demonstration purposes, we're adding one more param

    # You can put your custom logic here. For example, you can truncate long
    # bodies from malicious requests
    params['new-param-name'] = 'new-param-value'
    return urlencode(params)
```

Einschränkungen für Edge-Funktionen

In den folgenden Themen werden die Einschränkungen beschrieben, die für CloudFront Functions und Lambda @Edge gelten. Einige Einschränkungen gelten für alle Edge-Funktionen, während andere nur für CloudFront Functions oder Lambda @Edge gelten.

Jedes Thema enthält detaillierte Informationen zu den Einschränkungen und Einschränkungen, die Sie bei der Entwicklung und Bereitstellung von Edge-Funktionen mit CloudFront berücksichtigen sollten.

Wenn Sie diese Einschränkungen kennen, können Sie sicherstellen, dass Ihre Edge-Funktionen erwartungsgemäß funktionieren und den unterstützten Funktionen entsprechen.

Themen

- Einschränkungen für alle Edge-Funktionen
- Einschränkungen bei Funktionen CloudFront
- Einschränkungen für Lambda@Edge

Weitere Informationen zu Kontingenten (auch zuvor Limits genannt) finden Sie unter Kontingente für CloudFront Funktionen und Kontingente für Lambda@Edge.

Einschränkungen für alle Edge-Funktionen

Die folgenden Einschränkungen gelten für alle Edge-Funktionen, sowohl für CloudFront Functions als auch für Lambda @Edge.

Topics

- AWS-Konto Besitz
- CloudFront Funktionen mit Lambda @Edge kombinieren
- HTTP-Statuscodes
- HTTP-Header
- Abfragezeichenfolgen
- URI
- URI-, Abfragezeichenfolge- und Header-Kodierung
- Microsoft Smooth Streaming
- Tagging

AWS-Konto Besitz

Um eine Kantenfunktion einer CloudFront Verteilung zuzuordnen, müssen Funktion und Verteilung derselben gehören AWS-Konto.

CloudFront Funktionen mit Lambda @Edge kombinieren

Für jedes Cache-Verhalten gelten die folgenden Einschränkungen:

- Jeder Ereignistyp (Viewer-Anforderung, Ursprungsanforderung, Ursprungsantwort und Viewer-Antwort) kann nur eine Edge-Funktionszuordnung aufweisen.
- Sie können CloudFront Functions und Lambda @Edge nicht in Viewer-Ereignissen (Viewer-Anfrage und Viewer-Antwort) kombinieren.

Alle anderen Kombinationen von Edge-Funktionen sind erlaubt. In der folgenden Tabelle werden die erlaubten Kombinationen erläutert.

		CloudFront Funktionen	
		Viewer-Anforderung	Viewer-Antwort
Lambda@Edge	Viewer-Anforderung	Nicht zulässig	Nicht zulässig
	Ursprungsanfrage	Zulässig	Zulässig
	Ursprungsantwort	Zulässig	Zulässig
	Viewer-Antwort	Nicht zulässig	Nicht zulässig

HTTP-Statuscodes

CloudFront ruft keine Edge-Funktionen für Viewer-Response-Ereignisse auf, wenn der Ursprung den HTTP-Statuscode 400 oder höher zurückgibt.

Lambda@Edge-Funktionen für Ursprungsantwortereignisse werden für alle Ursprungsantworten aufgerufen, auch wenn der Ursprung einen HTTP-Statuscode 400 oder höher zurückgibt. Weitere Informationen finden Sie unter Aktualisieren von HTTP-Antworten in Ursprungsantwortauslösern.

HTTP-Header

Bestimmte HTTP-Header sind nicht zulässig, was bedeutet, dass sie nicht für Edge-Funktionen zugänglich sind und die Funktionen sie nicht hinzufügen können. Andere Header sind schreibgeschützt, was bedeutet, dass die Funktionen sie zwar lesen, aber nicht hinzufügen, ändern oder löschen können.

Topics

- Unzulässige Header
- Read-only Header

Unzulässige Header

Die folgenden HTTP-Header sind nicht für Edge-Funktionen verfügbar und die Funktionen können sie nicht hinzufügen. Wenn Ihre Funktion einen dieser Header hinzufügt, schlägt die CloudFront Validierung fehl und CloudFront gibt den HTTP-Statuscode 502 (Bad Gateway) an den Viewer zurück.

- `Connection`
- `Expect`
- `Keep-Alive`
- `Proxy-Authenticate`
- `Proxy-Authorization`
- `Proxy-Connection`
- `Trailer`
- `Upgrade`
- `X-Accel-Buffering`
- `X-Accel-Charset`
- `X-Accel-Limit-Rate`
- `X-Accel-Redirect`
- `X-Amz-Cf-*`
- `X-Amzn-Auth`
- `X-Amzn-Cf-Billing`
- `X-Amzn-Cf-Id`
- `X-Amzn-Cf-Xff`
- `X-Amzn-Errortype`
- `X-Amzn-Fle-Profile`
- `X-Amzn-Header-Count`

- `X-Amzn-Header-Order`
- `X-Amzn-Lambda-Integration-Tag`
- `X-Amzn-RequestId`
- `X-Cache`
- `X-Edge-*`
- `X-Forwarded-Proto`
- `X-Real-IP`
- `CloudFront-Viewer-Cert-PEM`
- `Client-Cert`
- `Client-Cert-Chain`

Read-only Header

Die folgenden Header sind schreibgeschützt. Ihre Funktion kann sie lesen und als Eingabe für die Funktionslogik verwenden, doch sie kann die Werte nicht ändern. Wenn Ihre Funktion einen schreibgeschützten Header hinzufügt oder bearbeitet, schlägt die CloudFront Überprüfung der Anforderung fehl und CloudFront gibt den HTTP-Statuscode 502 (Bad Gateway) an den Viewer zurück.

Read-only Header in Viewer-Anforderungsereignissen

Die folgenden Header sind bei Viewer-Anforderungsereignissen schreibgeschützt.

- `CDN-Loop`
- `Content-Length`
- `Host`
- `Transfer-Encoding`
- `Via`

Read-only Header in Ursprungsanforderungsereignissen (nur Lambda @Edge)

Die folgenden Header sind in Ursprungsanforderungsereignissen schreibgeschützt, die nur in Lambda@Edge vorhanden sind.

- `Accept-Encoding`

- `CDN-Loop`
- `Content-Length`
- `If-Modified-Since`
- `If-None-Match`
- `If-Range`
- `If-Unmodified-Since`
- `Transfer-Encoding`
- `Via`

Read-only Header in Ursprungs-Antwortereignissen (nur Lambda @Edge)

Die folgenden Header sind in Ursprungsantwortereignissen schreibgeschützt, die nur in Lambda@Edge vorhanden sind.

- `Transfer-Encoding`
- `Via`

Read-only Header in Zuschauer-Antwortereignissen

Die folgenden Header sind in Viewer-Antwortereignissen sowohl für CloudFront Functions als auch für Lambda @Edge schreibgeschützt.

- `Warning`
- `Via`

Die folgenden Header sind bei Viewer-Antwortereignissen für Lambda@Edge schreibgeschützt.

- `Content-Length`
- `Content-Encoding`
- `Transfer-Encoding`

Abfragezeichenfolgen

Die folgenden Einschränkungen gelten für Funktionen, die eine Abfragezeichenfolge in einem Anforderungs-URI lesen, aktualisieren oder erstellen.

- (Nur Lambda@Edge) Um auf die Abfragezeichenfolge in einer Ursprungsanforderung oder einer Ursprungsantwortfunktion zuzugreifen, muss die Cache-Richtlinie oder die Ursprungsanforderungsrichtlinie auf Alle auf Abfragezeichenfolgen gesetzt werden.
- Eine Funktion kann eine Abfragezeichenfolge für Viewer-Anforderungs- und Ursprungsanforderungsereignisse erstellen oder aktualisieren (Ursprungsanforderungsereignisse existieren nur in Lambda@Edge).
- Eine Funktion kann für Ursprungsantwort- und Viewer-Antwortereignisse eine Abfragezeichenfolge lesen, aber nicht erstellen oder aktualisieren (Ursprungsantwortereignisse existieren nur in Lambda@Edge).
- Wenn eine Funktion eine Abfragezeichenfolge erstellt oder aktualisiert, gelten folgende Einschränkungen:
  - Die Abfragezeichenfolge darf keine Leerzeichen, Steuerzeichen oder die Fragment-ID (#) enthalten.
  - Die Gesamtgröße der URI und der Abfragezeichenfolge darf nicht mehr als 8.192 Zeichen umfassen.
  - Wir empfehlen die Verwendung der Prozentkodierung für die URI und die Abfragezeichenfolge. Weitere Informationen finden Sie unter URI-, Abfragezeichenfolge- und Header-Kodierung.

URI

Wenn eine Funktion Änderungen an dem URI für eine Anforderung durchführt, ändert dies weder das Cache-Verhalten für die Anforderung noch den Ursprung, an den Anforderung weitergeleitet wird.

Die Gesamtgröße der URI und der Abfragezeichenfolge darf nicht mehr als 8.192 Zeichen umfassen.

URI-, Abfragezeichenfolge- und Header-Kodierung

Die Werte für den URI, die Abfragezeichenfolge und die Header, die an Edge-Funktionen übergeben werden, sind codiert. UTF-8 Ihre Funktion sollte die UTF-8 Kodierung für die URI-, Abfragezeichenfolge- und Header-Werte verwenden, die sie zurückgibt. Die prozentuale Kodierung ist mit der UTF-8 Kodierung kompatibel.

In der folgenden Liste wird erklärt, wie die Codierung für den URI, die Abfragezeichenfolge und die Header CloudFront behandelt wird:

- Wenn Werte in der Anfrage UTF-8 codiert sind, CloudFront leitet die Werte an Ihre Funktion weiter, ohne sie zu ändern.

- Wenn Werte in der Anforderung ISO-8859-1 codiert sind, CloudFront konvertiert die Werte in die UTF-8 Kodierung, bevor sie an Ihre Funktion weitergeleitet werden.
- Wenn Werte in der Anfrage mit einer anderen Zeichenkodierung codiert werden, CloudFront wird davon ausgegangen, dass sie ISO-8859-1 kodiert sind, und versucht, von in zu konvertieren. ISO-8859-1 UTF-8

> **Important**
>
> Die konvertierten Zeichen sind möglicherweise eine falsche Interpretation der Werte in der ursprünglichen Anfrage. Dies kann dazu führen, dass Ihre Funktion oder Ihr Ursprung ein unbeabsichtigtes Ergebnis produzieren.

Die Werte für den URI, die Abfragezeichenfolge und die Header, die an Ihren Ursprung CloudFront weitergeleitet werden, hängen davon ab, ob eine Funktion die Werte ändert:

- Wenn eine Funktion den URI, die Abfragezeichenfolge oder den Header nicht ändert, CloudFront leitet sie die Werte, die sie in der Anfrage erhalten hat, an Ihren Ursprung weiter.
- Wenn eine Funktion den URI, die Abfragezeichenfolge oder den Header ändert, CloudFront leitet sie die UTF-8 kodierten Werte weiter.

Microsoft Smooth Streaming

Sie können Edge-Funktionen nicht mit einer CloudFront Distribution verwenden, die Sie zum Streamen von Mediendateien verwenden, die Sie in das Microsoft Smooth Streaming-Format transkodiert haben.

Tagging

Sie können keine Tags zu Edge-Funktionen hinzufügen. Weitere Informationen zum Taggen finden Sie unter CloudFront. Kennzeichnen Sie eine Ressource CloudFront

Einschränkungen bei Funktionen CloudFront

Die folgenden Einschränkungen gelten nur für CloudFront Functions.

Inhalt

- Protokolle

- Anforderungstext
- Verwenden temporärer Anmeldeinformationen mit der CloudFront KeyValueStore API
- Laufzeit
- Computing-Auslastung

Weitere Informationen zu Kontingenten (zuvor Limits genannt) finden Sie unter Kontingente für CloudFront Funktionen.

Protokolle

Funktionsprotokolle in CloudFront Functions werden um 10 KB gekürzt.

Anforderungstext

CloudFront Funktionen können nicht auf den Hauptteil der HTTP-Anfrage zugreifen.

Verwenden temporärer Anmeldeinformationen mit der CloudFront KeyValueStore API

Sie können AWS -Security-Token-Service (AWS STS) verwenden, um temporäre Sicherheitsanmeldeinformationen (auch als Sitzungstoken bekannt) zu generieren. Mithilfe von Sitzungstoken können Sie vorübergehend eine AWS Identity and Access Management (IAM-) Rolle übernehmen, sodass Sie darauf zugreifen AWS-Services können.

Um die CloudFront KeyValueStore API aufzurufen, verwenden Sie einen regionalen Endpunkt, AWS STS um ein Sitzungstoken der Version 2 zurückzugeben. Wenn Sie den globalen Endpunkt für AWS STS (`sts.amazonaws.com`) verwenden, AWS STS wird ein Sitzungstoken der Version 1 generiert, das von Signature Version 4A (SigV4A) nicht unterstützt wird. Infolgedessen wird ein Authentifizierungsfehler angezeigt.

Um die CloudFront KeyValueStore API aufzurufen, können Sie die folgenden Optionen verwenden:

AWS CLI und AWS SDKs

Sie können das AWS CLI oder ein AWS SDK so konfigurieren, dass regionale AWS STS Endpunkte verwendet werden. Weitere Informationen finden Sie unter AWS STS -regionalisierte Endpunkte im Referenzhandbuch für AWS SDK und Tools.

Weitere Informationen zu verfügbaren AWS STS Endpunkten finden Sie unter Regionen und Endpunkte im IAM-Benutzerhandbuch.

SAML

Sie können SAML so konfigurieren, dass regionale Endpunkte verwendet werden. AWS STS Weitere Informationen finden Sie im Blogbeitrag How to use regional SAML endpoints for failover.

`SetSecurityTokenServicePreferences`-API

Anstatt einen regionalen AWS STS Endpunkt zu verwenden, können Sie den globalen Endpunkt so konfigurieren, AWS STS dass Sitzungstoken der Version 2 zurückgegeben werden. Verwenden Sie dazu den SetSecurityTokenServicePreferences API-Vorgang, um Ihre zu konfigurieren AWS-Konto.

Example Beispiel: IAM-CLI-Befehl

```
aws iam set-security-token-service-preferences --global-endpoint-token-version
 v2Token
```

> **Tip**
>
> Wir empfehlen, die AWS STS regionalen Endpunkte anstelle dieser Option zu verwenden. Regionale Endpunkte bieten eine höhere Verfügbarkeit und Failover-Szenarien.

Benutzerdefinierter Identitätsanbieter

Wenn Sie einen benutzerdefinierten Identitätsanbieter verwenden, der sich um den Verbund kümmert und die Rolle übernimmt, verwenden Sie eine der vorherigen Optionen für das übergeordnete Identitätsanbietersystem, das für die Generierung des Sitzungstokens verantwortlich ist.

Laufzeit

Die CloudFront Functions-Laufzeitumgebung unterstützt keine dynamische Codeauswertung und schränkt den Zugriff auf das Netzwerk, das Dateisystem, Umgebungsvariablen und Timer ein. Weitere Informationen finden Sie unter Eingeschränkte Funktionen.

> **Note**
>
> Um sie verwenden zu können CloudFront KeyValueStore, muss Ihre CloudFront Funktion JavaScript Runtime 2.0 verwenden.

Computing-Auslastung

CloudFront Funktionen haben ein Limit für die Ausführungsdauer, die als Rechenauslastung gemessen wird. Die Computing-Auslastung ist eine Zahl zwischen 0 und 100, mit der die Zeit angegeben wird, die die Ausführung der Funktion als Prozentsatz der maximal zulässigen Zeit in Anspruch genommen hat. Zum Beispiel bedeutet eine Rechenauslastung von 35, dass die Funktion in 35 % der maximal zulässigen Zeit abgeschlossen wurde.

Wenn Sie eine Funktion testen können Sie den Wert der Computing-Auslastung in der Ausgabe des Testereignisses sehen. Für Produktionsfunktionen können Sie die Metrik zur Rechenauslastung auf der Seite „Überwachung“ in der CloudFront Konsole oder unter einsehen CloudWatch.

Einschränkungen für Lambda@Edge

Die folgenden Einschränkungen gelten nur für Lambda@Edge.

Inhalt

- DNS-Auflösung
- HTTP-Statuscodes
- Version der Lambda-Funktion
- Lambda-Region
- Lambda-Rollenberechtigungen
- Lambda-Funktionen
- Unterstützte Laufzeiten
- CloudFront Header
- Einschränkungen für Anforderungstext mit der Option „Text einschließen“
- Antwort-Timeout und Keep-Alive-Timeout (nur benutzerdefinierte Ursprünge)

Hinweise zu Kontingenten finden Sie unter Kontingente für Lambda@Edge.

DNS-Auflösung

CloudFront führt eine DNS-Auflösung für den Namen der Ursprungsdomäne durch, bevor die Lambda @Edge -Funktion Ihrer ursprünglichen Anfrage ausgeführt wird. Wenn beim DNS-Dienst für Ihre Domain Probleme auftreten und der Domainname nicht aufgelöst werden CloudFront kann, um die IP-Adresse abzurufen, wird Ihre Lambda @Edge -Funktion nicht aufgerufen. CloudFrontgibt einen HTTP 502-Statuscode (Bad Gateway) an den Client zurück. Weitere Informationen finden Sie unter DNS-Fehler (`NonS3OriginDnsError`).

Wenn Ihre Funktionslogik den ursprünglichen Domainnamen ändert, CloudFront wird nach Abschluss der Ausführung der Funktion eine weitere DNS-Auflösung für den aktualisierten Domainnamen durchgeführt.

Weitere Informationen zur Verwaltung des DNS-Failovers finden Sie unter Konfigurieren von DNS-Failover im Entwicklerhandbuch für Amazon Route 53.

HTTP-Statuscodes

Lambda @Edge -Funktionen für Viewer-Antwortereignisse können den HTTP-Statuscode der Antwort nicht ändern, unabhängig davon, ob die Antwort aus dem Ursprung oder dem CloudFront Cache stammt.

Version der Lambda-Funktion

Sie müssen eine nummerierte Version der Lambda-Funktion verwenden, nicht jedoch `$LATEST` oder Aliase.

Lambda-Region

Die Lambda-Funktion muss sich in der Region USA Ost (Nord-Virginia) befinden.

Lambda-Rollenberechtigungen

Um Auslöser hinzufügen zu können, muss die Ihrer Lambda-Funktion zugewiesene IAM-Ausführungsrolle von den Haupt-Services `lambda.amazonaws.com` und `edgelambda.amazonaws.com` eingenommen werden können. Weitere Informationen finden Sie unter Einrichten von IAM-Berechtigungen und -Rollen für Lambda@Edge.

Lambda-Funktionen

Die folgenden Lambda-Funktionen werden von Lambda@Edge nicht unterstützt:

- Andere Lambda-Laufzeit-Verwaltungskonfigurationen als Automatisch (Standard)
- Konfiguration Ihrer Lambda-Funktion für den Zugriff auf Ressourcen in Ihrer VPC
- Warteschlangen für unzustellbare Briefe der Lambda-Funktion
- Lambda-Umgebungsvariablen (mit Ausnahme von reservierten Umgebungsvariablen, die automatisch unterstützt werden)
- Lambda-Funktionen mit dem Verwalten von AWS Lambda -Abhängigkeiten mit Ebenen
- Verwenden von AWS X-Ray
- Parallelität per Lambda

> **Note**
>
> Lambda@Edge-Funktionen nutzen dieselben regionalen Parallelitätsfunktionen wie alle Lambda-Funktionen. Weitere Informationen finden Sie unter Kontingente für Lambda@Edge.

- Erstellen Sie eine Lambda-Funktion mithilfe eines Container-Images
- Lambda-Funktionen, die die arm64-Architektur verwenden
- Lambda-Funktionen mit mehr als 512 MB flüchtigem Speicher
- Verwenden eines kundenseitig verwalteten Schlüssels zur Verschlüsselung Ihrer ZIP-Bereitstellungspakete
- Self-managed Amazon S3-Codespeicher

Unterstützte Laufzeiten

Lambda @Edge unterstützt die neuesten Versionen von Node.js und Python-Laufzeiten. Eine Liste der unterstützten Versionen und ihrer zukünftiger Verfallsdaten finden Sie unter Unterstützte Laufzeiten im Entwicklerhandbuch für AWS Lambda .

> **Tip**
>
> - Es hat sich bewährt, die neuesten Versionen der bereitgestellten Laufzeiten für Leistungsverbesserungen und neue Funktionen zu verwenden.
> - Sie können keine Funktionen mit veralteten Versionen von erstellen oder aktualisieren. Node.js Sie können vorhandene Funktionen mit diesen Versionen nur Distributionen

verknüpfen. CloudFront Funktionen dieser Versionen, die Distributionen zugeordnet sind, werden weiterhin ausgeführt. Wir empfehlen jedoch, dass Sie Ihre Funktion auf neuere Versionen von Node.js verschieben. Weitere Informationen finden Sie unter Runtime Deprecation Policy im AWS Lambda Developer Guide und im Node.js Release-Zeitplan unter. GitHub

CloudFront Header

Lambda @Edge -Funktionen können jeden der unter aufgeführten CloudFront Header lesen, bearbeiten, entfernen oder hinzufügen. CloudFront Anforderungsheader hinzufügen

Hinweise

- Wenn Sie diese Header hinzufügen CloudFront möchten, müssen Sie sie so konfigurieren CloudFront , dass sie mithilfe einer Cache-Richtlinie oder einer Origin-Request-Richtlinie hinzugefügt werden.
- CloudFront fügt die Header nach dem Viewer-Anforderungsereignis hinzu, was bedeutet, dass die Header für Lambda @Edge -Funktionen in einer Viewer-Anfrage nicht verfügbar sind. Die Header sind nur für Lambda@Edge-Funktionen in einer Ursprungsanfrage und einer Ursprungsantwort verfügbar.
- Wenn die Viewer-Anfrage Header mit diesen Namen enthält und Sie diese Header mithilfe einer Cache-Richtlinie oder einer Origin-Request-Richtlinie so konfiguriert CloudFront haben, dass diese Header hinzugefügt werden, werden die Header-Werte, die in der Viewer-Anfrage enthalten waren, CloudFront überschrieben. Viewer-facing Funktionen sehen den Header-Wert aus der Viewer-Anfrage, während Funktionen, die auf den Ursprung ausgerichtet sind, den hinzugefügten Header-Wert sehen. CloudFront
- Wenn eine Viewer-Anforderungsfunktion den `CloudFront-Viewer-Country` Header hinzufügt, schlägt die Überprüfung fehl und CloudFront gibt den HTTP-Statuscode 502 (Bad Gateway) an den Viewer zurück.

Einschränkungen für Anforderungstext mit der Option „Text einschließen“

Wenn Sie die Option Text einschließen auswählen, um den Anforderungstext an die Lambda@Edge-Funktion weiterzugeben, beachten Sie die folgenden Informationen und Größenbeschränkungen für die Teile des Textes, die offengelegt oder ersetzt werden.

- CloudFront Base64 kodiert den Anforderungstext immer, bevor er Lambda @Edge zur Verfügung gestellt wird.
- Wenn der Anforderungstext groß ist, wird er wie folgt CloudFront gekürzt, bevor er Lambda @Edge zur Verfügung gestellt wird:
  - Bei Viewer-Anforderungsereignissen wird der Fließtext bei 40 KB abgeschnitten.
  - Bei Ursprungsanforderungsereignissen wird der Fließtext bei 1 MB abgeschnitten.
- Wenn Sie schreibgeschützt auf den Anforderungstext zugreifen, wird der gesamte ursprüngliche CloudFront Anforderungstext an den Ursprung gesendet.
- Wenn Ihre Lambda@Edge-Funktion den Anforderungstext ersetzt, gelten die folgenden Größenbeschränkungen für den Text, den die Funktion zurückgibt:
  - Wenn die Lambda@Edge-Funktion den Textkörper als Klartext zurückgibt:
    - Bei Viewer-Anforderungsereignissen ist der Text auf 40 KB beschränkt.
    - Bei Ursprungsanforderungsereignissen ist der Text auf 1 MB beschränkt.
  - Wenn die Lambda@Edge Funktion den Textkörper als base64-codierten Text zurückgibt:
    - Bei Viewer-Anforderungsereignissen ist der Text auf 53,2 KB beschränkt.
    - Bei Ursprungsanforderungsereignissen ist der Text auf 1,33 MB beschränkt.

> **Note**
>
> Wenn Ihre Lambda@Edge-Funktion einen Text zurückgibt, der diese Grenzwerte überschreitet, schlägt Ihre Anforderung mit dem HTTP-Statuscode 502 (Lambda-Validierungsfehler) fehl. Wir empfehlen Ihnen, Ihre Lambda@Edge-Funktion so zu aktualisieren, dass der Text diese Grenzwerte nicht überschreitet.

Antwort-Timeout und Keep-Alive-Timeout (nur benutzerdefinierte Ursprünge)

Wenn Sie Lambda@Edge-Funktionen verwenden, um das Antwort-Timeout oder das Keep-Alive-Timeout für Ihre Distributionsursprünge festzulegen, stellen Sie sicher, dass Sie einen Wert angeben, den Ihr Ursprung unterstützen kann. Weitere Informationen finden Sie unter Kontingent für Antwort- und Keep-Alive-Timeout.

Berichte, Metriken und Protokolle

CloudFront bietet verschiedene Optionen für die Berichterstattung, Überwachung und Protokollierung Ihrer CloudFront Ressourcen. Sie können folgende Aufgaben erledigen:

- Sie können Berichte anzeigen und herunterladen, um die Nutzung und Aktivität Ihrer CloudFront Distributionen zu sehen, darunter Abrechnungsberichte, Cache-Statistiken, beliebte Inhalte und Top-Referrer.
- Überwachen und verfolgen Sie CloudFront, einschließlich Ihrer Edge-Computing-Funktionen, direkt in der CloudFront Konsole oder mithilfe von Amazon CloudWatch. CloudFront sendet Metriken an CloudWatch für Verteilungen und Edge-Funktionen, sowohl an Lambda @Edge als auch an Functions. CloudFront
- Zeigen Sie Protokolle für die Viewer-Anfragen an, die Ihre CloudFront Distributionen erhalten, mit Standardprotokollen oder Echtzeit-Zugriffsprotokollen. Zusätzlich zu Viewer-Anforderungsprotokollen können Sie CloudWatch Logs verwenden, um Protokolle für Ihre Edge-Funktionen, sowohl Lambda @Edge als auch CloudFront Functions, abzurufen. Sie können es auch verwenden AWS CloudTrail , um Protokolle der CloudFront API-Aktivitäten in Ihrem AWS-Konto abzurufen.
- Verfolgen Sie Konfigurationsänderungen an Ihren CloudFront Ressourcen mithilfe von AWS Config.

Weitere Informationen zu diesen Optionen enthalten die folgenden Abschnitte.

Themen

- AWS Abrechnungs- und Nutzungsberichte für CloudFront
- CloudFront Berichte in der Konsole anzeigen
- Überwachen Sie CloudFront Metriken mit Amazon CloudWatch
- CloudFront und Edge-Funktionsprotokollierung
- Verfolgen Sie Konfigurationsänderungen mit AWS Config

AWS Abrechnungs- und Nutzungsberichte für CloudFront

AWS bietet zwei Nutzungsberichte für CloudFront:

- Der AWS Abrechnungsbericht bietet eine allgemeine Übersicht über alle Aktivitäten AWS-Services , die Sie verwenden, einschließlich CloudFront.
- Der AWS Nutzungsbericht ist eine Zusammenfassung der Aktivitäten für einen bestimmten Dienst, aggregiert nach Stunden, Tagen oder Monaten. Er enthält auch Nutzungsdiagramme, die eine grafische Darstellung Ihrer CloudFront Nutzung bieten.

> **Note**
>
> Wie bei anderen AWS-Services wird Ihnen nur das CloudFront berechnet, was Sie verwenden. Weitere Informationen finden Sie unter CloudFront Preise.

Themen

- Sehen Sie sich das an AWS Abrechnungsbericht für CloudFront
- Sehen Sie sich das an AWS Nutzungsbericht für CloudFront
- Interpretieren Sie Ihre AWS Rechnungs- und Nutzungsberichte für CloudFront

Sehen Sie sich das an AWS Abrechnungsbericht für CloudFront

Eine Zusammenfassung Ihrer AWS Nutzung und Ihrer Gebühren, sortiert nach Diensten, finden Sie auf der Seite Rechnungen in der AWS Fakturierung und Kostenmanagement Konsole.

Um das einzusehen AWS Abrechnungsbericht

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die AWS Fakturierung und Kostenmanagement Konsole unter https://console.aws.amazon.com/costmanagement/.
2. Wählen Sie im Navigationsbereich Rechnungen aus.
3. Wählen Sie einen Abrechnungszeitraum (zum Beispiel August 2023).
4. Wählen Sie auf der Registerkarte Gebühren nach Dienst die Option Global oder den AWS-Region Namen aus CloudFront, und erweitern Sie sie dann.
5. Um einen detaillierten Abrechnungsbericht im CSV-Format herunterzuladen, wählen Sie Alles als CSV herunterladen aus.

Weitere Informationen zu Ihrer AWS Rechnung finden Sie im AWS Billing Benutzerhandbuch unter Ihre Rechnung anzeigen.

Der Abrechnungsbericht enthält die folgenden Werte, die gelten für CloudFront:

- ProductCode – `AmazonCloudFront`
- UsageType— Einer der folgenden Werte:
  - Ein Code zur Identifizierung des Typs der Datenübertragung
  - `Invalidations`
  - `Executions-CloudFrontFunctions`
  - `KeyValueStore-APIOperations`
  - `KeyValueStore-EdgeReads`
  - `RealTimeLog-KinesisDataStream`
  - `SSL-Cert-Custom`
- ItemDescription— Eine Beschreibung des Abrechnungssatzes für den UsageType.
- UsageStart Datum und UsageEndDate — Der Tag, für den die Nutzung gilt, in koordinierter Weltzeit (UTC).
- UsageQuantity— Einer der folgenden Werte:
  - Die Anzahl der Anforderungen in dem angegebenen Zeitraum
  - Die übertragene Datenmenge in Gigabyte
  - Die Anzahl der invalidierten Objekte
  - Die Summe der anteiligen Monate, in denen Sie über SSL-Zertifikate verfügten, die mit aktivierten CloudFront Distributionen verknüpft waren. Wenn Sie beispielsweise ein Zertifikat für eine aktivierte Verteilung für einen kompletten Monat und ein anderes Zertifikat für eine aktivierte Verteilung für die Hälfte des Monats aktiviert ist, beträgt dieser Wert 1,5.

Sehen Sie sich das an AWS Nutzungsbericht für CloudFront

AWS stellt einen CloudFront Nutzungsbericht bereit, der detaillierter ist als der Abrechnungsbericht, aber weniger detailliert als CloudFront Zugriffsprotokolle. Der Nutzungsbericht bietet gesammelte Nutzungsdaten nach Stunde, Tag und Monat und listet Operationen nach Region und Verwendungstyp auf, z. B. die Übertragung von Daten aus der Australien-Region.

Um den einzusehen AWS Nutzungsbericht

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die AWS Fakturierung und Kostenmanagement Konsole unter https://console.aws.amazon.com/costmanagement/.

2. Wählen Sie im Navigationsbereich Cost Explorer aus.
3. Wählen Sie auf der Seite Neuer Kosten- und Nutzungsbericht im Bereich Berichtsparameter einen Datumsbereich und eine Granularität für den Bericht aus.
4. Wählen Sie unter Filter die Option Service aus CloudFront.
5. Wählen Sie den Verwendungstyp aus.
6. Wählen Sie unter Aufschlüsselung der Kosten und Nutzung die Option Als CSV herunterladen aus.

Weitere Informationen zum AWS Nutzungsbericht finden Sie im AWS Data Exports Benutzerhandbuch unter AWS Nutzungsbericht.

Der CloudFront Nutzungsbericht enthält die folgenden Werte:

- Service – `AmazonCloudFront`
- Operation – HTTP-Methode. Werte sind u. a. angegeben für `DELETE`, `GET`, `HEAD`, `OPTIONS`, `PATCH`, `POST` und `PUT`.
- UsageType— Einer der folgenden Werte:
  - Ein Code zur Identifizierung des Typs der Datenübertragung
  - `Invalidations`
  - `Executions-CloudFrontFunctions`
  - `KeyValueStore-APIOperations`
  - `KeyValueStore-EdgeReads`
  - `RealTimeLog-KinesisDataStream`
  - `SSL-Cert-Custom`
- Ressource — Entweder die ID der CloudFront Distribution, die der Nutzung zugeordnet ist, oder die Zertifikat-ID eines SSL-Zertifikats, das Sie einer CloudFront Distribution zugeordnet haben.
- StartTime/EndTime— Der Tag, für den die Verwendung gilt, in koordinierter Weltzeit (UTC).
- UsageValue— 1) Die Anzahl der Anfragen während des angegebenen Zeitraums oder 2) die übertragene Datenmenge in Byte.

Wenn Sie Amazon S3 als Quelle für verwenden CloudFront, sollten Sie erwägen, auch den Nutzungsbericht für Amazon S3 zu erstellen. Wenn Sie Amazon S3 jedoch für andere Zwecke als als

Ursprung für Ihre CloudFront Distributionen verwenden, ist möglicherweise nicht klar, welcher Teil auf Ihre CloudFront Nutzung zutrifft.

> **Tip**
>
> Für detaillierte Informationen zu allen Anfragen, die Sie CloudFront für Ihre Objekte erhalten, aktivieren Sie die CloudFront Zugriffsprotokolle für Ihre Distribution. Weitere Informationen finden Sie unter the section called “Zugriffsprotokolle (Standardprotokolle)”.

Weitere Informationen zum Verständnis der CloudFront Gebühren und Nutzungsarten in Ihren Berichten finden Sie unterthe section called “Interpretieren Sie Ihre AWS Rechnungs- und Nutzungsberichte für CloudFront”.

Interpretieren Sie Ihre AWS Rechnungs- und Nutzungsberichte für CloudFront

Nachdem Sie den Abrechnungsbericht und den Nutzungsbericht erstellt haben, können Sie dieses Thema verwenden, um zu verstehen, wie die einzelnen CloudFront Gebühren, die auf Ihrer Rechnung erscheinen, und die entsprechende Nutzungsart für jede Gebühr zu interpretieren sind. Dieses Thema enthält die Codes und AWS-Region Abkürzungen, die in beiden Berichten vorkommen können.

Die meisten Codes in den beiden Spalten haben eine Abkürzung aus zwei Buchstaben, die den Ausführungsort der Aktivität angibt. In der folgenden Tabelle wird *region* in Ihrer AWS Rechnung und im Nutzungsbericht ein Code durch eine der folgenden aus zwei Buchstaben bestehenden Abkürzungen ersetzt:

- AP: Hongkong, Philippinen, Südkorea, Taiwan, und Singapur (Asien-Pazifik)
- AU: Australien
- CA: Kanada
- EU: Europa und Israel
- IN: Indien
- JP: Japan
- ME: Naher Osten
- SA: Südamerika

- US: Vereinigte Staaten
- ZA: Südafrika

Weitere Informationen zur Preisgestaltung nach AWS-Region finden Sie unter CloudFront Amazon-Preise.

Hinweise

- In dieser Tabelle sind keine Gebühren für die Übertragung Ihrer Objekte von einem Amazon S3-Bucket zu CloudFront Edge-Standorten enthalten. Diese Gebühren, sofern vorhanden, finden sich im Bereich AWS -Datenübertragung Ihrer AWS -Rechnung.
- In der ersten Spalte sind die Gebühren aufgeführt, die in Ihrem AWS Rechnungsbericht erscheinen, und es wird erklärt, was die einzelnen Gebühren bedeuten.
- In der zweiten Spalte sind die Elemente aufgeführt, die im AWS Nutzungsbericht erscheinen, und es wird die Korrelation zwischen den Rechnungsgebühren und den Elementen des Nutzungsberichts angezeigt.

CloudFront Gebühren in Ihrer AWS Rechnung	Werte in der UsageType Spalte im AWS Nutzungsbericht
region-DataTransfer-Out-Bytes Gesamtzahl der Byte, die von CloudFront *region* Edge-Standorten als Antwort auf Benutzer `GET` und `HEAD` Anfragen bereitgestellt wurden.	*region*-Out-Bytes-HTTP-Static: Byte, die mithilfe von HTTP für Objekte mit TTL ≥ 3.600 Sekunden bereitgestellt wurden. *region*-Out-Bytes-HTTPS-Static: Bytes, die mithilfe von HTTPS für Objekte mit TTL ≥ 3.600 Sekunden bereitgestellt werden. *region*-Out-Bytes-HTTP-Dynamic: Bytes, die mithilfe von HTTP für Objekte mit TTL < 3.600 Sekunden bereitgestellt werden. *region*-Out-Bytes-HTTPS-Dynamic:

CloudFront Gebühren in Ihrer AWS Rechnung	Werte in der UsageType Spalte im AWS Nutzungsbericht
	Bytes, die mithilfe von HTTPS für Objekte mit TTL < 3.600 Sekunden bereitgestellt werden. *region*-Out-Bytes-HTTP-Proxy: Bytes CloudFront , die mithilfe von HTTP als Antwort auf`DELETE`,`OPTIONS`, `PATCH` und `PUT` Anfragen von den Viewern zurückgegeben wurden. `POST` *region*-Out-Bytes-HTTPS-Proxy: Bytes CloudFront , die mithilfe von HTTPS als Antwort auf`DELETE`, `OPTIONSPATCH`, `POST` und `PUT` Anfragen von Viewern an Zuschauer zurückgegeben wurden. Dies schließt Bytes ein, die mithilfe von gRPC von CloudFront den Viewern zurückgegeben wurden.

CloudFront Gebühren in Ihrer AWS Rechnung	Werte in der UsageType Spalte im AWS Nutzungsbericht
region-DataTransfer-Out-OBytes Gesamtzahl der Byte, die als Antwort auf,, und PUT Anfragen von CloudFront Edge-Standorten zu DELETE Ihrem Ursprung oder Ihrer Edge-Funktion übertragen wurden. OPTIONS PATCH POST Die Gebühren beinhalten die Datenübertragung für WebSocket Daten vom Client zum Server.	*region*-Out-OBytes-HTTP-Proxy Gesamtzahl der Byte, die als Antwort auf,, und PUT Anfragen mithilfe von HTTP von CloudFront Edge-Standorten zu DELETE Ihrem Ursprung oder Ihrer Edge-Funktion übertragen wurden. OPTIONS PATCH POST *region*-Out-OBytes-HTTPS-Proxy Gesamtzahl der Bytes, die mithilfe von HTTPS von CloudFront Edge-Standorten zu Ihrem Ursprung oder Ihrer Edge-Funktion als Antwort auf DELETEOPTIONS, PATCHPOST, und PUT Anfragen übertragen wurden. Dies schließt Bytes ein, die mithilfe von gRPC von CloudFront Edge-Standorten zu Ihrem Ursprung oder Ihren CloudFront Funktionen übertragen wurden.
region-Requests-Tier1 Anzahl der HTTP-GET- und -HEAD-Anforderungen	*region*-Requests-HTTP-Static Anzahl der HTTP-GET- und -HEAD-Anforderungen für Objekte mit TTL ≥ 3 600 Sekunden *region*-Requests-HTTP-Dynamic Anzahl der HTTP-GET- und -HEAD-Anforderungen für Objekte mit TTL < 3 600 Sekunden

CloudFront Gebühren in Ihrer AWS Rechnung	Werte in der UsageType Spalte im AWS Nutzungsbericht
region-Requests-Tier2-HTTPS Anzahl der HTTPS-GET- und -HEAD-Anforderungen	*region*-Requests-HTTPS-Static Anzahl der HTTPS-GET- und -HEAD-Anforderungen für Objekte mit TTL ≥ 3 600 Sekunden *region*-Requests-HTTPS-Dynamic Anzahl der HTTPS-GET- und -HEAD-Anforderungen für Objekte mit TTL < 3 600 Sekunden
region-Requests-HTTP-Proxy Anzahl der HTTP-DELETE,OPTIONS,PATCH, und PUT AnfragenPOST, die an Ihre Ursprungs- oder Edge-Funktion CloudFront weitergeleitet werden. Enthält auch die Anzahl der WebSocket HTTP-Anfragen (GETAnfragen mit dem `Upgrade: websocket` Header), die an Ihre Ursprungs- oder Edge-Funktion CloudFront weitergeleitet werden.	*region*-Requests-HTTP-Proxy Entspricht dem entsprechenden Artikel auf Ihrer CloudFront Rechnung.

CloudFront Gebühren in Ihrer AWS Rechnung	Werte in der UsageType Spalte im AWS Nutzungsbericht
region-Requests-HTTPS-Proxy Anzahl der HTTPS-`DELETE`, `OPTIONSPATCH`, und `PUT` Anfragen`POST`, die an Ihre Origin- oder Edge-Funktion CloudFront weitergeleitet werden. Dies umfasst auch folgende Anforderu ngstypen: • Die Anzahl der WebSocket HTTPS-Anf ragen (`GET`Anfragen mit dem `Upgrade: websocket` Header), die an Ihre Ursprungs - oder Edge-Funktion CloudFront weitergel eitet werden. • Die Anzahl der HTTPS-gRPC-Anfragen.	*region*-Requests-HTTPS-Proxy Entspricht dem entsprechenden Artikel in Ihrer CloudFront Rechnung.
region-Requests-HTTPS-Proxy-FLE Anzahl der HTTPS-`DELETE`,, und `POST` Anfragen `OPTIONSPATCH`, die mit einer Verschlüsselung auf Feldebene verarbeitet wurden, die an Ihre Ursprungs- oder Edge-Funktion CloudFront weitergeleitet wird.	*region*-Requests-HTTPS-Proxy-FLE Entspricht dem entsprechenden Artikel auf Ihrer Rechnung. CloudFront
region-Bytes-OriginShield Gesamtzahl der Bytes, die vom Ursprung auf einen beliebigen regionalen Edge-Cach e übertragen wurden, einschließlich des regionalen Edge-Caches, der als Origin Shield aktiviert ist.	*region*-Bytes-OriginShield Entspricht dem entsprechenden Artikel auf Ihrer CloudFront Rechnung.

CloudFront Gebühren in Ihrer AWS Rechnung	Werte in der UsageType Spalte im AWS Nutzungsbericht
region-OBytes-OriginShield Gesamtzahl der Bytes, die zum Ursprung von einem beliebigen regionalen Edge-Cache übertragen wurden, einschließlich des regionalen Edge-Caches, der als Origin Shield aktiviert ist.	*region*-OBytes-OriginShield Entspricht dem entsprechenden Artikel auf Ihrer CloudFront Rechnung.
region-Requests-OriginShield Anzahl der Anfragen, die zu Origin Shield als inkrementelle Ebene gehen. Bei dynamischen (nicht zwischenspeicherbaren) Anforderungen, die an den Ursprung weitergeleitet werden, ist Origin Shield immer eine inkrementelle Ebene. Bei zwischenspeicherbaren Anfragen ist Origin Shield manchmal eine inkrementelle Ebene. Weitere Informationen finden Sie unter the section called “Schätzung der Kosten für Origin Shield”.	*region*-Requests-OriginShield Entspricht dem entsprechenden Artikel auf Ihrer CloudFront Rechnung.
Invalidations Die Gebühr für die Ungültigerklärung von Objekten (das Entfernen der Objekte an den CloudFront Kanten). Weitere Informationen finden Sie unter Kosten der Aufhebung der Gültigkeit von Dateien.	Invalidations Entspricht dem entsprechenden Artikel in Ihrer CloudFront Rechnung.

CloudFront Gebühren in Ihrer AWS Rechnung	Werte in der UsageType Spalte im AWS Nutzungsbericht
SSL-Cert-Custom Die Gebühr für die Verwendung eines SSL-Zertifikats mit einem CloudFront alternativen Domainnamen wie example.com anstelle des CloudFront Standard-SSL-Zertifikats und des Domainnamens, CloudFront der Ihrer Distribut ion zugewiesen wurde.	SSL-Cert-Custom Entspricht dem entsprechenden Artikel auf Ihrer CloudFront Rechnung.
RealTimeLog-KinesisDataStream Die Gebühr für die Anzahl der Zeilen, die für Zugriffsprotokolle in Echtzeit generiert werden.	RealTimeLog-KinesisDataStream Entspricht dem entsprechenden Artikel auf Ihrer CloudFront Rechnung.
Executions-CloudFrontFunctions Die Gebühr für die Anzahl der CloudFront Funktionsaufrufen.	Executions-CloudFrontFunctions Entspricht dem entsprechenden Artikel auf Ihrer CloudFront Rechnung.
region-Lambda-Edge-Request Die Gebühr für die Anzahl der Lambda@Edge-Funktionsaufrufe.	***region***-Lambda-Edge-Request Entspricht dem entsprechenden Artikel auf Ihrer CloudFront Rechnung.
region-Lambda-Edge-GB-Second Die Gebühr für den Zeitraum vom Aufruf Ihrer Lambda@Edge-Funktion bis zur Rückgabe einer Antwort oder Beendigung.	***region***-Lambda-Edge-GB-Second Entspricht dem entsprechenden Artikel auf Ihrer CloudFront Rechnung.
KeyValueStore-EdgeReads Die Gebühr für die Anzahl der Leseanruf e der CloudFront KeyValueStore Methoden`get()`,`exists()`, und`meta()`. Weitere Informationen finden Sie unter Hilfsmethoden für Schlüsselwertspeicher.	KeyValueStore-EdgeReads Entspricht dem entsprechenden Artikel auf Ihrer CloudFront Rechnung.

CloudFront Gebühren in Ihrer AWS Rechnung	Werte in der UsageType Spalte im AWS Nutzungsbericht
KeyValueStore-APIOperations Die Gebühr für die Anzahl der CloudFront KeyValueStore API-Aufrufe.	KeyValueStore-APIOperations Entspricht dem entsprechenden Artikel auf Ihrer CloudFront Rechnung.

CloudFront Berichte in der Konsole anzeigen

Jeder Bericht bietet detaillierte Informationen und Visualisierungen, sodass Sie die Bereitstellung von Inhalten optimieren, Leistungsengpässe identifizieren und datengestützte Entscheidungen treffen können. Ganz gleich, ob Sie die Cache-Effizienz überwachen, Verkehrsmuster analysieren oder Ihre Zuschauer besser verstehen möchten, mit diesen Berichten können Sie Ihre CloudFront Verteilungen effektiv überwachen und analysieren.

In der Konsole kannst du dir die folgenden Berichte über deine CloudFront Aktivitäten ansehen:

Themen

- Berichte zu CloudFront Cache-Statistiken anzeigen
- Berichte über CloudFront beliebte Objekte anzeigen
- Berichte mit den CloudFront häufigsten Referrern anzeigen
- CloudFront Nutzungsberichte anzeigen
- Berichte von CloudFront Zuschauern anzeigen

Die meisten dieser Berichte basieren auf den Daten in den CloudFront Zugriffsprotokollen, die detaillierte Informationen zu jeder eingegangenen Benutzeranfrage CloudFront enthalten. Sie müssen Zugriffsprotokolle nicht aktivieren, um die Berichte anzeigen zu können. Weitere Informationen finden Sie unter Zugriffsprotokolle (Standardprotokolle).

Berichte zu CloudFront Cache-Statistiken anzeigen

Der CloudFront Amazon-Cache-Statistikbericht enthält die folgenden Informationen:

- Gesamtzahl der Anforderungen – die Gesamtzahl der Anforderungen für alle HTTP-Statuscodes (z. B. 200 oder 404) und für alle Methoden (z. B. GET, HEAD oder POST)

- Prozentsatz der Zuschaueranfragen nach Ergebnistyp — Treffer, Fehlschläge und Fehler als Prozentsatz der gesamten Zuschaueranfragen für die ausgewählte CloudFront Verteilung
- An Viewer übertragene Bytes – Gesamtzahl der Bytes und Zahl der Bytes für Fehlschläge
- HTTP-Statuscodes – Viewer-Anforderungen nach HTTP-Statuscode
- Prozentsatz der GET-Anforderungen ohne abgeschlossenen Download – GET-Anforderungen des Viewers, bei denen der Download des angeforderten Objekts nicht abgeschlossen wurde, als Prozentsatz aller Anforderungen

Die Daten für diese Statistiken stammen aus derselben Quelle wie die CloudFront Zugriffsprotokolle. Sie müssen die Zugriffsprotokollierung jedoch nicht aktivieren, um Cache-Statistiken anzeigen zu können.

Sie können Diagramme für einen bestimmten Datumsbereich in den vergangenen 60 Tagen mit einzelnen Datenpunkten pro Stunde oder pro Tag anzeigen. In der Regel können Sie aktuelle Daten bis zu Anfragen anzeigen, die CloudFront erst vor einer Stunde empfangenen hat; gelegentlich werden Daten aber mit bis zu 24 Stunden Verspätung übermittelt.

Themen

- Berichte zu CloudFront Cache-Statistiken in der Konsole anzeigen
- Herunterladen von Daten im CSV-Format
- Wie sich Cache-Statistikdiagramme auf Daten in den CloudFront Standardprotokollen (Zugriffsprotokollen) beziehen

Berichte zu CloudFront Cache-Statistiken in der Konsole anzeigen

Sie können den CloudFront Cache-Statistikbericht in der Konsole anzeigen.

Um den CloudFront Cache-Statistikbericht anzuzeigen

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home.
2. Wählen Sie im Navigationsbereich Cache-Statistiken aus.
3. Wählen Sie im Bereich CloudFront Cache Statistics Reports unter Start Date (Startdatum) und End Date (Enddatum) den Datumsbereich aus, für den Sie die Diagramme zu Cache-Statistiken anzeigen möchten. Die verfügbaren Datumsbereiche sind von dem Wert abhängig, den Sie unter Granularity (Granularität) auswählen:

- Daily (Täglich) – Um Diagramme mit einem einzigen Datenpunkt pro Tag anzuzeigen, wählen Sie einen Datumsbereich in den letzten 60 Tagen aus.
- Hourly (Stündlich) – Um Diagramme mit einem einzigen Datenpunkt pro Stunde anzeigen, wählen Sie einen Datumsbereich von bis zu 14 Tagen in den letzten 60 Tagen aus.

Datum und Uhrzeit entsprechen der Zeitzone UTC (Coordinated Universal Time).

4. Wählen Sie unter Granularity (Granularität) aus, ob Sie Diagramme mit einem Datenpunkt pro Tag oder einem Datenpunkt pro Stunde anzeigen möchten. Wenn Sie einen Datumsbereich größer als 14 Tage angeben, ist die Option "Ein Datenpunkt pro Stunde" nicht verfügbar.
5. Wählen Sie unter Viewer Location (Standort des Betrachters) den Kontinent aus, von dem die Anfragen gesendet werden, oder wählen Sie die Option All Locations (Alle Standorte). Die Cache-Statistikdiagramme enthalten Daten für Anfragen, die vom angegebenen Standort CloudFront empfangen wurden.
6. Wählen Sie in der Liste Distribution die Verteilungen aus, für die Sie in den Nutzungsdiagrammen Daten anzeigen möchten:

   - Eine individuelle Verteilung — Die Diagramme zeigen Daten für die ausgewählte CloudFront Verteilung an. Unter Distribution werden die Verteilungs-ID und die alternativen Domain-Namen (CNAMEs) für die Verteilung angezeigt, sofern vorhanden. Wenn für eine Distribution keine alternativen Domainnamen vorhanden sind, enthält die Liste die Ursprungsdomainnamen für die Distribution.
   - Alle Verteilungen — Die Diagramme zeigen summierte Daten für alle Verteilungen an, die der aktuellen Verteilung zugeordnet sind, mit Ausnahme der Verteilungen AWS-Konto, die Sie gelöscht haben.
7. Wählen Sie Aktualisieren aus.

> **Tip**
>
> - Sie können die Daten für einen täglichen oder stündlichen Datenpunkt in einem Diagramm anzeigen, indem Sie den Mauszeiger auf dem Datenpunkt platzieren.
> - Bei Diagrammen, die weitergeleitete Daten anzeigen, können Sie die vertikale Skalierung in GB, MB oder KB ändern.

Herunterladen von Daten im CSV-Format

Sie können den Cache Statistics-Bericht im CSV-Format herunterladen. In diesem Abschnitt wird erklärt, wie der heruntergeladen wird. Außerdem werden die Werte in dem Bericht beschrieben.

So laden Sie den Cache Statistics-Bericht im CSV-Format herunter

1. Klicken Sie bei der Anzeige des Cache-Statistikberichts auf CSV.
2. Wählen Sie im Dialogfeld zum Öffnen der Datei, ob Sie die Datei öffnen oder speichern möchten.

Informationen zu dem Bericht

Die ersten Zeilen des Berichts enthalten die folgenden Informationen:

Version

: Die Version des Formats für diese CSV-Datei

Bericht

: Der Name des Berichts.

DistributionID

: Die ID der Verteilung, für die der Bericht ausgeführt wurde, oder `ALL`, wenn der Bericht für alle Verteilungen ausgeführt wurde

StartDateUTC

: Der Beginn des Datumsbereichs, für den Sie den Bericht ausgeführt haben, in koordinierter Weltzeit (Coordinated Universal Time, UTC).

EndDateUTC

: Das Ende des Datumsbereichs, für den Sie den Bericht ausgeführt haben, in koordinierter Weltzeit (Coordinated Universal Time, UTC).

GeneratedTimeUTC

: Das Datum und die Uhrzeit, zu der Sie den Bericht ausgeführt haben, in koordinierter Weltzeit (Coordinated Universal Time, UTC).

Granularity

: Ob die einzelnen Zeilen in dem Bericht für eine Stunde oder einen Tag stehen

ViewerLocation

Der Kontinent, von dem die Betrachteranfragen gesendet wurden, oder ALL, wenn Sie einen Bericht für alle Standorte herunterladen möchten

Daten im Cache Statistics-Bericht

Der Bericht enthält die folgenden Werte:

DistributionID

Die ID der Verteilung, für die der Bericht ausgeführt wurde, oder ALL, wenn der Bericht für alle Verteilungen ausgeführt wurde

FriendlyName

Ein alternativer Domänennamen (CNAMEs) für die Verteilung (sofern vorhanden).
Wenn eine Verteilung keine alternativen Domänennamen hat, enthält die Liste einen Ursprungsdomänennamen für die Verteilung.

ViewerLocation

Der Kontinent, von dem die Betrachteranfragen gesendet wurden, oder ALL, wenn Sie einen Bericht für alle Standorte herunterladen möchten

TimeBucket

Die Stunde bzw. der Tag, für den die Daten gelten, in koordinierter Weltzeit (Coordinated Universal Time, UTC).

RequestCount

Die Gesamtzahl der Anfragen für alle HTTP-Statuscodes (z. B. 200 oder 404) und alle Methoden (z. B. GET, HEAD oder POST)

HitCount

Die Anzahl der Viewer-Anfragen, für die das Objekt aus einem CloudFront Edge-Cache bedient wird.

MissCount

Die Anzahl der Viewer-Anfragen, für die sich das Objekt derzeit nicht in einem Edge-Cache befindet, also das Objekt von Ihrem Ursprung abrufen CloudFront muss.

ErrorCount

Die Anzahl der Betrachteranfragen, die zu einem Fehler führten, sodass CloudFront das Objekt nicht gesendet hat

IncompleteDownloadCount

Die Anzahl der Betrachteranfragen, für die der Betrachter den Download eines Objekts begonnen aber nicht abgeschlossen hat

HTTP2xx

Die Anzahl der Betrachteranfragen, für die als HTTP-Statuscode ein 2xx-Wert (erfolgreich abgeschlossen) zurückgegeben wurde

HTTP3xx

Die Anzahl der Betrachteranfragen, für die als HTTP-Statuscode ein 3xx-Wert (zusätzliche Aktion erforderlich) zurückgegeben wurde

HTTP4xx

Die Anzahl der Betrachteranfragen, für die als HTTP-Statuscode ein 4xx-Wert (Client-Fehler) zurückgegeben wurde

HTTP5xx

Die Anzahl der Betrachteranfragen, für die als HTTP-Statuscode ein 5xx-Wert (Server-Fehler) zurückgegeben wurde

TotalBytes

Die Gesamtzahl der Byte, die Viewern als Antwort CloudFront auf alle Anfragen für alle HTTP-Methoden zur Verfügung gestellt wurden.

BytesFromMisses

Die Anzahl der an Viewer übertragenen Bytes für Objekte, die zum Zeitpunkt der Anfrage nicht im Edge-Cache vorhanden waren. Dieser Wert ist ein guter Näherungswert für die Anzahl der Byte, die von Ihren Ursprungs- zu den CloudFront Edge-Caches übertragen werden. Hier sind allerdings Anfragen für Objekte, die sich bereits im Edge-Cache befinden aber abgelaufen sind, nicht enthalten.

Wie sich Cache-Statistikdiagramme auf Daten in den CloudFront Standardprotokollen (Zugriffsprotokollen) beziehen

Die folgende Tabelle zeigt, wie die Cache-Statistikdiagramme in der CloudFront Konsole den Werten in den CloudFront Zugriffsprotokollen entsprechen. Weitere Hinweise zu CloudFront Zugriffsprotokollen finden Sie unterZugriffsprotokolle (Standardprotokolle).

Anfragen insgesamt

Dieses Diagramm zeigt die Gesamtzahl der Anfragen für alle HTTP-Statuscodes (z. B. 200 oder 404) und alle Methoden (z. B. `GET`, `HEAD` oder `POST`) an. Die gesamten Anfragen in dieser Tabelle entsprechen der Gesamtzahl der Anfragen im Zugriffsprotokoll für den gleichen Zeitraum.

Prozentsatz der Betrachteranfragen nach Ergebnistyp

Dieses Diagramm zeigt Treffer, Fehlschläge und Fehler als Prozentsatz der gesamten Zuschaueranfragen für die ausgewählte CloudFront Distribution:

- Treffer — Eine Viewer-Anfrage, für die das Objekt aus einem CloudFront Edge-Cache bedient wird. In den Zugriffsprotokollen sind das die Anfragen, für die unter `x-edge-response-result-type` als Wert `Hit` aufgeführt ist.
- Fehlschlag — Eine Viewer-Anfrage, bei der sich das Objekt derzeit nicht in einem Edge-Cache befindet und das Objekt daher von Ihrem Ursprung abgerufen werden CloudFront muss. In den Zugriffsprotokollen sind das die Anfragen, für die unter `x-edge-response-result-type` als Wert `Miss` aufgeführt ist.
- Fehler — Eine Viewer-Anfrage, die zu einem Fehler geführt hat, das Objekt also CloudFront nicht bereitgestellt hat. In den Zugriffsprotokollen sind das die Anfragen, für die unter `x-edge-response-result-type` als Wert `Error`, `LimitExceeded` oder `CapacityExceeded` aufgeführt ist.

Das Diagramm enthält keine Aktualisierungstreffer. Dies sind Anforderungen für Objekte, die sich zwar im Edge-Cache befinden, aber abgelaufen sind. In den Zugriffsprotokollen sind das die Anfragen, für die unter `x-edge-response-result-type` als Wert `RefreshHit` aufgeführt ist.

An Betrachter weitergeleitete Bytes

Dieses Diagramm zeigt zwei Werte:

- Gesamtzahl der Byte — Die Gesamtzahl der Byte, die Viewern als Antwort CloudFront auf alle Anfragen für alle HTTP-Methoden zur Verfügung gestellt wurden. In CloudFront Zugriffsprotokollen ist die Gesamtzahl der Bytes die Summe der Werte in der `sc-bytes` Spalte für alle Anfragen im gleichen Zeitraum.

- Bytes from Misses (Bytes aus Fehlschlägen) – die Anzahl der Bytes, die Betrachtern für Objekte bereitgestellt wurden, die sich zum Zeitpunkt der Anforderung nicht im Edge-Cache befunden haben. In CloudFront Zugriffsprotokollen ist Byte aus Fehlschlägen die Summe der Werte in der `sc-bytes` Spalte für Anfragen, für die der Wert von `x-edge-result-type` ist`Miss`. Dieser Wert ist ein guter Näherungswert für die Anzahl der Byte, die von Ihrem Ursprung an die CloudFront Edge-Caches übertragen werden. Hier sind allerdings Anfragen für Objekte, die sich bereits im Edge-Cache befinden aber abgelaufen sind, nicht enthalten.

HTTP-Statuscodes

In diesem Diagramm werden Betrachteranfragen nach HTTP-Statuscodes angezeigt. In CloudFront den Zugriffsprotokollen werden Statuscodes in der `sc-status` folgenden Spalte angezeigt:

- 2xx – Die Anforderung war erfolgreich.
- 3xx – Es sind zusätzliche Aktionen erforderlich. Beispiel: 301 (dauerhaft verschoben) bedeutet, dass das angeforderte Objekt an einen anderen Speicherort verschoben wurde.
- 4xx – Für den Client ist anscheinend ein Fehler aufgetreten. Beispiel: 404 (nicht gefunden) bedeutet, dass der Client ein Objekt angefordert hat, das nicht gefunden wurde.
- 5xx – Der Ursprungs-Server hat nicht auf die Anforderung geantwortet. Beispiel: 503 (Service nicht verfügbar) bedeutet, dass der Ursprungsserver zurzeit nicht verfügbar ist.

Prozentsatz der GET-Anfragen, für die der Download nicht abgeschlossen wurde

In diesem Diagramm werden `GET`-Anfragen, bei denen der Download des angeforderten Objekts nicht abgeschlossen wurde, als Prozentsatz an den Anfragen insgesamt angezeigt. In der Regel wird das Herunterladen eines Objekts nicht abgeschlossen, weil der Betrachter den Download abgebrochen hat, z. B. indem er einen anderen Link ausgewählt oder den Browser geschlossen hat. In CloudFront Zugriffsprotokollen haben diese Anfragen den Wert `200` in der `sc-status` Spalte und den Wert `Error` in der `x-edge-result-type` Spalte.

Berichte über CloudFront beliebte Objekte anzeigen

Sehen Sie sich den Amazon-Bericht über CloudFront beliebte Objekte an, um die 50 beliebtesten Objekte für eine Verteilung in einem bestimmten Zeitraum in den letzten 60 Tagen zu sehen. Sie können auch Statistiken zu diesen Objekten einsehen, darunter die folgenden:

- Anzahl der Anforderungen für das Objekt
- Anzahl der Treffer und Fehler

- Trefferrate
- Anzahl der für Fehler bereitgestellten Bytes
- Gesamtzahl der bereitgestellten Bytes
- Anzahl unvollständiger Downloads
- Anzahl der Anforderungen nach HTTP-Statuscode (2xx, 3xx, 4xx und 5xx)

Die Daten für diese Statistiken stammen aus derselben Quelle wie die CloudFront Zugriffsprotokolle. Sie müssen die Zugriffsprotokollierung jedoch nicht aktivieren, um die beliebten Objekte anzeigen zu können.

Themen

- Sehen Sie sich Berichte über CloudFront beliebte Objekte in der Konsole an
- Wie CloudFront berechnet man Statistiken zu beliebten Objekten
- Herunterladen von Daten im CSV-Format
- Wie die Daten im Bericht über beliebte Objekte mit den Daten in den CloudFront Standardprotokollen (Zugriffsprotokollen) zusammenhängen

Sehen Sie sich Berichte über CloudFront beliebte Objekte in der Konsole an

Sie können den Bericht über CloudFront beliebte Objekte in der Konsole anzeigen.

So zeigen Sie beliebte Objekte für eine CloudFront Distribution an

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home.
2. Klicken Sie im Navigationsbereich auf Beliebte Objekte.
3. Wählen Sie im Bereich CF; Popular Objects ReportCloudFront (Popular Objects-Bericht) unter Start Date (Startdatum) und End Date (Enddatum) den Datumsbereich aus, für den Sie eine Liste der beliebten Objekte anzeigen möchten. Sie können alle Datumsbereiche in den vergangenen 60 Tagen auswählen.

   Datum und Uhrzeit entsprechen der Zeitzone UTC (Coordinated Universal Time).
4. Wählen Sie in der Liste Distribution die Verteilung aus, für die Sie eine Liste der beliebten Objekte anzeigen möchten.

5. Wählen Sie Aktualisieren aus.

Wie CloudFront berechnet man Statistiken zu beliebten Objekten

Um eine genaue Anzahl der 50 wichtigsten Objekte in Ihrer Verteilung zu erhalten, CloudFront zählen Sie die Anfragen für alle Ihre Objekte ab Mitternacht in Intervallen von 10 Minuten und behalten die 150 wichtigsten Objekte für die nächsten 24 Stunden bei. (behält CloudFront außerdem die täglichen Gesamtwerte für die 150 wichtigsten Objekte 60 Tage lang bei.)

Am unteren Ende der Liste kommen ständig neue Objekte hinzu, während andere von der Liste gestrichen werden, sodass die Summen für diese Objekte immer eine Annäherung darstellen. Die 50 Objekte ganz oben in der Liste der Top-150 können sich innerhalb der Liste ebenfalls nach oben oder unten bewegen, werden aber selten gestrichen, sodass die Summen für diese Objekte genauere Angaben darstellen.

Wenn ein Objekt aus der Liste der 150 besten Objekte fällt und dann im Laufe eines Tages wieder in die Liste aufsteigt, wird eine geschätzte Anzahl von Anfragen für den Zeitraum CloudFront hinzugefügt, in dem das Objekt in der Liste gefehlt hat. Die Schätzung basiert auf der Anzahl der Anfragen für das Objekt, das in diesem Zeitraum am Ende der Liste geführt wurde.

Wenn das Objekt später am Tag in die Top 50 der Objekte aufsteigt, führt die geschätzte Anzahl der Anfragen, die CloudFront empfangen wurden, während das Objekt nicht zu den obersten 150 Objekten gehörte, normalerweise dazu, dass die Anzahl der Anfragen im Bericht über beliebte Objekte die Anzahl der Anfragen übersteigt, die in den Zugriffsprotokollen für dieses Objekt erscheinen.

Herunterladen von Daten im CSV-Format

Sie können den Popular Objects-Bericht im CSV-Format herunterladen. In diesem Abschnitt wird erklärt, wie der heruntergeladen wird. Außerdem werden die Werte in dem Bericht beschrieben.

So laden Sie den Popular Objects-Bericht im CSV-Format herunter

1. Wählen Sie bei der Anzeige des Berichts der beliebtesten Objekte CSV aus.
2. Wählen Sie im Dialogfeld zum Öffnen der Datei, ob Sie die Datei öffnen oder speichern möchten.

Informationen zu dem Bericht

Die ersten Zeilen des Berichts enthalten die folgenden Informationen:

Version

Die Version des Formats für diese CSV-Datei

Bericht

Der Name des Berichts.

DistributionID

Die ID der Verteilung, für die Sie den Bericht ausgeführt haben.

StartDateUTC

Der Beginn des Datumsbereichs, für den Sie den Bericht ausgeführt haben, in koordinierter Weltzeit (Coordinated Universal Time, UTC).

EndDateUTC

Das Ende des Datumsbereichs, für den Sie den Bericht ausgeführt haben, in koordinierter Weltzeit (Coordinated Universal Time, UTC).

GeneratedTimeUTC

Das Datum und die Uhrzeit, zu der Sie den Bericht ausgeführt haben, in koordinierter Weltzeit (Coordinated Universal Time, UTC).

Daten im Popular Objects-Bericht

Der Bericht enthält die folgenden Werte:

DistributionID

Die ID der Verteilung, für die Sie den Bericht ausgeführt haben.

FriendlyName

Ein alternativer Domänennamen (CNAMEs) für die Verteilung (sofern vorhanden). Wenn eine Verteilung keine alternativen Domänennamen hat, enthält die Liste einen Ursprungsdomänennamen für die Verteilung.

Objekt

Die letzten 500 Zeichen der URL für das Objekt

RequestCount

Die Gesamtzahl der Anfragen für dieses Objekt

HitCount

Die Anzahl der Viewer-Anfragen, für die das Objekt aus einem CloudFront Edge-Cache bedient wird.

MissCount

Die Anzahl der Viewer-Anfragen, für die sich das Objekt derzeit nicht in einem Edge-Cache befindet, also das Objekt von Ihrem Ursprung abrufen CloudFront muss.

HitCountPct

Der Wert von `HitCount` als Prozentsatz des Werts unter `RequestCount`

BytesFromMisses

Das Volumen der an die Viewer übertragenen Bytes für dieses Objekt, wenn es zum Zeitpunkt der Anfrage nicht im Edge-Cache vorhanden war

TotalBytes

Die Gesamtzahl der Byte, die Viewern CloudFront für dieses Objekt als Antwort auf alle Anfragen für alle HTTP-Methoden zur Verfügung gestellt wurden.

IncompleteDownloadCount

Die Anzahl der Betrachteranfragen für dieses Objekt, für die der Betrachter den Download eines Objekts begonnen aber nicht abgeschlossen hat

HTTP2xx

Die Anzahl der Betrachteranfragen, für die als HTTP-Statuscode ein 2xx-Wert (erfolgreich abgeschlossen) zurückgegeben wurde

HTTP3xx

Die Anzahl der Betrachteranfragen, für die als HTTP-Statuscode ein 3xx-Wert (zusätzliche Aktion erforderlich) zurückgegeben wurde

HTTP4xx

Die Anzahl der Betrachteranfragen, für die als HTTP-Statuscode ein 4xx-Wert (Client-Fehler) zurückgegeben wurde

HTTP5xx

Die Anzahl der Betrachteranfragen, für die als HTTP-Statuscode ein 5xx-Wert (Server-Fehler) zurückgegeben wurde

Wie die Daten im Bericht über beliebte Objekte mit den Daten in den CloudFront Standardprotokollen (Zugriffsprotokollen) zusammenhängen

Die folgende Liste zeigt, wie die Werte im Bericht über beliebte Objekte in der CloudFront Konsole den Werten in den CloudFront Zugriffsprotokollen entsprechen. Weitere Informationen zu CloudFront-Zugriffsprotokollen finden Sie unter Zugriffsprotokolle (Standardprotokolle).

URL

Die letzten 500 Zeichen der URL, die Betrachter verwenden, um auf das Objekt zuzugreifen

Anforderungen

Die Gesamtzahl der Anfragen für das Objekt Dieser Wert entspricht im Allgemeinen weitgehend der Anzahl der `GET` Anfragen für das Objekt in den CloudFront Zugriffsprotokollen.

Treffer

Die Anzahl der Betrachteranfragen, für die das Objekt aus einem CloudFront -Edge-Cache gesendet wurde. In den Zugriffsprotokollen sind das die Anfragen, für die unter `x-edge-response-result-type` als Wert `Hit` aufgeführt ist.

Fehlgriffe

Die Anzahl der Viewer-Anfragen, bei denen sich das Objekt nicht in einem Edge-Cache befand, also das Objekt von Ihrem Ursprung CloudFront abgerufen wurde. In den Zugriffsprotokollen sind das die Anfragen, für die unter `x-edge-response-result-type` als Wert `Miss` aufgeführt ist.

Trefferrate

Der Wert in der Spalte Hits (Treffer) als Prozentsatz des Werts in der Spalte Requests (Anfragen).

Bytes für Fehlgriffe

Die Anzahl der an Viewer übertragenen Bytes für Objekte, die zum Zeitpunkt der Anfrage nicht im Edge-Cache vorhanden waren. In CloudFront Zugriffsprotokollen ist Byte aus Fehlschlägen die Summe der Werte in der `sc-bytes` Spalte für Anfragen, für die der Wert von `x-edge-result-type` ist`Miss`.

Bytes insgesamt

Die Gesamtzahl der Byte, die den Viewern als Antwort auf alle Anfragen für das Objekt für alle HTTP-Methoden zur Verfügung CloudFront gestellt wurden. In CloudFront Zugriffsprotokollen ist die Gesamtzahl der Byte die Summe der Werte in der `sc-bytes` Spalte für alle Anfragen im gleichen Zeitraum.

Unvollständige Downloads

Die Anzahl der Betrachteranfragen, für die der Download des angeforderten Objekts nicht abgeschlossen wurde. In der Regel liegt der Grund dafür, dass ein Download nicht abgeschlossen wird, darin, dass der Zuschauer ihn abgebrochen hat, z. B. indem er einen anderen Link ausgewählt oder den Browser geschlossen hat. In CloudFront Zugriffsprotokollen haben diese Anfragen den Wert `200` in der `sc-status` Spalte und den Wert von `Error` in der `x-edge-result-type` Spalte.

2xx

Die Anzahl der Anfragen mit dem HTTP-Statuscode `2xx`, `Successful`. In CloudFront Zugriffsprotokollen werden Statuscodes in der `sc-status` Spalte angezeigt.

3xx

Die Anzahl der Betrachteranfragen, für die als HTTP-Statuscode ein `3xx`, `Redirection`. `3xx` Statuscodes zeigen an, dass zusätzliche Aktionen erforderlich sind. Beispiel: 301 (dauerhaft verschoben) bedeutet, dass das angeforderte Objekt an einen anderen Speicherort verschoben wurde.

4xx

Die Anzahl der Betrachteranfragen, für die als HTTP-Statuscode ein `4xx`, `Client Error`. `4xx` Statuscodes zeigen an, dass zusätzliche Aktionen erforderlich sind. Beispiel: 404 (nicht gefunden) bedeutet, dass der Client ein Objekt angefordert hat, das nicht gefunden wurde.

5xx

Die Anzahl der Betrachteranfragen, für die als HTTP-Statuscode ein `5xx`, `Server Error`. `5xx` Statuscodes zeigen an, dass der Ursprungs-Server nicht auf die Anfrage geantwortet hat. Beispiel: 503 (Service nicht verfügbar) bedeutet, dass der Ursprungsserver zurzeit nicht verfügbar ist.

Berichte mit den CloudFront häufigsten Referrern anzeigen

Der CloudFront Bericht über die häufigsten Referrer enthält die folgenden Informationen für einen beliebigen Zeitraum der letzten 60 Tage:

- Die 25 häufigsten Referrer (Domains von Websites, die die meisten HTTP- und HTTPS-Anfragen für Objekte gesendet haben, die CloudFront für Ihre Distribution verteilt werden)
- Anzahl der Anforderungen von einem Referrer

- Anzahl der von einem Referrer gesendeten Anforderungen als Prozentsatz aller Anforderungen während des ausgewählten Zeitraums

Die Daten für den Bericht mit den häufigsten Referrern stammen aus derselben Quelle wie die Zugriffsprotokolle. CloudFront Sie müssen die Zugriffsprotokollierung jedoch nicht aktivieren, um die Liste der häufigsten Referrer anzeigen zu können.

Bei den häufigsten Referrern kann es sich um Suchmaschinen, andere Websites mit Direktlinks zu Ihren Objekten oder Ihre eigene Website handeln. Wenn z. B. `https://example.com/index.html` auf 10 Grafiken verlinkt, dann ist `example.com` der Referrer für alle 10 Grafiken.

> **Note**
>
> Wenn ein Benutzer eine URL direkt in die Adressleiste des Browsers eingibt, wird kein Referrer für das angeforderte Objekt gesendet.

Themen

- Sehen Sie sich die CloudFront Berichte mit den häufigsten Referrern in der Konsole an
- Wie CloudFront berechnet man die Statistik der Top-Referrer
- Herunterladen von Daten im CSV-Format
- Wie die Daten im Bericht mit den häufigsten Referrern mit den Daten in den CloudFront Standardprotokollen (Zugriffsprotokollen) zusammenhängen

Sehen Sie sich die CloudFront Berichte mit den häufigsten Referrern in der Konsole an

Sie können den Bericht über die häufigsten CloudFront Referrer in der Konsole einsehen.

Um die Top-Referrer für eine Distribution anzuzeigen CloudFront

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CloudFront Konsole unter. https://console.aws.amazon.com/cloudfront/v4/home
2. Klicken Sie im Navigationsbereich auf Häufigste Referrer.
3. Wählen Sie im Bereich CF; Top Referrers ReportCloudFront (Top Referrers-Bericht) unter Start Date (Startdatum) und End Date (Enddatum) den Datumsbereich aus, für den Sie eine Liste der häufigsten Referrer anzeigen möchten.

Datum und Uhrzeit entsprechen der Zeitzone UTC (Coordinated Universal Time).

4. Wählen Sie in der Liste Distribution die Verteilung aus, für die Sie eine Liste der häufigsten Referrer anzeigen möchten.
5. Wählen Sie Aktualisieren aus.

Wie CloudFront berechnet man die Statistik der Top-Referrer

Zählt die Anfragen für alle Ihre Objekte in Intervallen von 10 Minuten, um eine genaue CloudFront Zählung der 25 häufigsten Referrer zu erhalten. Die 75 besten Referrer werden als Gesamtwert berechnet. Am unteren Ende der Liste kommen ständig neue Referrer hinzu, während andere von der Liste gestrichen werden, sodass die Summen für diese Referrer immer eine Annäherung darstellen.

Die 25 Referrer ganz oben in der Liste der 75 Referrer können sich innerhalb der Liste ebenfalls nach oben oder unten bewegen, werden aber selten gestrichen, sodass die Summen für diese Referrer in der Regel genauere Angaben darstellen.

Herunterladen von Daten im CSV-Format

Sie können den Top Referrers-Bericht im CSV-Format herunterladen. In diesem Abschnitt wird erklärt, wie der heruntergeladen wird. Außerdem werden die Werte in dem Bericht beschrieben.

So laden Sie den Top Referrers-Bericht im CSV-Format herunter

1. Wählen Sie bei der Anzeige des Berichts zu den häufigsten Referrern CSV aus.
2. Wählen Sie im Dialogfeld zum Öffnen der Datei, ob Sie die Datei öffnen oder speichern möchten.

Informationen zu dem Bericht

Die ersten Zeilen des Berichts enthalten die folgenden Informationen:

Version

Die Version des Formats für diese CSV-Datei

Bericht

Der Name des Berichts.

DistributionID

Die ID der Verteilung, für die der Bericht ausgeführt wurde, oder ALL, wenn der Bericht für alle Verteilungen ausgeführt wurde

StartDateUTC

Der Beginn des Datumsbereichs, für den Sie den Bericht ausgeführt haben, in koordinierter Weltzeit (Coordinated Universal Time, UTC).

EndDateUTC

Das Ende des Datumsbereichs, für den Sie den Bericht ausgeführt haben, in koordinierter Weltzeit (Coordinated Universal Time, UTC).

GeneratedTimeUTC

Das Datum und die Uhrzeit, zu der Sie den Bericht ausgeführt haben, in koordinierter Weltzeit (Coordinated Universal Time, UTC).

Daten im Top Referrers-Bericht

Der Bericht enthält die folgenden Werte:

DistributionID

Die ID der Verteilung, für die der Bericht ausgeführt wurde, oder ALL, wenn der Bericht für alle Verteilungen ausgeführt wurde

FriendlyName

Ein alternativer Domänennamen (CNAMEs) für die Verteilung (sofern vorhanden).
Wenn eine Verteilung keine alternativen Domänennamen hat, enthält die Liste einen Ursprungsdomänennamen für die Verteilung.

Referrer

Der Domain-Name des Referrers

RequestCount

Die Gesamtzahl der Anfragen von dem Domain-Namen in der Spalte `Referrer`

RequestsPct

Die Anzahl der von einem Referrer gesendeten Anfragen als Prozentsatz an der Gesamtzahl der Anfragen während des angegebenen Zeitraums

Wie die Daten im Bericht mit den häufigsten Referrern mit den Daten in den CloudFront Standardprotokollen (Zugriffsprotokollen) zusammenhängen

Die folgende Liste zeigt, wie die Werte im Bericht „Top Referrers“ in der CloudFront Konsole den Werten in den Zugriffsprotokollen entsprechen. CloudFront Weitere Informationen zu CloudFront-Zugriffsprotokollen finden Sie unter Zugriffsprotokolle (Standardprotokolle).

Referrer

Der Domain-Name des Referrers In den Zugriffsprotokollen werden die Referrer in der Spalte `cs(Referer)` aufgeführt.

Anzahl der Anfragen

Die Gesamtzahl der Anfragen von dem Domain-Namen in der Spalte Referrer. Dieser Wert entspricht im Allgemeinen weitgehend der Anzahl der `GET` Anfragen des Referrers in CloudFront den Zugriffsprotokollen.

Anfrage %

Die Anzahl der von einem Referrer gesendeten Anfragen als Prozentsatz an der Gesamtzahl der Anfragen während des angegebenen Zeitraums Wenn mehr als 25 Referrer vorhanden sind, können Sie Request % (Anfrage %) nicht basierend auf den Daten in dieser Tabelle berechnen, da in der Spalte Request Count (Anzahl der Anfragen) nicht alle Anfragen für den angegebenen Zeitraum enthalten sind.

CloudFront Nutzungsberichte anzeigen

Die CloudFront Nutzungsberichte enthalten die folgenden Informationen:

- Anzahl der Anfragen — Zeigt die Gesamtzahl der Anfragen an, die in jedem Zeitintervall für die angegebene CloudFront Verteilung von Edge-Standorten in der ausgewählten Region CloudFront beantwortet wurden.
- Per Protokoll übertragene Daten und per Ziel übertragene Daten — Beide zeigen die Gesamtmenge der Daten, die während jedes Zeitintervalls für die angegebene CloudFront Verteilung von CloudFront Edge-Standorten in der ausgewählten Region übertragen wurden. Sie trennen die Daten auf unterschiedliche Weise, wie folgt:
  - Nach Protokoll – Trennt die Daten nach Protokoll: HTTP oder HTTPS.
  - Nach Ziel – trennt die Daten nach Ziel: zu Ihren Viewern oder zu Ihrem Ursprung.

Der CloudFront Nutzungsbericht basiert auf dem AWS Nutzungsbericht für CloudFront. Dieser Bericht erfordert keine zusätzliche Konfiguration. Weitere Informationen finden Sie unter Sehen Sie sich das an AWS Nutzungsbericht für CloudFront.

Sie können Berichte für einen bestimmten Datumsbereich in den vergangenen 60 Tagen mit einzelnen Datenpunkten pro Stunde oder pro Tag anzeigen. In der Regel können Sie Daten zu Anfragen einsehen, die erst vor vier Stunden CloudFront eingegangen sind. Gelegentlich können sich Daten jedoch um bis zu 24 Stunden verzögern.

Weitere Informationen finden Sie unter Wie hängen die Nutzungsdiagramme mit den Daten im CloudFront Nutzungsbericht zusammen.

Themen

- CloudFront Nutzungsberichte in der Konsole anzeigen
- Herunterladen von Daten im CSV-Format
- Wie hängen die Nutzungsdiagramme mit den Daten im CloudFront Nutzungsbericht zusammen

CloudFront Nutzungsberichte in der Konsole anzeigen

Sie können den CloudFront Nutzungsbericht in der Konsole anzeigen.

Um CloudFront Nutzungsberichte einzusehen

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home.
2. Klicken Sie im Navigationsbereich auf Nutzungsberichte.
3. Wählen Sie im Bereich CloudFront Cache Usage Reports unter Start Date (Startdatum) und End Date (Enddatum) den Datumsbereich aus, für den Sie die Nutzungsdiagramme anzeigen möchten. Die verfügbaren Datumsbereiche sind von dem Wert abhängig, den Sie unter Granularity (Granularität) auswählen:

   - Daily (Täglich) – Um Diagramme mit einem einzigen Datenpunkt pro Tag anzuzeigen, wählen Sie einen Datumsbereich in den letzten 60 Tagen aus.
   - Hourly (Stündlich) – Um Diagramme mit einem einzigen Datenpunkt pro Stunde anzeigen, wählen Sie einen Datumsbereich von bis zu 14 Tagen in den letzten 60 Tagen aus.

   Datum und Uhrzeit entsprechen der Zeitzone UTC (Coordinated Universal Time).

4. Wählen Sie unter Granularity (Granularität) aus, ob Sie Diagramme mit einem Datenpunkt pro Tag oder einem Datenpunkt pro Stunde anzeigen möchten. Wenn Sie einen Datumsbereich größer als 14 Tage angeben, ist die Option "Ein Datenpunkt pro Stunde" nicht verfügbar.
5. Wählen Sie als Abrechnungsregion die CloudFront Abrechnungsregion aus, die die Daten enthält, die Sie anzeigen möchten, oder wählen Sie Alle Regionen aus. Nutzungsdiagramme enthalten Daten für Anfragen, die CloudFront an Edge-Standorten in der angegebenen Region verarbeitet. Die Region, in der Anfragen CloudFront bearbeitet werden, entspricht möglicherweise nicht dem Standort deiner Zuschauer.

   Wählen Sie nur Regionen aus, die in der Preisklasse für Ihre Distribution enthalten sind. Andernfalls enthalten die Nutzungsdiagramme möglicherweise keine Daten. Wenn du beispielsweise die Preisklasse 200 für deinen Vertrieb ausgewählt hast, sind die Abrechnungsregionen Südamerika und Australien nicht enthalten, sodass deine Anfragen aus diesen Regionen in der CloudFront Regel nicht bearbeitet werden. Weitere Informationen zu Preisklassen finden Sie unter CloudFront Preisgestaltung.
6. Wählen Sie in der Liste Distribution die Verteilungen aus, für die Sie in den Nutzungsdiagrammen Daten anzeigen möchten:

   - Eine individuelle Verteilung — Die Diagramme zeigen Daten für die ausgewählte CloudFront Verteilung an. Unter Distribution werden die Verteilungs-ID und die alternativen Domain-Namen (CNAMEs) für die Verteilung angezeigt, sofern vorhanden. Wenn für eine Verteilung keine alternativen Domain-Namen vorhanden sind, enthält die Liste die Domain-Namen der Ursprungsserver für die Verteilung.
   - Alle Verteilungen (ausgenommen gelöschte) — Die Diagramme zeigen summierte Daten für alle Verteilungen an, die dem aktuellen AWS Konto zugeordnet sind, mit Ausnahme von Verteilungen, die Sie gelöscht haben.
   - Alle gelöschten Verteilungen — In den Diagrammen werden summierte Daten für alle Verteilungen angezeigt, die dem aktuellen AWS Konto zugeordnet sind und in den letzten 60 Tagen gelöscht wurden.
7. Wählen Sie Diagramm aktualisieren aus.

> **Tip**
>
> - Sie können die Daten für einen täglichen oder stündlichen Datenpunkt in einem Diagramm anzeigen, indem Sie den Mauszeiger auf dem Datenpunkt platzieren.

- Beachten Sie bei Diagrammen, die weitergeleitete Daten anzeigen, dass Sie die vertikale Skalierung in GB, MB oder KB ändern können.

Herunterladen von Daten im CSV-Format

Sie können den Nutzungsbericht im CSV-Format herunterladen. In diesem Abschnitt wird erklärt, wie der heruntergeladen wird. Außerdem werden die Werte in dem Bericht beschrieben.

So laden Sie den Usage-Bericht im CSV-Format herunter

1. Wählen Sie bei der Anzeige des Nutzungsberichts CSV aus.
2. Wählen Sie im Dialogfeld zum Öffnen der Datei, ob Sie die Datei öffnen oder speichern möchten.

Informationen zu dem Bericht

Die ersten Zeilen des Berichts enthalten die folgenden Informationen:

Version

Die Version des Formats für diese CSV-Datei

Bericht

Der Name des Berichts.

DistributionID

Die ID der Verteilung, für die der Bericht ausgeführt wurde; oder `ALL`, wenn der Bericht für alle Verteilungen ausgeführt wurde; oder `ALL_DELETED`, wenn der Bericht für alle gelöschten Verteilungen ausgeführt wurde

StartDateUTC

Der Beginn des Datumsbereichs, für den Sie den Bericht ausgeführt haben, in koordinierter Weltzeit (Coordinated Universal Time, UTC).

EndDateUTC

Das Ende des Datumsbereichs, für den Sie den Bericht ausgeführt haben, in koordinierter Weltzeit (Coordinated Universal Time, UTC).

GeneratedTimeUTC

Das Datum und die Uhrzeit, zu der Sie den Bericht ausgeführt haben, in koordinierter Weltzeit (Coordinated Universal Time, UTC).

Granularity

Ob die einzelnen Zeilen in dem Bericht für eine Stunde oder einen Tag stehen

BillingRegion

Der Kontinent, von dem die Betrachteranfragen gesendet wurden, oder `ALL`, wenn Sie einen Bericht für alle Fakturierungsregionen herunterladen möchten

Daten im Usage-Bericht

Der Bericht enthält die folgenden Werte:

DistributionID

Die ID der Verteilung, für die der Bericht ausgeführt wurde; oder `ALL`, wenn der Bericht für alle Verteilungen ausgeführt wurde; oder `ALL_DELETED`, wenn der Bericht für alle gelöschten Verteilungen ausgeführt wurde

FriendlyName

Ein alternativer Domänennamen (CNAMEs) für die Verteilung (sofern vorhanden). Wenn eine Verteilung keine alternativen Domänennamen hat, enthält die Liste einen Ursprungsdomänennamen für die Verteilung.

BillingRegion

Die CloudFront Abrechnungsregion, für die Sie den Bericht erstellt haben, oder`ALL`.

TimeBucket

Die Stunde bzw. der Tag, für den die Daten gelten, in koordinierter Weltzeit (Coordinated Universal Time, UTC).

HTTP

Die Anzahl der HTTP-Anfragen, die in jedem Zeitintervall für die angegebene CloudFront Verteilung von Edge-Standorten in der ausgewählten Region CloudFront beantwortet wurden. Gültige Werte sind:

- Die Anzahl der HEAD Anfragen GET und Anfragen, die CloudFront dazu führen, dass Daten an Ihre Zuschauer übertragen werden
- Die Anzahl derDELETE,, OPTIONSPATCH, und PUT AnfragenPOST, die dazu führen, dass Daten CloudFront an deinen Absender übertragen werden

HTTPS

Die Anzahl der HTTPS-Anfragen, die in jedem Zeitintervall für die angegebene CloudFront Verteilung von Edge-Standorten in der ausgewählten Region CloudFront beantwortet wurden. Gültige Werte sind:

- Die Anzahl der HEAD Anfragen GET und Anfragen, die CloudFront dazu führen, dass Daten an Ihre Zuschauer übertragen werden
- Die Anzahl derDELETE,, OPTIONSPATCH, und PUT AnfragenPOST, die dazu führen, dass Daten CloudFront an deinen Absender übertragen werden

HTTPBytes

Die Gesamtmenge der Daten, die über HTTP von CloudFront Edge-Standorten in der ausgewählten Abrechnungsregion während des Zeitraums für die angegebene CloudFront Verteilung übertragen wurden. Gültige Werte sind:

- Daten, die als Antwort CloudFront auf HEAD Anfragen GET und Anfragen von Ihren Zuschauern übertragen wurden
- Daten, die von deinen Zuschauern an CloudFront FormulareDELETE,OPTIONS, PATCHPOST, und PUT Anfragen übertragen wurden
- Daten CloudFront , die von deinen Zuschauern als Antwort aufDELETE, OPTIONS PATCHPOST, und PUT Anfragen übertragen wurden

HTTPSBytes

Die Gesamtmenge der Daten, die über HTTPS von CloudFront Edge-Standorten in der ausgewählten Abrechnungsregion während des Zeitraums für die angegebene CloudFront Verteilung übertragen wurden. Gültige Werte sind:

- Daten, die als Antwort CloudFront auf HEAD Anfragen GET und Anfragen von Ihren Zuschauern übertragen wurden
- Daten, die von deinen Zuschauern an CloudFront FormulareDELETE,OPTIONS, PATCHPOST, und PUT Anfragen übertragen wurden
- Daten CloudFront , die von deinen Zuschauern als Antwort aufDELETE, OPTIONS PATCHPOST, und PUT Anfragen übertragen wurden

BytesIn

Die Gesamtmenge der Daten, die für`DELETE`, `OPTIONS` `PATCH``POST`, und `PUT` Anfragen in der ausgewählten Region während jedes Zeitintervalls für die angegebene CloudFront Verteilung von zu deinem Ausgangspunkt übertragen wurden. CloudFront

BytesOut

Die Gesamtmenge der Daten, die in jedem Zeitintervall für die angegebene CloudFront Verteilung über HTTP und HTTPS von CloudFront Ihren Zuschauern in der ausgewählten Region übertragen wurden. Gültige Werte sind:

- Daten, die als Antwort CloudFront auf `GET` und `HEAD` Anfragen von Ihren Zuschauern an Ihre Zuschauer übertragen wurden
- Daten CloudFront , die als Antwort auf`DELETE`,`OPTIONS`, `PATCH` und `PUT` Anfragen von Ihren Zuschauern an Ihre Zuschauer übertragen wurden `POST`

Wie hängen die Nutzungsdiagramme mit den Daten im CloudFront Nutzungsbericht zusammen

Die folgende Liste zeigt, wie die Nutzungsdiagramme in der CloudFront Konsole den Werten in der Spalte Verwendungstyp im CloudFront Nutzungsbericht entsprechen.

Themen

- Anzahl der Anfragen
- Weitergeleitete Daten nach Protokoll
- Weitergeleitete Daten nach Zieladresse

Anzahl der Anfragen

Dieses Diagramm zeigt die Gesamtzahl der Anfragen, die von Edge-Standorten in der ausgewählten Region in jedem Zeitintervall für die angegebene CloudFront Verteilung CloudFront beantwortet wurden, getrennt nach Protokoll (HTTP oder HTTPS) und Typ (statisch, dynamisch oder Proxy).

Anzahl der HTTP-Anfragen

- *`region`*-Requests-HTTP-Static: Anzahl der HTTP `GET` - und `HEAD` Requests, die für Objekte mit TTL ≥ 3600 Sekunden bedient wurden

- *region*-Requests-HTTP-Dynamic: Anzahl der HTTP GET - und HEAD Requests, die für Objekte mit TTL < 3600 Sekunden bedient wurden
- *region*-Requests-HTTP-Proxy: Anzahl der HTTP-DELETE,, OPTIONSPATCH, und PUT AnfragenPOST, die an Ihren Ursprung CloudFront weitergeleitet werden

Anzahl der HTTPS-Anfragen

- *region*-Requests-HTTPS-Static: Anzahl der HTTPS GET - und HEAD Requests, die für Objekte mit TTL ≥ 3600 Sekunden bedient wurden
- *region*-Requests-HTTPS-Dynamic: Anzahl von HTTPS GET und HEAD Anfragen, die für Objekte mit TTL < 3600 Sekunden bedient wurden
- *region*-Requests-HTTPS-Proxy: Anzahl der HTTPS-DELETE,OPTIONS, und PUT Anfragen PATCHPOST, die an Ihren Ursprung CloudFront weitergeleitet werden

Weitergeleitete Daten nach Protokoll

Dieses Diagramm zeigt die Gesamtmenge der Daten, die während jedes Zeitintervalls für die angegebene CloudFront Verteilung von CloudFront Edge-Standorten in der ausgewählten Region übertragen wurden, getrennt nach Protokoll (HTTP oder HTTPS), Typ (statisch, dynamisch oder Proxy) und Ziel (Zuschauer oder Herkunft).

Über HTTP weitergeleitete Daten

- *region*—Out-Bytes-HTTP-Static: Byte, die mithilfe von HTTP für Objekte mit TTL ≥ 3600 Sekunden bereitgestellt werden
- *region*-Out-Bytes-HTTP-Dynamic: Bytes, die mithilfe von HTTP für Objekte mit TTL < 3600 Sekunden bereitgestellt werden
- *region*-Out-Bytes-HTTP-Proxy: Bytes, die von CloudFront Viewern mithilfe von HTTP als Antwort aufDELETE, OPTIONS PATCHPOST, und Anfragen zurückgegeben wurden PUT
- *region*-Out-OBytes-HTTP-Proxy: Gesamtzahl der Bytes, die als Antwort auf,, und PUT Anfragen mithilfe von HTTP von CloudFront Edge-Standorten zu DELETE Ihrem Ursprung übertragen wurden OPTIONS PATCH POST

Über HTTPS weitergeleitete Daten

- *region*-Out-Bytes-HTTPS-Static: Bytes, die mithilfe von HTTPS für Objekte mit TTL ≥ 3600 Sekunden bereitgestellt werden
- *region*-Out-Bytes-HTTPS-Dynamic: Bytes, die mithilfe von HTTPS für Objekte mit TTL < 3600 Sekunden bereitgestellt werden

- *region*-Out-Bytes-HTTPS-Proxy: Bytes, die mithilfe von CloudFront HTTPS als Antwort auf`DELETE`,`OPTIONS`, `PATCH` und Anfragen von den Viewern zurückgegeben wurden `POST` `PUT`
- *region*-Out-OBytes-HTTPS-Proxy: Gesamtzahl der Bytes, die als Antwort auf,, und `PUT` Anfragen mithilfe von HTTPS von CloudFront Edge-Standorten zu `DELETE` Ihrem Ursprung übertragen wurden `OPTIONS` `PATCH` `POST`

Weitergeleitete Daten nach Zieladresse

Dieses Diagramm zeigt die Gesamtmenge der Daten, die während jedes Zeitintervalls für die angegebene CloudFront Verteilung von CloudFront Edge-Standorten in der ausgewählten Region übertragen wurden, getrennt nach Ziel (Viewer oder Herkunft), Protokoll (HTTP oder HTTPS) und Typ (statisch, dynamisch oder Proxy).

Daten, die von CloudFront an Ihre Zuschauer übertragen wurden

- *region*-Out-Bytes-HTTP-Static: Bytes, die mithilfe von HTTP für Objekte mit TTL ≥ 3600 Sekunden bereitgestellt werden
- *region*-Out-Bytes-HTTPS-Static: Bytes, die mithilfe von HTTPS für Objekte mit TTL ≥ 3600 Sekunden bereitgestellt werden
- *region*-Out-Bytes-HTTP-Dynamic: Bytes, die mithilfe von HTTP für Objekte mit TTL < 3600 Sekunden bereitgestellt werden
- *region*-Out-Bytes-HTTPS-Dynamic: Bytes, die mithilfe von HTTPS für Objekte mit TTL < 3600 Sekunden bereitgestellt werden
- *region*-Out-Bytes-HTTP-Proxy: Bytes, die von CloudFront Viewern mithilfe von HTTP als Antwort auf`DELETE`, `OPTIONS` `PATCHPOST`, und Anfragen zurückgegeben wurden `PUT`
- *region*-Out-Bytes-HTTPS-Proxy: BytesCloudFront , die von Zuschauern zurückgegeben wurden, indem HTTPS als Antwort auf`DELETE`, `OPTIONS` `PATCHPOST`, und `PUT` Anfragen verwendet wurde

Daten, die von CloudFront zu deinem Ursprung übertragen wurden

- *region*-Out-OBytes-HTTP-Proxy: Gesamtzahl der Bytes, die als Antwort auf,, und `PUT` Anfragen mithilfe von HTTP von CloudFront Edge-Standorten zu `DELETE` Ihrem Ursprung übertragen wurden `OPTIONS` `PATCH` `POST`
- *region*-Out-OBytes-HTTPS-Proxy: Gesamtzahl der Bytes, die als Antwort auf,, und `PUT` Anfragen mithilfe von HTTPS von CloudFront Edge-Standorten zu `DELETE` Ihrem Ursprung übertragen wurden `OPTIONS` `PATCH` `POST`

Berichte von CloudFront Zuschauern anzeigen

Die CloudFront Zuschauerberichte enthalten die folgenden Informationen für einen beliebigen Zeitraum der letzten 60 Tage:

- Geräte – die Arten von Geräten, die am häufigsten für den Zugriff auf Ihre Inhalte verwendet werden (z. B. Desktop oder Handy)
- Browser – die zehn Browser, die am häufigsten für den Zugriff auf Ihre Inhalte verwendet werden (wie Chrome oder Firefox)
- Betriebssysteme – die 10 Betriebssysteme, die am häufigsten für den Zugriff auf Ihre Inhalte verwendet werden (wie Linux, macOS oder Windows)
- Standorte — Die 50 besten Standorte (Länder oder U.S. states/territories) der Zuschauer, die am häufigsten auf deine Inhalte zugreifen
  - Sie können die Standorte auch mit Datenpunkten für jede Stunde in einem beliebigen Datumsbereich von bis zu 14 Tagen in den vergangenen 60 Tagen anzeigen.

> **Note**
>
> Sie müssen die Zugriffsprotokollierung nicht aktivieren, um Viewer-Diagramme und -Berichte anzeigen zu können.

Themen

- Anzeigen von Viewer-Diagrammen und -Berichten in der Konsole
- Herunterladen von Daten im CSV-Format
- In den Viewer-Berichten enthaltene Daten
- Wie die Daten im Standortbericht mit den Daten in den CloudFront Standardprotokollen (Zugriffsprotokollen) zusammenhängen

Anzeigen von Viewer-Diagrammen und -Berichten in der Konsole

Du kannst dir die Charts und Berichte der CloudFront Zuschauer in der Konsole ansehen.

Um Diagramme und Berichte von CloudFront Zuschauern anzusehen

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home.
2. Wählen Sie im Navigationsbereich Viewer aus.
3. Wählen Sie im Bereich CloudFront Viewers unter Start Date (Startdatum) und End Date (Enddatum) den Datumsbereich aus, für den Sie die Betrachterdiagramme und -berichte anzeigen möchten.

   Für das Diagramm zu den Standorten sind die verfügbaren Datumsbereiche von dem Wert abhängig, den Sie unter Granularity (Granularität) auswählen:

   - Daily (Täglich) – Um Diagramme mit einem einzigen Datenpunkt pro Tag anzuzeigen, wählen Sie einen Datumsbereich in den letzten 60 Tagen aus.
   - Hourly (Stündlich) – Um Diagramme mit einem einzigen Datenpunkt pro Stunde anzeigen, wählen Sie einen Datumsbereich von bis zu 14 Tagen in den letzten 60 Tagen aus.

   Datum und Uhrzeit entsprechen der Zeitzone UTC (Coordinated Universal Time).
4. (Nur Diagramme zu Browsern und Betriebssystemen) Geben Sie unter Grouping (Gruppierung) an, ob Sie Browser und Betriebssysteme nach Namen (Chrome, Firefox) oder nach Namen und Version (Chrom 40.0, Firefox 35.0) gruppieren möchten.
5. (Nur Diagramme zu Standorten) Wählen Sie unter Granularity (Granularität) aus, ob Sie Diagramme mit einem Datenpunkt pro Tag oder einem Datenpunkt pro Stunde anzeigen möchten. Wenn Sie einen Datumsbereich größer als 14 Tage angeben, ist die Option "Ein Datenpunkt pro Stunde" nicht verfügbar.
6. (Nur Standort-Diagramm) Geben Sie für Details an, ob die Top-Standorte nach Ländern oder Bundesstaaten angezeigt werden U.S sollen.
7. Wählen Sie in der Liste Distribution die Verteilung aus, für die Sie in den Nutzungsdiagrammen Daten anzeigen möchten:

   - Eine individuelle Verteilung — In den Diagrammen werden Daten für die ausgewählte CloudFront Verteilung angezeigt. Unter Distribution werden die Verteilungs-ID und ein alternativer Domain-Namen (CNAME) für die Verteilung angezeigt, sofern vorhanden. Wenn eine Verteilung keine alternativen Domänennamen hat, enthält die Liste einen Ursprungsdomänennamen für die Verteilung.

- Alle Verteilungen (außer gelöschten) – Die Diagramme zeigen zusammengefasste Daten für alle Verteilungen an, die dem aktuellen AWS -Konto zugeordnet sind. Ausgenommen sind Verteilungen, die Sie gelöscht haben.

8. Wählen Sie Aktualisieren aus.

Sie können die Daten für einen täglichen oder stündlichen Datenpunkt in einem Diagramm anzeigen, indem Sie den Mauszeiger auf dem Datenpunkt platzieren.

Herunterladen von Daten im CSV-Format

Sie können alle Viewer-Berichte im CSV-Format herunterladen. In diesem Abschnitt wird erklärt, wie die Berichte heruntergeladen werden; außerdem werden die Werte in dem Bericht beschrieben.

So laden Sie den Viewer-Bericht im CSV-Format herunter

1. Wählen Sie während der Anzeige des Viewer-Berichts CSV aus.
2. Wählen Sie die Daten aus, die Sie herunterladen möchten, z. B. zu Devices (Geräten) oder Devices Trends (Gerätetendenzen).
3. Wählen Sie im Dialogfeld zum Öffnen der Datei, ob Sie die Datei öffnen oder speichern möchten.

In den Viewer-Berichten enthaltene Daten

Die ersten Zeilen eines Berichts enthalten jeweils die folgenden Informationen:

Version

Die Version des Formats für diese CSV-Datei

Bericht

Der Name des Berichts.

DistributionID

Die ID der Verteilung, für die der Bericht ausgeführt wurde, oder `ALL`, wenn der Bericht für alle Verteilungen ausgeführt wurde

StartDateUTC

Der Beginn des Datumsbereichs, für den Sie den Bericht ausgeführt haben, in koordinierter Weltzeit (Coordinated Universal Time, UTC).

EndDateUTC

Das Ende des Datumsbereichs, für den Sie den Bericht ausgeführt haben, in koordinierter Weltzeit (Coordinated Universal Time, UTC).

GeneratedTimeUTC

Das Datum und die Uhrzeit, zu der Sie den Bericht ausgeführt haben, in koordinierter Weltzeit (Coordinated Universal Time, UTC).

Grouping (nur Berichte zu Browsern und Betriebssystemen)

Ob die Daten nach dem Namen oder nach dem Namen und der Version des Browsers oder des Betriebssystems gruppiert werden

Granularity

Ob die einzelnen Zeilen in dem Bericht für eine Stunde oder einen Tag stehen

Details (nur Berichte zu Standorten)

Ob Anfragen nach Land oder U.S Bundesland aufgelistet sind.

In den folgenden Themen werden die Informationen in den verschiedenen Viewer-Berichten beschrieben.

Themen

- Devices-Bericht
- Device Trends-Bericht
- Browsers-Bericht
- Browser Trends-Bericht
- Operating Systems-Bericht
- Operating Systems Trends-Bericht
- Locations-Bericht
- Location Trends-Bericht

Devices-Bericht

Der Bericht enthält die folgenden Werte:

DistributionID

Die ID der Verteilung, für die der Bericht ausgeführt wurde, oder `ALL`, wenn der Bericht für alle Verteilungen ausgeführt wurde

FriendlyName

Ein alternativer Domänennamen (CNAMEs) für die Verteilung (sofern vorhanden). Wenn eine Verteilung keine alternativen Domänennamen hat, enthält die Liste einen Ursprungsdomänennamen für die Verteilung.

Anforderungen

Die Anzahl der Anfragen, die von jedem Gerätetyp CloudFront eingegangen sind.

RequestsPct

Die Anzahl der Anfragen, die von jedem Gerätetyp CloudFront empfangen wurden, als Prozentsatz der Gesamtzahl der Anfragen, die von allen Geräten CloudFront empfangen wurden.

Benutzerdefiniert

Anfragen, für die der Wert des HTTP-`User-Agent`-Headers nicht einem der Standard-Gerätetypen wie `Desktop` oder `Mobile` zugeordnet werden konnte

Device Trends-Bericht

Der Bericht enthält die folgenden Werte:

DistributionID

Die ID der Verteilung, für die der Bericht ausgeführt wurde, oder `ALL`, wenn der Bericht für alle Verteilungen ausgeführt wurde

FriendlyName

Ein alternativer Domänennamen (CNAMEs) für die Verteilung (sofern vorhanden). Wenn eine Verteilung keine alternativen Domänennamen hat, enthält die Liste einen Ursprungsdomänennamen für die Verteilung.

TimeBucket

Die Uhrzeit oder das Datum für die Nutzung in koordinierter Weltzeit (Coordinated Universal Time, UTC).

Desktop

Die Anzahl der Anfragen, die während des Zeitraums von Desktop-Computern CloudFront eingegangen sind.

Mobil

Die Anzahl der Anfragen, die während des Zeitraums von Mobilgeräten CloudFront eingegangen sind. Mobile Geräte können sowohl Tablets als auch Smartphones sein. Wenn nicht festgestellt CloudFront werden kann, ob eine Anfrage von einem Mobilgerät oder einem Tablet stammt, wird sie in der `Mobile` Spalte gezählt.

Smart-TV

Die Anzahl der Anfragen, die während des Zeitraums von Smart-TVs CloudFront eingegangen sind.

Tablet

Die Anzahl der Anfragen, die während des Zeitraums von Tablets CloudFront eingegangen sind. Wenn nicht festgestellt CloudFront werden kann, ob eine Anfrage von einem Mobilgerät oder einem Tablet stammt, wird sie in der `Mobile` Spalte gezählt.

Unbekannt

Anfragen, für die der Wert des HTTP-`User-Agent`-Headers nicht einem der Standard-Gerätetypen wie `Desktop` oder `Mobile` zugeordnet werden konnte

Leer

Die Anzahl der CloudFront eingegangenen Anfragen, die während des Zeitraums keinen Wert im `User-Agent` HTTP-Header enthielten.

Browsers-Bericht

Der Bericht enthält die folgenden Werte:

DistributionID

Die ID der Verteilung, für die der Bericht ausgeführt wurde, oder `ALL`, wenn der Bericht für alle Verteilungen ausgeführt wurde

FriendlyName

Ein alternativer Domänennamen (CNAMEs) für die Verteilung (sofern vorhanden). Wenn eine Verteilung keine alternativen Domänennamen hat, enthält die Liste einen Ursprungsdomänennamen für die Verteilung.

Group (Gruppieren)

Der Browser oder der Browser und die Version, von der Anfragen CloudFront empfangen wurden, abhängig vom Wert von`Grouping`. Neben den Browsernamen sind die folgenden Werte möglich:

- Bot/Crawler— hauptsächlich Anfragen von Suchmaschinen, die Ihre Inhalte indexieren.
- Empty (Leer) – Anforderungen, bei denen der Wert des HTTP-Headers `User-Agent` leer war.
- Andere — Browser, die zwar CloudFront identifiziert wurden, aber nicht zu den beliebtesten gehören. Wenn `Bot/CrawlerEmpty`, and/or `Unknown` nicht unter den ersten neun Werten erscheinen, sind sie ebenfalls enthalten`Other`.
- Unknown (Unbekannt) – Anforderungen bei denen der Wert des HTTP-Headers `User-Agent` keinem Standard-Browser zugeordnet werden konnte. Die meisten Anfragen in dieser Kategorie stammen von benutzerdefinierten Anwendungen oder Skripts.

Anforderungen

Die Anzahl der Anfragen, die von jedem Browsertyp CloudFront empfangen wurden.

RequestsPct

Die Anzahl der Anfragen, die von jedem Browsertyp CloudFront empfangen wurden, als Prozentsatz der Gesamtzahl der Anfragen, die während des Zeitraums CloudFront eingegangen sind.

Browser Trends-Bericht

Der Bericht enthält die folgenden Werte:

DistributionID

Die ID der Verteilung, für die der Bericht ausgeführt wurde, oder `ALL`, wenn der Bericht für alle Verteilungen ausgeführt wurde

FriendlyName

Ein alternativer Domänennamen (CNAMEs) für die Verteilung (sofern vorhanden). Wenn eine Verteilung keine alternativen Domänennamen hat, enthält die Liste einen Ursprungsdomänennamen für die Verteilung.

TimeBucket

Die Uhrzeit oder das Datum für die Nutzung in koordinierter Weltzeit (Coordinated Universal Time, UTC).

(Browser)

Die restlichen Spalten in dem Bericht enthalten die Browser oder die Browser und deren Versionen, entsprechend dem Wert unter `Grouping`. Neben den Browsernamen sind die folgenden Werte möglich:

- Bot/Crawler— hauptsächlich Anfragen von Suchmaschinen, die Ihre Inhalte indexieren.
- Empty (Leer) – Anforderungen, bei denen der Wert des HTTP-Headers `User-Agent` leer war.
- Andere — Browser, die zwar CloudFront identifiziert wurden, aber nicht zu den beliebtesten gehören. Wenn `Bot/CrawlerEmpty`, and/or `Unknown` nicht unter den ersten neun Werten erscheinen, sind sie ebenfalls enthalten`Other`.
- Unknown (Unbekannt) – Anforderungen bei denen der Wert des HTTP-Headers `User-Agent` keinem Standard-Browser zugeordnet werden konnte. Die meisten Anfragen in dieser Kategorie stammen von benutzerdefinierten Anwendungen oder Skripts.

Operating Systems-Bericht

Der Bericht enthält die folgenden Werte:

DistributionID

Die ID der Verteilung, für die der Bericht ausgeführt wurde, oder `ALL`, wenn der Bericht für alle Verteilungen ausgeführt wurde

FriendlyName

Ein alternativer Domänennamen (CNAMEs) für die Verteilung (sofern vorhanden). Wenn eine Verteilung keine alternativen Domänennamen hat, enthält die Liste einen Ursprungsdomänennamen für die Verteilung.

Group (Gruppieren)

Das Betriebssystem oder das Betriebssystem und die Version, von dem CloudFront Anfragen erhalten hat, entsprechend dem Wert unter `Grouping`. Neben den Namen der Betriebssysteme sind die folgenden Werte möglich:

- Bot/Crawler— hauptsächlich Anfragen von Suchmaschinen, die Ihre Inhalte indexieren.
- Empty (Leer) – Anforderungen, bei denen der Wert des HTTP-Headers `User-Agent` leer war.
- Andere — Betriebssysteme, die zwar CloudFront identifiziert wurden, aber nicht zu den beliebtesten gehören. Wenn`Bot/Crawler`,`Empty`, and/or `Unknown` nicht unter den ersten neun Werten vorkommen, sind sie ebenfalls enthalten`Other`.
- Unknown (Unbekannt) – Anforderungen bei denen der Wert des HTTP-Headers `User-Agent` keinem Standard-Browser zugeordnet werden konnte. Die meisten Anfragen in dieser Kategorie stammen von benutzerdefinierten Anwendungen oder Skripts.

Anforderungen

Die Anzahl der Anfragen, die von jedem Betriebssystemtyp CloudFront eingegangen sind.

RequestsPct

Die Anzahl der Anfragen, die von jedem Betriebssystemtyp CloudFront eingegangen sind, als Prozentsatz der Gesamtzahl der Anfragen, die während des Zeitraums CloudFront eingegangen sind.

Operating Systems Trends-Bericht

Der Bericht enthält die folgenden Werte:

DistributionID

Die ID der Verteilung, für die der Bericht ausgeführt wurde, oder `ALL`, wenn der Bericht für alle Verteilungen ausgeführt wurde

FriendlyName

Ein alternativer Domänennamen (CNAMEs) für die Verteilung (sofern vorhanden).
Wenn eine Verteilung keine alternativen Domänennamen hat, enthält die Liste einen Ursprungsdomänennamen für die Verteilung.

TimeBucket

Die Uhrzeit oder das Datum für die Nutzung in koordinierter Weltzeit (Coordinated Universal Time, UTC).

(Betriebssysteme)

Die restlichen Spalten in dem Bericht enthalten die Betriebssysteme oder die Betriebssysteme und deren Versionen, entsprechend dem Wert unter `Grouping`. Neben den Namen der Betriebssysteme sind die folgenden Werte möglich:

- Bot/Crawler— hauptsächlich Anfragen von Suchmaschinen, die Ihre Inhalte indexieren.
- Empty (Leer) – Anforderungen, bei denen der Wert des HTTP-Headers `User-Agent` leer war.
- Andere — Betriebssysteme, die zwar CloudFront identifiziert wurden, aber nicht zu den beliebtesten gehören. Wenn`Bot/Crawler`,`Empty`, and/or `Unknown` nicht unter den ersten neun Werten vorkommen, sind sie ebenfalls enthalten`Other`.
- Unknown (Unbekannt) – Anforderungen, bei denen das Betriebssystem nicht im HTTP-Header `User-Agent` angegeben ist.

Locations-Bericht

Der Bericht enthält die folgenden Werte:

DistributionID

Die ID der Verteilung, für die der Bericht ausgeführt wurde, oder `ALL`, wenn der Bericht für alle Verteilungen ausgeführt wurde

FriendlyName

Ein alternativer Domänennamen (CNAMEs) für die Verteilung (sofern vorhanden).
Wenn eine Verteilung keine alternativen Domänennamen hat, enthält die Liste einen Ursprungsdomänennamen für die Verteilung.

LocationCode

Die Abkürzung für den Standort, von dem Anfragen CloudFront empfangen wurden. Weitere Informationen zu möglichen Werten finden Sie in der Beschreibung der Standorte unter Wie die Daten im Standortbericht mit den Daten in den CloudFront Standardprotokollen (Zugriffsprotokollen) zusammenhängen.

LocationName

Der Name des Standorts, von dem Anfragen CloudFront empfangen wurden.

Anforderungen

Die Anzahl der Anfragen, die von jedem Standort CloudFront empfangen wurden.

RequestsPct

Die Anzahl der Anfragen, die von jedem Standort CloudFront eingegangen sind, als Prozentsatz der Gesamtzahl der Anfragen, die während des Zeitraums von allen Standorten CloudFront eingegangen sind.

TotalBytes

Die Anzahl der Byte, die CloudFront Zuschauern in diesem Land oder Bundesstaat für die angegebene Verteilung und den angegebenen Zeitraum zur Verfügung gestellt wurden.

Location Trends-Bericht

Der Bericht enthält die folgenden Werte:

DistributionID

Die ID der Verteilung, für die der Bericht ausgeführt wurde, oder `ALL`, wenn der Bericht für alle Verteilungen ausgeführt wurde

FriendlyName

Ein alternativer Domänennamen (CNAMEs) für die Verteilung (sofern vorhanden).
Wenn eine Verteilung keine alternativen Domänennamen hat, enthält die Liste einen Ursprungsdomänennamen für die Verteilung.

TimeBucket

Die Uhrzeit oder das Datum für die Nutzung in koordinierter Weltzeit (Coordinated Universal Time, UTC).

(Standorte)

Die restlichen Spalten im Bericht enthalten die Standorte, die von denen CloudFront Anfragen erhalten hat. Weitere Informationen zu möglichen Werten finden Sie in der Beschreibung der Standorte unter Wie die Daten im Standortbericht mit den Daten in den CloudFront Standardprotokollen (Zugriffsprotokollen) zusammenhängen.

Wie die Daten im Standortbericht mit den Daten in den CloudFront Standardprotokollen (Zugriffsprotokollen) zusammenhängen

Die folgende Liste zeigt, wie die Daten im Standortbericht in der CloudFront Konsole den Werten in den CloudFront Zugriffsprotokollen entsprechen. Weitere Informationen zu CloudFront Zugriffsprotokollen finden Sie unterZugriffsprotokolle (Standardprotokolle).

Speicherort

Das Land oder U.S der Staat, in dem sich der Zuschauer befindet. Die Spalte `c-ip` in den Zugriffsprotokollen enthält die IP-Adresse des Geräts, auf dem der Betrachter ausgeführt wird. Wir verwenden Geolocation-Daten zur Bestimmung des geografischen Standorts auf Basis der IP-Adresse des Geräts.

Wenn Sie den Bericht Standorte nach Ländern anzeigen, basiert die Liste der Länder auf ISO 3166-2, Codes für die Darstellung von Ländernamen und ihrer Unterteilungen – Teil 2: Codes für die Unterteilung von Ländern. Die Liste der Länder enthält die folgenden zusätzlichen Werte:

- Anonymous Proxy (Anonymer Proxy) – Die Anforderung stammt von einem anonymen Proxy.
- Satellite Provider (Satellitenanbieter) – Die Anforderung stammt von einem Satellitenanbieter, der Internetverbindungen für mehrere Länder bereitstellt. Viewer befinden sich möglicherweise Ländern mit einem hohen Betrugsrisiko.
- Europe (Unknown) (Europa (Unbekannt) – Die Anforderung stammt von einer IP-Adresse in einem Block, der von mehreren europäischen Ländern verwendet wird. Das Land, aus dem die Anfrage stammt, kann nicht ermittelt werden. CloudFront verwendet Europa (Unbekannt) als Standard.
- Asia/Pacific (Unbekannt) — Die Anfrage ging von einer IP in einem Block aus, der von mehreren Ländern in der Asia/Pacific Region verwendet wird. Das Land, aus dem die Anfrage stammt, kann nicht ermittelt werden. CloudFront verwendet Asia/Pacific (Unbekannt) als Standard.

Wenn Sie den Bericht „Standorte“ nach U.S Bundesland anzeigen, beachten Sie, dass der Bericht auch Gebiete und enthalten U.S kann U.S. Regionen für die US-Streitkräfte.

> **Note**
>
> Wenn der Standort eines Benutzers nicht ermittelt werden CloudFront kann, wird der Standort in Zuschauerberichten als Unbekannt angezeigt.

Anzahl der Anfragen

Die Gesamtzahl der Anfragen aus dem Land oder U.S dem Staat, in dem sich der Zuschauer aufhält, für den angegebenen Versand und Zeitraum. Dieser Wert entspricht im Allgemeinen weitgehend der Anzahl der GET Anfragen von IP-Adressen in diesem Land oder Bundesstaat in CloudFront den Zugriffsprotokollen.

Anfrage %

Einer der folgenden Werte, entsprechend dem Wert unter Details:

- Countries (Länder) – Die Anforderungen aus diesem Land als Prozentsatz der Gesamtzahl der Anforderungen.
- U.S. Bundesstaaten — Die Anfragen aus diesem Bundesstaat als Prozentsatz der Gesamtzahl der Anfragen aus den Vereinigten Staaten.

Wenn Anfragen aus mehr als 50 Ländern aufgetreten sind, können Sie Request % (Anfrage %) nicht basierend auf den Daten in dieser Tabelle berechnen, da in der Spalte Request Count (Anzahl der Anfragen)nicht alle Anfragen für den angegebenen Zeitraum enthalten sind.

Bytes

Die Anzahl der Byte, die CloudFront Zuschauern in diesem Land oder Bundesstaat für die angegebene Verteilung und den angegebenen Zeitraum zugestellt wurden. Sie können die Anzeige der Daten in dieser Spalte in KB, MB oder GB ändern, indem Sie den Link in der Spaltenüberschrift auswählen.

Überwachen Sie CloudFront Metriken mit Amazon CloudWatch

Amazon CloudFront ist in Amazon integriert CloudWatch und veröffentlicht automatisch Betriebsmetriken für Distributionen und Edge-Funktionen (sowohl Lambda @Edge als auch CloudFront Functions). Sie können diese Metriken zur Fehlersuche sowie zum Nachverfolgen und Beheben von Problemen verwenden. Viele dieser Metriken werden in einer Reihe von Diagrammen in der CloudFront Konsole angezeigt und sind auch über die CloudFront API oder CLI zugänglich. Alle diese Metriken sind in der CloudWatch Konsole oder über die CloudWatch API oder CLI verfügbar. CloudFront Metriken werden nicht auf CloudWatch Kontingente (früher als Limits bezeichnet) angerechnet und es fallen keine zusätzlichen Kosten an.

Zusätzlich zu den Standardmetriken für CloudFront Verteilungen können Sie gegen zusätzliche Kosten zusätzliche Metriken aktivieren. Die zusätzlichen Metriken gelten für CloudFront Verteilungen

und müssen für jede Verteilung separat aktiviert werden. Weitere Informationen zu den Kosten finden Sie unter the section called “Schätzen Sie die Kosten für die zusätzlichen CloudFront Metriken”.

Sie können auch Alarme auf der Grundlage dieser Metriken in der CloudFront Konsole oder in der CloudWatch Konsole, API oder CLI einrichten. Sie können beispielsweise einen Alarm basierend auf der Metrik `5xxErrorRate` festlegen, die den Prozentsatz aller Viewer-Anforderungen darstellt, für die sich der HTTP-Statuscode der Antwort im Bereich `500` bis einschließlich `599` befindet. Wenn die Fehlerquote einen bestimmten Wert für eine bestimmte Zeit erreicht (z. B. 5 % der Anforderungen für 5 kontinuierliche Minuten), wird der Alarm ausgelöst. Sie geben den Wert des Alarms und seine Zeiteinheit an, wenn Sie den Alarm erstellen.

Hinweise

- Wenn Sie in der CloudFront Konsole einen CloudWatch Alarm erstellen, wird in der Region USA Ost (Nord-Virginia) ein Alarm für Sie erstellt (`us-east-1`). Wenn Sie von der CloudWatch Konsole aus einen Alarm erstellen, müssen Sie dieselbe Region verwenden. Da CloudFront es sich um einen globalen Service handelt, werden die Messdaten für den Service nach USA Ost (Nord-Virginia) gesendet.
- Bei der Erstellung von Alarmen gelten die CloudWatch Standardpreise.

Themen

- Funktionsmetriken anzeigen CloudFront und erweitern
- Erstellen von -Alarmen für -Metriken
- Herunterladen von Metrikdaten im CSV-Format
- Arten von Metriken für CloudFront

Funktionsmetriken anzeigen CloudFront und erweitern

In der Konsole können Sie Betriebsmetriken zu Ihren CloudFront Distributionen und Edge-Funktionen einsehen. CloudFront

So können Sie Funktionsmetriken in anzeigen CloudFront und erweitern CloudFront

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home.
2. Wählen Sie im Navigationsbereich Monitoring (Überwachung) aus.
3. Um Diagramme zur Aktivität für eine bestimmte CloudFront Verteilungs- oder Edge-Funktion anzuzeigen, wählen Sie eine aus und wählen Sie dann Verteilungsmetriken anzeigen oder Metriken anzeigen aus.
4. Sie können die Diagramme anpassen, indem Sie die folgenden Schritte ausführen:

    a. Zum Ändern des Zeitraums für die im Diagramm angezeigten Informationen wählen Sie 1h (1 Stunde), 3 h (3 Stunden) oder einen anderen Zeitraum aus. Alternativ können Sie auch einen benutzerdefinierten Bereich angeben.

    b. Um zu ändern, wie oft die Informationen im Diagramm CloudFront aktualisiert werden, klicken Sie auf den Abwärtspfeil neben dem Aktualisierungssymbol und wählen Sie dann eine Aktualisierungsrate aus. Die standardmäßige Aktualisierungsrate beträgt 1 Minute. Sie können jedoch auch andere Optionen auswählen.
5. Um CloudFront Diagramme in der CloudWatch Konsole anzuzeigen, wählen Sie Zum Dashboard hinzufügen. Sie müssen die Region USA Ost (Nord-Virginia) verwenden, um die Grafiken in der CloudWatch Konsole anzeigen zu können.

Themen

- Standardmetriken für den CloudFront Vertrieb
- Schalten Sie zusätzliche CloudFront Verteilungsmetriken ein
- Standardmetriken für Lambda@Edge-Funktionen
- Metriken für CloudFront Standardfunktionen

Standardmetriken für den CloudFront Vertrieb

Die folgenden Standardmetriken sind für alle CloudFront Verteilungen ohne zusätzliche Kosten enthalten:

Anforderungen

Die Gesamtzahl der Viewer-Anfragen CloudFront, die von allen HTTP-Methoden und sowohl für HTTP- als auch für HTTPS-Anfragen empfangen wurden.

Heruntergeladene Bytes

Die Gesamtzahl der von Viewern für `GET`- und `HEAD`-Anforderungen heruntergeladenen Bytes.

Hochgeladene Bytes

Die Gesamtzahl der Byte, die Zuschauer hochgeladen CloudFront`OPTIONS`, verwendet `POST` und `PUT` angefordert haben.

4xx-Fehlerrate

Der Prozentsatz aller Viewer-Anforderungen, für die der HTTP-Statuscode der Antwort `4xx` lautet.

5xx-Fehlerrate

Der Prozentsatz aller Viewer-Anforderungen, für die der HTTP-Statuscode der Antwort `5xx` lautet.

Gesamte Fehlerrate

Der Prozentsatz aller Viewer-Anforderungen, für die der HTTP-Statuscode der Antwort `4xx` oder `5xx` lautet.

Diese Metriken werden in Diagrammen für jede CloudFront Verteilung auf der Monitoring-Seite der CloudFront Konsole angezeigt. In jedem Diagramm wird die Gesamtzahl in Zeiteinheiten von 1 Minute angezeigt. Zusätzlich zum Anzeigen der Diagramme können Sie Metrikberichte auch als CSV-Dateien herunterladen.

Schalten Sie zusätzliche CloudFront Verteilungsmetriken ein

Zusätzlich zu den Standardmetriken können Sie für eine Zusatzgebühr weitere Metriken aktivieren. Weitere Informationen zu den Kosten finden Sie unter the section called “Schätzen Sie die Kosten für die zusätzlichen CloudFront Metriken”.

Diese zusätzlichen Metriken müssen für jede Verteilung separat aktiviert werden:

Cache-Trefferrate

Der Prozentsatz aller zwischenspeicherbaren Anfragen, für die der Inhalt aus dem Cache CloudFront bereitgestellt wurde. HTTP `POST`- und `PUT`-Anforderungen und Fehler werden nicht als cachebare Anforderungen betrachtet.

Ursprungslatenz

Die Gesamtzeit zwischen dem CloudFront Empfang einer Anfrage und dem Beginn der Antwort an das Netzwerk (nicht an den Betrachter) für Anfragen, die vom Ursprung und nicht

vom CloudFront Cache aus bedient werden. Dies wird auch als Latenz des ersten Byte oder bezeichnet time-to-first-byte.

Fehlerquote nach Statuscode

Der Prozentsatz aller Viewer-Anforderungen, für die der HTTP-Statuscode der Antwort ein bestimmter Code im Bereich `4xx` oder `5xx` ist. Diese Metrik ist für alle folgenden Fehlercodes verfügbar: `401`, `403`, `404`, `502`, `503` und `504`.

Sie können zusätzliche Metriken in der CloudFront Konsole, mit CloudFormation, mit der AWS Command Line Interface (AWS CLI) oder mit der CloudFront API aktivieren.

Console

So aktivieren Sie zusätzliche Metriken

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home.
2. Wählen Sie im Navigationsbereich Monitoring (Überwachung) aus.
3. Wählen Sie die Verteilung aus, für die zusätzliche Metriken aktiviert werden sollen, und wählen Sie View distribution metrics (Verteilungsmetriken anzeigen) aus.
4. Wählen Sie Manage additional metrics (Zusätzliche Metriken verwalten) aus.
5. Aktivieren Sie im Fenster Manage additional metrics (Zusätzliche Metriken verwalten) die Option Enabled (Aktiviert). Nachdem Sie die zusätzlichen Metriken aktiviert haben, können Sie das Fenster Manage additional metrics (Zusätzliche Metriken verwalten) schließen.

   Nachdem Sie die zusätzlichen Metriken aktiviert haben, werden sie in Diagrammen angezeigt. In jedem Diagramm wird die Gesamtzahl in Zeiteinheiten von 1 Minute angezeigt. Zusätzlich zum Anzeigen der Diagramme können Sie Metrikberichte auch als CSV-Dateien herunterladen.

CloudFormation

Verwenden Sie den `AWS::CloudFront::MonitoringSubscription` Ressourcentyp CloudFormation, um zusätzliche Messwerte für zu aktivieren. Das folgende Beispiel zeigt die CloudFormation Vorlagensyntax im YAML-Format zur Aktivierung zusätzlicher Metriken.

```
Type: AWS::CloudFront::MonitoringSubscription
```

```
Properties:
  DistributionId: EDFDVBD6EXAMPLE
  MonitoringSubscription:
    RealtimeMetricsSubscriptionConfig:
      RealtimeMetricsSubscriptionStatus: Enabled
```

CLI

Verwenden Sie einen der folgenden Befehle, um zusätzliche Metriken mit dem AWS Command Line Interface (AWS CLI) zu verwalten:

So aktivieren Sie zusätzliche Metriken für eine Distribution

- Verwenden Sie den Befehl create-monitoring-subscription wie im folgenden Beispiel. *EDFDVBD6EXAMPLE*Ersetzen Sie es durch die ID der Distribution, für die Sie zusätzliche Metriken aktivieren.

```
aws cloudfront create-monitoring-subscription --
distribution-id EDFDVBD6EXAMPLE --monitoring-subscription
 RealtimeMetricsSubscriptionConfig={RealtimeMetricsSubscriptionStatus=Enabled}
```

So prüfen Sie, ob zusätzliche Metriken für eine Distribution aktiviert sind

- Verwenden Sie den Befehl get-monitoring-subscription wie im folgenden Beispiel. *EDFDVBD6EXAMPLE*Ersetzen Sie es durch die ID der Verteilung, die Sie überprüfen.

```
aws cloudfront get-monitoring-subscription --distribution-id EDFDVBD6EXAMPLE
```

So deaktivieren Sie zusätzliche Metriken für eine Distribution

- Verwenden Sie den Befehl delete-monitoring-subscription wie im folgenden Beispiel. *EDFDVBD6EXAMPLE*Ersetzen Sie es durch die ID der Verteilung, für die Sie zusätzliche Metriken deaktivieren möchten.

```
aws cloudfront delete-monitoring-subscription --distribution-id EDFDVBD6EXAMPLE
```

API

Verwenden Sie eine der folgenden CloudFront API-Operationen, um zusätzliche Metriken mit der API zu verwalten.

- Um zusätzliche Metriken für eine Distribution zu aktivieren, verwenden Sie CreateMonitoringSubscription.
- Um zu sehen, ob zusätzliche Metriken für eine Verteilung aktiviert sind, verwenden Sie GetMonitoringSubscription.
- Um zusätzliche Metriken für eine Verteilung zu deaktivieren, verwenden Sie DeleteMonitoringSubscription.

Weitere Informationen zu diesen API-Vorgängen finden Sie in der API-Referenzdokumentation für Ihr AWS SDK oder einen anderen API-Client.

Schätzen Sie die Kosten für die zusätzlichen CloudFront Metriken

Wenn Sie zusätzliche Metriken für eine Verteilung aktivieren, werden bis zu 8 Messwerte CloudWatch in die Region USA Ost (Nord-Virginia) CloudFront gesendet. CloudWatch berechnet für jede Metrik einen niedrigen, festen Tarif. Diese Rate wird nur einmal pro Monat pro Metrik berechnet (bis zu 8 Metriken pro Distribution). Da es sich um einen festen Tarif handelt, bleiben Ihre Kosten gleich, unabhängig von der Anzahl der Anfragen oder Antworten, die die CloudFront Distribution empfängt oder sendet. Die Rate pro Metrik finden Sie auf der CloudWatch Amazon-Preisseite und im CloudWatchPreisrechner. Zusätzliche API-Gebühren fallen an, wenn Sie die Metriken mit der CloudWatch API abrufen.

Standardmetriken für Lambda@Edge-Funktionen

Sie können CloudWatch Metriken verwenden, um Probleme mit Ihren Lambda @Edge -Funktionen in Echtzeit zu überwachen. Für diese Metriken fallen keine zusätzlichen Gebühren an.

Wenn Sie eine Lambda @Edge -Funktion an ein Cache-Verhalten in einer CloudFront Distribution anhängen, beginnt Lambda, automatisch Metriken an zu CloudWatch senden. Metriken sind für alle Lambda-Regionen verfügbar, aber um Metriken in der CloudWatch Konsole anzuzeigen oder die Metrikdaten von der CloudWatch API abzurufen, müssen Sie die Region USA Ost (Nord-Virginia) verwenden (`us-east-1`). Der Name der Metrikgruppe ist wie folgt formatiert:`AWS/CloudFront/`*`distribution-ID`*, wobei *`distribution-ID`* es sich um die ID der CloudFront

Distribution handelt, der die Lambda @Edge -Funktion zugeordnet ist. Weitere Informationen zu CloudWatch Metriken finden Sie im CloudWatch Amazon-Benutzerhandbuch.

Die folgenden Standardmetriken werden in Diagrammen für jede Lambda @Edge -Funktion auf der Monitoring-Seite der CloudFront Konsole angezeigt:

- `5xx`-Fehlerrate für Lambda@Edge
- Lambda-Ausführungsfehler
- Ungültige Lambda-Antworten
- Lambda-Drosselungen

Die Diagramme umfassen die Anzahl der Aufrufe, Fehler, Drosselungen und so weiter. In jedem Diagramm wird die Gesamtzahl in Zeiteinheiten von 1 Minute, gruppiert nach AWS -Region, angezeigt.

Wenn Sie einen Anstieg der Fehler feststellen, die Sie untersuchen möchten, können Sie eine Funktion auswählen und dann die Protokolldateien nach AWS Regionen anzeigen, bis Sie festgestellt haben, welche Funktion die Probleme verursacht und in welcher AWS Region. Weitere Informationen zur Behebung von Lambda@Edge-Fehlern finden Sie unter:

- the section called “So bestimmten Sie den Typ des Fehlers”
- Vier Schritte zum Debuggen Ihrer Inhaltsbereitstellung am AWS

Metriken für CloudFront Standardfunktionen

CloudFront Functions sendet Betriebsmetriken an Amazon, CloudWatch damit Sie Ihre Funktionen überwachen können. Das Anzeigen dieser Metriken kann Ihnen helfen, Probleme zu beheben, zu verfolgen und zu debuggen. CloudFront Functions veröffentlicht die folgenden Kennzahlen an CloudWatch:

- Invocations (`FunctionInvocations`) – Die Häufigkeit, mit der die Funktion in einem bestimmten Zeitraum gestartet (aufgerufen) wurde.
- Validierungsfehler (`FunctionValidationErrors`) – Die Anzahl der Validierungsfehler, die von der Funktion in einem bestimmten Zeitraum erzeugt werden. Validierungsfehler treten auf, wenn die Funktion erfolgreich ausgeführt wird, aber ungültige Daten (ein ungültiges Ereignisobjekt) zurückgibt.

- Ausführungsfehler (`FunctionExecutionErrors`) – Die Anzahl der Ausführungsfehler, die in einem bestimmten Zeitraum aufgetreten sind. Ausführungsfehler treten auf, wenn die Funktion nicht erfolgreich abgeschlossen werden kann.
- Auslastung berechnen (`FunctionComputeUtilization`) – Die Zeit, die die Ausführung der Funktion in Anspruch nahm, als Prozentsatz der maximal zulässigen Zeit. Zum Beispiel bedeutet ein Wert von 35, dass die Funktion in 35 % der maximal zulässigen Zeit abgeschlossen wurde. Diese Metrik ist eine Zahl zwischen 0 und 100.

  Wenn dieser Wert 100 erreicht oder sich diesem Wert annähert, hat die Funktion die zulässige Ausführungszeit vollständig oder fast aufgebraucht und nachfolgende Anforderungen werden möglicherweise gedrosselt. Wenn Ihre Funktion bei einer Auslastung von 80 % oder mehr ausgeführt wird, empfehlen wir Ihnen, Ihre Funktion zu überprüfen, um die Ausführungszeit zu verkürzen und die Auslastung zu verbessern. Beispielsweise können Sie nur Fehler protokollieren, komplexe Regex-Ausdrücke vereinfachen oder unnötiges Parsen komplexer JSON-Objekte vermeiden.
- Drosselungen (`FunctionThrottles`) – Die Häufigkeit, mit der die Funktion in einem bestimmten Zeitraum gedrosselt wurde. Funktionen können aus folgenden Gründen gedrosselt werden:
  - Die Funktion überschreitet kontinuierlich die maximal zulässige Ausführungszeit.
  - Die Funktion führt zu Kompilierungsfehlern.
  - Es gibt eine ungewöhnlich hohe Anzahl von Anforderungen pro Sekunde.

CloudFront KeyValueStore sendet außerdem die folgenden Betriebskennzahlen an Amazon CloudWatch:

- Leseanforderungen (`KvsReadRequests`) – wie oft die Funktion innerhalb eines bestimmten Zeitraums Daten aus dem Schlüsselwertspeicher erfolgreich gelesen hat.
- Lesefehler (`KvsReadErrors`) – wie oft die Funktion innerhalb eines bestimmten Zeitraums Daten aus dem Schlüsselwertspeicher nicht lesen konnte.

Alle diese Kennzahlen werden CloudWatch in der Region USA Ost (Nord-Virginia) (`us-east-1`) im CloudFront Namespace veröffentlicht. Sie können diese Metriken auch in der CloudWatch Konsole anzeigen. In der CloudWatch Konsole können Sie die Metriken pro Funktion oder pro Funktion pro Verteilung anzeigen.

Sie können damit auch Alarme einrichten CloudWatch , die auf diesen Metriken basieren.
Sie können beispielsweise einen Alarm basierend auf der Ausführungszeitmetrik

(`FunctionComputeUtilization`) festlegen, die den Prozentsatz der verfügbaren Zeit darstellt, die Ihre Funktion zur Ausführung benötigte. Wenn die Ausführungszeit für eine bestimmte Zeit einen bestimmten Wert erreicht. Wenn Sie beispielsweise mehr als 70 % der verfügbaren Zeit für 15 Minuten wählen, wird der Alarm ausgelöst. Sie geben den Wert des Alarms und seine Zeiteinheit an, wenn Sie den Alarm erstellen.

> **Note**
>
> CloudFront Functions sendet Metriken CloudWatch nur an Funktionen in der `LIVE` Phase, die als Reaktion auf Produktionsanfragen und -antworten ausgeführt werden. Wenn Sie eine Funktion testen, sendet CloudFront keine Metriken an CloudWatch. Die Testausgabe enthält Informationen über Fehler, Rechenauslastung und Funktionsprotokolle (`console.log()`Anweisungen), aber diese Informationen werden nicht an gesendet CloudWatch.

Informationen darüber, wie Sie diese Metriken mit der CloudWatch API abrufen können, finden Sie unterthe section called “CloudFront Metriken”.

Erstellen von -Alarmen für -Metriken

In der CloudFront Konsole können Sie Alarme so einrichten, dass Sie von Amazon Simple Notification Service (Amazon SNS) auf der Grundlage bestimmter CloudFront Messwerte benachrichtigt werden.

So erstellen Sie Alarme für Metriken

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home.
2. Klicken Sie im Navigationsbereich auf Alarms (Alarme).
3. Wählen Sie Create Alarm (Alarm erstellen) aus.
4. Geben Sie unter Details Folgendes an:
   a. Name des Alarms – einen Namen für den Alarm
   b. Verteilung — Die CloudFront Distribution, für die Sie den Alarm erstellen.
5. Geben Sie unter Bedingung Folgendes an:
   a. Metrik – die Metrik, für die Sie den Alarm erstellen

b. „IF“ <condition>— Der Schwellenwert, ab dem ein Alarm ausgelöst und eine Benachrichtigung an das Amazon SNS SNS-Thema gesendet werden CloudWatch soll. Um beispielsweise eine Benachrichtigung zu erhalten, wenn die `5xx`-Fehlerquote 1 % überschreitet, geben Sie Folgendes an:

   5xx-Fehlerrate > 1

c. „FOR“ aufeinanderfolgende Perioden – den Zeitraum, in dem die Bedingung erfüllt werden muss, bevor ein Alarm ausgelöst wird. Bei der Auswahl eines Wertes sollten Sie ein angemessenes Gleichgewicht anstreben, um nicht bei temporären Problemen Alarm zu schlagen, aber bei anhaltenden oder realen Probleme.

d. (Optional) Benachrichtigen – das Amazon-SNS-Thema, an das eine Benachrichtigung gesendet werden soll, wenn diese Metrik einen Alarm auslöst

6. Wählen Sie Alarm erstellen aus.

> **Hinweise**
>
> - Verwenden Sie bei der Eingabe der Werte für die Bedingung ganze Zahlen ohne Satzzeichen. Wenn Sie beispielsweise Tausend angeben möchten, geben Sie ei **`1000`**.
> - Für die drei Fehlerquotenkategorien `4xx`, `5xx` und „Total (Insgesamt)“ der Wert, den Sie als Prozentsatz angeben.
> - Bei Anforderungen, heruntergeladenen Bytes und hochgeladenen Bytes handelt es sich bei dem von Ihnen angegebenen Wert um Einheiten. Zum Beispiel 1073742000 Bytes.

Weitere Informationen finden Sie unter Erstellen eines Amazon-SNS-Themas im Entwicklerhandbuch für Amazon Simple Notification Service.

Herunterladen von Metrikdaten im CSV-Format

Sie können die CloudWatch Metrikdaten für eine CloudFront Verteilung im CSV-Format herunterladen.

So laden Sie Metrikdaten im CSV-Format herunter

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home.

2. Wählen Sie im Navigationsbereich Monitoring (Überwachung) aus.
3. Wählen Sie die Distribution und dann Distributionsmetriken anzeigen aus.
4. Wählen Sie CSV herunterladen und dann den Zeitraum aus, wie z. B. Für den letzten Tag (Zeitraum von 1 Stunde).
5. Nachdem Ihre Datei heruntergeladen wurde, öffnen Sie sie, um die folgenden Informationen anzuzeigen.

Themen

- Informationen zu dem Bericht
- Daten in dem Bericht zu Metriken

Informationen zu dem Bericht

Die ersten Zeilen des Berichts enthalten die folgenden Informationen:

Version

Die CloudFront Berichtsversion.

Bericht

Der Name des Berichts.

DistributionID

Die ID der Verteilung, für die Sie den Bericht ausgeführt haben.

StartDateUTC

Der Beginn des Datumsbereichs, für den Sie den Bericht ausgeführt haben, in koordinierter Weltzeit (Coordinated Universal Time, UTC).

EndDateUTC

Das Ende des Datumsbereichs, für den Sie den Bericht ausgeführt haben, in koordinierter Weltzeit (Coordinated Universal Time, UTC).

GeneratedTimeUTC

Das Datum und die Uhrzeit, zu der Sie den Bericht ausgeführt haben, in koordinierter Weltzeit (Coordinated Universal Time, UTC).

Granularity

Der Zeitraum für jede Zeile im Bericht, z. B, `ONE_MINUTE`.

Daten in dem Bericht zu Metriken

Der Bericht enthält die folgenden Werte:

DistributionID

Die ID der Verteilung, für die Sie den Bericht ausgeführt haben.

FriendlyName

Ein alternativer Domänennamen (CNAMEs) für die Verteilung (sofern vorhanden). Wenn eine Verteilung keine alternativen Domänennamen hat, enthält die Liste einen Ursprungsdomänennamen für die Verteilung.

TimeBucket

Die Uhrzeit oder das Datum für die Nutzung in koordinierter Weltzeit (Coordinated Universal Time, UTC).

Anforderungen

Die Gesamtzahl der Anforderungen für alle HTTP-Statuscodes (z. B. `200` oder `404`) und alle Methoden (z. B. `GET`, `HEAD`, `POST` usw.) während des Zeitraums.

BytesDownloaded

Die Anzahl der Bytes, die während des Zeitraums für die angegebene Verteilung von Betrachtern heruntergeladen worden sind.

BytesUploaded

Die Anzahl der Bytes, die während des Zeitraums für die angegebene Verteilung von Viewern hochgeladen wurden.

TotalErrorRatePct

Anforderungen, für die während des Zeitraums der angegebenen Verteilung der HTTP-Statuscode ein `4xx`- oder `5xx`-Fehler war, in Prozent.

4 xxErrorRate Pkt

Anforderungen, für die während des Zeitraums der angegebenen Verteilung der HTTP-Statuscode ein `4xx`-Fehler war, in Prozent.

5 Pkt xxErrorRate

Anforderungen, für die während des Zeitraums der angegebenen Verteilung der HTTP-Statuscode ein `5xx`-Fehler war, in Prozent.

Wenn Sie zusätzliche Metriken für Ihre Verteilung aktiviert haben, enthält der Bericht auch die folgenden zusätzlichen Werte:

401 ErrorRatePct

Anforderungen, für die während des Zeitraums der angegebenen Verteilung der HTTP-Statuscode ein `401`-Fehler war, in Prozent.

403 ErrorRatePct

Anforderungen, für die während des Zeitraums der angegebenen Verteilung der HTTP-Statuscode ein `403`-Fehler war, in Prozent.

404 ErrorRatePct

Anforderungen, für die während des Zeitraums der angegebenen Verteilung der HTTP-Statuscode ein `404`-Fehler war, in Prozent.

502 ErrorRatePct

Anforderungen, für die während des Zeitraums der angegebenen Verteilung der HTTP-Statuscode ein `502`-Fehler war, in Prozent.

503 ErrorRatePct

Anforderungen, für die während des Zeitraums der angegebenen Verteilung der HTTP-Statuscode ein `503`-Fehler war, in Prozent.

504 ErrorRatePct

Anforderungen, für die während des Zeitraums der angegebenen Verteilung der HTTP-Statuscode ein `504`-Fehler war, in Prozent.

OriginLatency

Die Gesamtzeit in Millisekunden vom CloudFront Empfang einer Anfrage bis zum Beginn der Antwort an das Netzwerk (nicht an den Betrachter) für Anfragen, die vom Ursprung und nicht vom Cache aus bedient wurden. CloudFront Dies wird auch als Latenz des ersten Byte oder bezeichnet. time-to-first-byte

CacheHitRate

Der Prozentsatz aller zwischenspeicherbaren Anfragen, für die der Inhalt aus dem Cache CloudFront bereitgestellt wurde. HTTP `POST`- und `PUT`-Anforderungen und Fehler werden nicht als cachebare Anforderungen betrachtet.

Arten von Metriken für CloudFront

Sie können die CloudWatch API oder AWS Command Line Interface (AWS CLI) verwenden, um die CloudFront Metriken in Programmen oder Anwendungen abzurufen, die Sie erstellen. Sie können die Rohdaten verwenden, um eigene benutzerdefinierte Dashboards, eigene alarmierende Tools usw. zu erstellen.

Weitere Informationen finden Sie get-metric-datain der AWS CLI Befehlsreferenz oder unter GetMetricDataAPI-Operation in der Amazon CloudWatch API-Referenz.

Themen

- Werte für alle CloudFront Metriken
- Werte für CloudFront Vertriebsmetriken
- Werte für CloudFront Funktionsmetriken

> **Note**
>
> Um die CloudFront Metriken von der CloudWatch API abzurufen, müssen Sie die Region USA Ost (Nord-Virginia) (`us-east-1`) verwenden. Sie müssen auch bestimmte Werte und Typen für jede Metrik kennen.

Werte für alle CloudFront Metriken

Die folgenden Werte gelten für alle CloudFront Metriken:

`Namespace`

Der Wert für `Namespace` ist immer `AWS/CloudFront`.

Dimensionen

Jede CloudFront Metrik hat die folgenden Dimensionen:

`DistributionId`

Die ID der CloudFront Distribution, für die Sie Metriken abrufen möchten.

`FunctionName`

Der Name der Funktion (in CloudFront Funktionen), für die Sie Metriken abrufen möchten.

Diese Dimension gilt nur für Funktionen.

`Region`

Der Wert für `Region` ist immer`Global`, weil CloudFront es sich um einen globalen Dienst handelt.

Werte für CloudFront Vertriebsmetriken

Verwenden Sie Informationen aus der folgenden Liste, um Details zu bestimmten CloudFront Verteilungsmetriken aus der CloudWatch API abzurufen. Einige dieser Metriken sind nur verfügbar, wenn Sie zusätzliche Metriken für die Verteilung aktiviert haben.

> **Note**
>
> Für jede Metrik ist nur eine Statistik, `Average` oder `Sum`, anwendbar. Die folgende Liste gibt an, welche Statistik auf diese Metrik anwendbar ist.

4xx-Fehlerrate

Der Prozentsatz aller Viewer-Anforderungen, für die der HTTP-Statuscode der Antwort `4xx` lautet.

- Metrikname: `4xxErrorRate`
- Gültige Statistik: `Average`
- Einheit: `Percent`

401-Fehlerquote

Der Prozentsatz aller Viewer-Anforderungen, für die der HTTP-Statuscode der Antwort `401` lautet. Um diese Metrik abzurufen, müssen Sie zunächst zusätzliche Metriken aktivieren.

- Metrikname: `401ErrorRate`
- Gültige Statistik: `Average`

- Einheit: `Percent`

403-Fehlerquote

Der Prozentsatz aller Viewer-Anforderungen, für die der HTTP-Statuscode der Antwort `403` lautet. Um diese Metrik abzurufen, müssen Sie zunächst zusätzliche Metriken aktivieren.

- Metrikname: `403ErrorRate`
- Gültige Statistik: `Average`
- Einheit: `Percent`

404-Fehlerquote

Der Prozentsatz aller Viewer-Anforderungen, für die der HTTP-Statuscode der Antwort `404` lautet. Um diese Metrik abzurufen, müssen Sie zunächst zusätzliche Metriken aktivieren.

- Metrikname: `404ErrorRate`
- Gültige Statistik: `Average`
- Einheit: `Percent`

5xx-Fehlerrate

Der Prozentsatz aller Viewer-Anforderungen, für die der HTTP-Statuscode der Antwort `5xx` lautet.

- Metrikname: `5xxErrorRate`
- Gültige Statistik: `Average`
- Einheit: `Percent`

502-Fehlerquote

Der Prozentsatz aller Viewer-Anforderungen, für die der HTTP-Statuscode der Antwort `502` lautet. Um diese Metrik abzurufen, müssen Sie zunächst zusätzliche Metriken aktivieren.

- Metrikname: `502ErrorRate`
- Gültige Statistik: `Average`
- Einheit: `Percent`

503-Fehlerquote

Der Prozentsatz aller Viewer-Anforderungen, für die der HTTP-Statuscode der Antwort `503` lautet. Um diese Metrik abzurufen, müssen Sie zunächst zusätzliche Metriken aktivieren.

- Metrikname: `503ErrorRate`
- Gültige Statistik: `Average`

- Einheit: `Percent`

504-Fehlerquote

Der Prozentsatz aller Viewer-Anforderungen, für die der HTTP-Statuscode der Antwort `504` lautet. Um diese Metrik abzurufen, müssen Sie zunächst zusätzliche Metriken aktivieren.

- Metrikname: `504ErrorRate`
- Gültige Statistik: `Average`
- Einheit: `Percent`

Heruntergeladene Bytes

Die Gesamtzahl der von Viewern für `GET`- und `HEAD`-Anforderungen heruntergeladenen Bytes.

- Metrikname: `BytesDownloaded`
- Gültige Statistik: `Sum`
- Einheit: `None`

Hochgeladene Bytes

Die Gesamtzahl der Byte, die Zuschauer hochgeladen CloudFront`OPTIONS`, verwendet `POST` und `PUT` angefordert haben.

- Metrikname: `BytesUploaded`
- Gültige Statistik: `Sum`
- Einheit: `None`

Cache-Trefferrate

Der Prozentsatz aller zwischenspeicherbaren Anfragen, für die der Inhalt aus dem Cache CloudFront bereitgestellt wurde. HTTP `POST`- und `PUT`-Anforderungen und Fehler werden nicht als cachebare Anforderungen betrachtet. Um diese Metrik abzurufen, müssen Sie zunächst zusätzliche Metriken aktivieren.

- Metrikname: `CacheHitRate`
- Gültige Statistik: `Average`
- Einheit: `Percent`

Ursprungslatenz

Die Gesamtzeit in Millisekunden vom CloudFront Empfang einer Anfrage bis zum Beginn der Antwort an das Netzwerk (nicht an den Betrachter) für Anfragen, die vom Ursprung und nicht

vom Cache aus bedient werden. CloudFront Dies wird auch als Latenz des ersten Byte oder bezeichnet. time-to-first-byte Um diese Metrik abzurufen, müssen Sie zunächst zusätzliche Metriken aktivieren.

- Metrikname: `OriginLatency`
- Gültige Statistik: `Percentile`
- Einheit: `Milliseconds`

> **Note**
>
> Um eine `Percentile` Statistik von der CloudWatch API abzurufen, verwenden Sie den `ExtendedStatistics` Parameter not`Statistics`. Weitere Informationen finden Sie GetMetricStatisticsin der Amazon CloudWatch API-Referenz oder in der Referenzdokumentation für AWS SDKs.

Anforderungen

Die Gesamtzahl der von allen HTTP-Methoden und sowohl für HTTP- als auch für HTTPS-Anfragen empfangenen Zuschaueranfragen. CloudFront

- Metrikname: `Requests`
- Gültige Statistik: `Sum`
- Einheit: `None`

Gesamte Fehlerrate

Der Prozentsatz aller Viewer-Anforderungen, für die der HTTP-Statuscode der Antwort `4xx` oder `5xx` lautet.

- Metrikname: `TotalErrorRate`
- Gültige Statistik: `Average`
- Einheit: `Percent`

Werte für CloudFront Funktionsmetriken

Verwenden Sie Informationen aus der folgenden Liste, um Details zu bestimmten CloudFront Funktionsmetriken aus der CloudWatch API abzurufen.

> **Note**
>
> Für jede Metrik ist nur eine Statistik, `Average` oder `Sum`, anwendbar. Die folgende Liste gibt an, welche Statistik auf diese Metrik anwendbar ist.

Aufrufe

Die Häufigkeit, mit der die Funktion in einem bestimmten Zeitraum gestartet (aufgerufen) wurde.

- Metrikname: `FunctionInvocations`
- Gültige Statistik: `Sum`
- Einheit: `None`

Validierungsfehler

Die Anzahl der Validierungsfehler, die von der Funktion in einem bestimmten Zeitraum erzeugt werden. Validierungsfehler treten auf, wenn die Funktion erfolgreich ausgeführt wird, aber ungültige Daten (ein ungültiges Ereignisobjekt) zurückgibt.

- Metrikname: `FunctionValidationErrors`
- Gültige Statistik: `Sum`
- Einheit: `None`

Ausführungsfehler

Die Anzahl der Ausführungsfehler, die in einem bestimmten Zeitraum aufgetreten sind. Ausführungsfehler treten auf, wenn die Funktion nicht erfolgreich abgeschlossen werden kann.

- Metrikname: `FunctionExecutionErrors`
- Gültige Statistik: `Sum`
- Einheit: `None`

Computing-Auslastung

Die Zeit (0–100), die die Ausführung der Funktion in Anspruch genommen hat, als Prozentsatz der maximal zulässigen Zeit. Zum Beispiel bedeutet ein Wert von 35, dass die Funktion in 35 % der maximal zulässigen Zeit abgeschlossen wurde.

- Metrikname: `FunctionComputeUtilization`
- Gültige Statistik: `Average`
- Einheit: `Percent`

Drosselungen

Die Häufigkeit, mit der die Funktion in einem bestimmten Zeitraum gedrosselt wurde.

- Metrikname: `FunctionThrottles`
- Gültige Statistik: `Sum`
- Einheit: `None`

CloudFront und Edge-Funktionsprotokollierung

Amazon CloudFront bietet verschiedene Arten der Protokollierung an. Sie können die Zuschaueranfragen protokollieren, die zu Ihren CloudFront Distributionen kommen, oder Sie können die CloudFront Serviceaktivität (API-Aktivität) in Ihrem AWS Konto protokollieren. Sie können auch Protokolle von Ihren CloudFront Functions- und Lambda @Edge -Funktionen abrufen.

Protokollieren von Anfragen

CloudFront bietet die folgenden Möglichkeiten, die Anfragen zu protokollieren, die an Ihre Distributionen gesendet werden.

Zugriffsprotokolle (Standardprotokolle)

CloudFrontZugriffsprotokolle enthalten detaillierte Aufzeichnungen über jede Anfrage, die an eine Distribution gestellt wird. Sie können die Protokolle für Szenarien wie Sicherheits- und Zugriffsprüfungen verwenden.

CloudFrontZugriffsprotokolle werden an das von Ihnen angegebene Lieferziel übermittelt.

Verwenden Sie Zugriffsprotokolle, wenn Sie Folgendes benötigen:

- Historische Analyse und Berichterstattung
- Sicherheitsaudits und Compliance-Anforderungen
- Kostengünstige langfristige Aufbewahrung von Protokollen

Weitere Informationen finden Sie unter Zugriffsprotokolle (Standardprotokolle).

Echtzeitzugriffsprotokolle

CloudFrontZugriffsprotokolle in Echtzeit werden innerhalb von Sekunden nach Eingang der Anfragen übermittelt und enthalten Informationen über Anfragen an eine Distribution in Echtzeit.

Sie können die Stichprobenrate für Ihre Echtzeit-Zugriffsprotokolle wählen, d. h. den Prozentsatz der Anfragen, für die Sie Zugriffsprotokolleinträge in Echtzeit erhalten möchten. Sie können auch die spezifischen Felder auswählen, die Sie in den Protokolldatensätzen erhalten möchten. Zugriffsprotokolle in Echtzeit eignen sich ideal für die Live-Überwachung der Leistung der Inhaltsbereitstellung.

CloudFront Echtzeit-Zugriffsprotokolle werden an den Datenstream Ihrer Wahl in Amazon Kinesis Data Streams übermittelt. CloudFront Gebühren für Echtzeit-Zugriffsprotokolle, zusätzlich zu den Gebühren, die Ihnen für die Nutzung von Kinesis Data Streams entstehen.

Verwenden Sie Echtzeit-Zugriffsprotokolle, wenn Sie Folgendes benötigen:

- Überwachung und Warnmeldungen in Echtzeit
- Live-Dashboards und betriebliche Einblicke

Weitere Informationen finden Sie unter Verwenden Sie Zugriffsprotokolle in Echtzeit.

Verbindungsprotokolle.

Verbindungsprotokolle enthalten detaillierte Informationen über die Verbindung zwischen dem Server und dem Client für mTLS-fähige Distributionen. Verbindungsprotokolle bieten Einblick in die Zertifikatsinformationen des Clients, die Gründe für fehlgeschlagene mTLS-Authentifizierung und darüber, ob eine Verbindung zugelassen oder abgelehnt wurde.

Wie Zugriffsprotokolle (Standardprotokolle) werden Verbindungsprotokolle an das von Ihnen angegebene Zustellungsziel übermittelt.

> **Note**
>
> Um Verbindungsprotokolle zu aktivieren, müssen Sie zuerst mTLS für Ihre Distribution aktivieren.

Verwenden Sie Verbindungsprotokolle, wenn Sie Folgendes benötigen:

- Gründe für erfolgreiche oder erfolglose Verbindungen während des TLS-Handshakes
- Einblick in die Informationen zum Client-Zertifikat

Weitere Informationen finden Sie unter Beobachtbarkeit mithilfe von Verbindungsprotokollen.

Protokollieren von Edge-Funktionen

Sie können Amazon CloudWatch Logs verwenden, um Protokolle für Ihre Edge-Funktionen, sowohl Lambda @Edge als auch CloudFront Functions, abzurufen. Sie können über die CloudWatch Konsole oder die Logs-API auf die CloudWatch Protokolle zugreifen. Weitere Informationen finden Sie unter the section called “Protokolle für Edge-Funktionen”.

Protokollieren von Service-Aktivität

Sie können AWS CloudTrail sie verwenden, um die CloudFront Serviceaktivität (API-Aktivität) in Ihrem AWS Konto zu protokollieren. CloudTrail bietet eine Aufzeichnung der API-Aktionen, die von einem Benutzer, einer Rolle oder einem AWS Dienst in ausgeführt wurden CloudFront. Anhand der von gesammelten Informationen können Sie die API-Anfrage ermitteln CloudTrail, an die die Anfrage gestellt wurde CloudFront, die IP-Adresse, von der aus die Anfrage gestellt wurde, wer die Anfrage gestellt hat, wann sie gestellt wurde, und weitere Details.

Weitere Informationen finden Sie unter Protokollieren Amazon CloudFront Amazon-API-Aufrufen mit AWS CloudTrail.

Weitere Informationen zur Protokollierung finden Sie in den folgenden Themen:

Themen

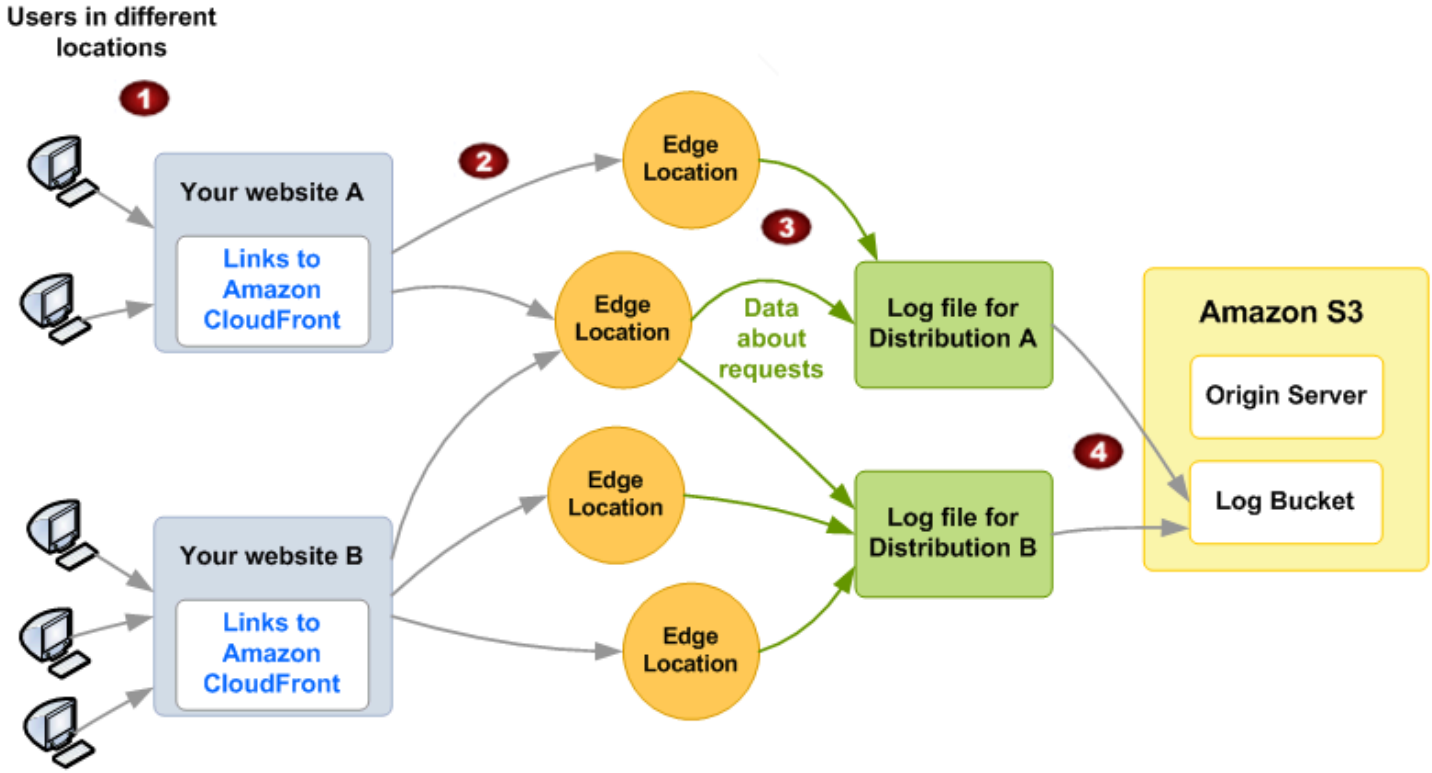
- Zugriffsprotokolle (Standardprotokolle)
- Verwenden Sie Zugriffsprotokolle in Echtzeit
- Protokolle für Edge-Funktionen
- Protokollieren Amazon CloudFront Amazon-API-Aufrufen mit AWS CloudTrail

Zugriffsprotokolle (Standardprotokolle)

Sie können so konfigurieren CloudFront , dass Protokolldateien erstellt werden, die detaillierte Informationen zu jeder CloudFront empfangenen Benutzeranfrage (Viewer) enthalten. Diese werden als Zugriffsprotokolle bezeichnet, die auch als Standardprotokolle bezeichnet werden.

Jedes Protokoll enthält Informationen wie die Zeit, zu der die Anforderung einging, Verarbeitungszeit, Anforderungspfade und Serverantworten. Sie können diese Zugriffsprotokolle verwenden, um Antwortzeiten zu analysieren und Probleme zu beheben.

Das folgende Diagramm zeigt, wie Informationen über Anfragen für Ihre Objekte CloudFront protokolliert werden. In diesem Beispiel sind die Distributionen so konfiguriert, dass sie Zugriffsprotokolle an einen Amazon S3-Bucket senden.



1. In diesem Beispiel haben Sie zwei Websites, A und B, und zwei entsprechende CloudFront Distributionen. Benutzer fragen Ihre Objekte mithilfe von URLs an, die Ihren Verteilungen zugewiesen sind.
2. CloudFront leitet jede Anforderung an den entsprechenden Edge-Standort weiter.
3. CloudFront schreibt Daten zu jeder Anforderung in eine Protokolldatei, die für diese Verteilung spezifisch ist. In diesem Beispiel werden Informationen zu Anforderungen mit Bezug auf Distribution A in einer Protokolldatei für Distribution A gespeichert. Informationen zu Anforderungen mit Bezug auf Distribution B werden in einer Protokolldatei für Distribution B gespeichert.
4. CloudFront speichert regelmäßig die Protokolldatei für eine Verteilung im Amazon S3-Bucket, den Sie bei der Aktivierung der Protokollierung angegeben haben. CloudFront beginnt dann, Informationen über nachfolgende Anfragen in einer neuen Protokolldatei für die Verteilung zu speichern.

   Wenn Viewer eine Stunde lang nicht auf Ihre Inhalte zugreifen, erhalten Sie für diesen Zeitraum auch keine Protokolldateien.

> **Note**
>
> Wir empfehlen, dass Sie die Protokolle verwenden, um die Art der Anfragen für Ihre Inhalte nachzuvollziehen, und nicht, um alle Anfragen vollständig aufzulisten. CloudFront stellt Zugriffsprotokolle nach bestem Bemühen bereit. Der Protokolleintrag für eine bestimmte Anfrage kann lange nach der tatsächlichen Bearbeitung der Anfrage zugestellt werden. In seltenen Fällen wird ein Protokolleintrag möglicherweise überhaupt nicht übermittelt. Wenn ein Protokolleintrag nicht in den Zugriffsprotokollen enthalten ist, stimmt die Anzahl der Einträge in den Zugriffsprotokollen nicht mit deren Anzahl in den Abrechnungs- und Nutzungsberichten für AWS überein.

CloudFront unterstützt zwei Versionen der Standardprotokollierung. Die Standardprotokollierung (Legacy) unterstützt nur das Senden Ihrer Zugriffsprotokolle an Amazon S3. Die Standardprotokollierung (v2) unterstützt zusätzliche Bereitstellungsziele. Sie können beide oder eine der beiden Protokollierungsoptionen für Ihre Distribution konfigurieren. Weitere Informationen finden Sie unter den folgenden Themen:

Themen

- Konfigurieren der Standardprotokollierung (v2)
- Konfigurieren der Standardprotokollierung (Legacy)
- Referenz zur Standardprotokollierung

> **Tip**
>
> CloudFront bietet auch Zugriffsprotokolle in Echtzeit, die Ihnen in Echtzeit Informationen über Anfragen an eine Distribution geben (Protokolle werden innerhalb von Sekunden nach Erhalt der Anfragen zugestellt). Sie können Zugriffsprotokolle in Echtzeit verwenden, um die Leistung der Inhaltsbereitstellung zu überwachen, zu analysieren und Maßnahmen zu ergreifen. Weitere Informationen finden Sie unter Verwenden Sie Zugriffsprotokolle in Echtzeit.

Konfigurieren der Standardprotokollierung (v2)

Sie können Zugriffsprotokolle (Standardprotokolle) aktivieren, wenn Sie eine Distribution erstellen oder aktualisieren. Die Standardprotokollierung (v2) umfasst die folgenden Funktionen:

- Senden Sie Zugriffsprotokolle an Amazon CloudWatch Logs, Amazon Data Firehose und Amazon Simple Storage Service (Amazon S3).
- Auswahl der gewünschten Protokollfelder. Sie können auch eine Teilmenge von Echtzeitzugriffsprotokollfeldern auswählen.
- Auswahl zusätzlicher Ausgabe-Protokolldateiformate

Wenn Sie Amazon S3 verwenden, verfügen Sie über die folgenden optionalen Funktionen:

- Senden Sie Protokolle, um sich anzumelden. AWS-Regionen
- Organisieren Ihrer Protokolle anhand von Partitionierung
- Aktivieren Sie Hive-compatible Dateinamen.

Weitere Informationen finden Sie unter Senden von Protokollen an Amazon S3.

Führen Sie für den Einstieg in die Standardprotokollierung die folgenden Schritte aus:

1. Richten Sie Ihre erforderlichen Berechtigungen für die angegebenen Personen ein AWS-Service , die Ihre Protokolle erhalten sollen.
2. Konfigurieren Sie die Standardprotokollierung CloudFront über die Konsole oder die CloudWatch API.
3. Zeigen Sie Ihre Zugriffsprotokolle an.

> **Note**
>
> - Wenn Sie die Standardprotokollierung (v2) aktivieren, hat dies keine Auswirkungen auf die Standardprotokollierung (Legacy) und verändert sie nicht. Sie können zusätzlich zur Standardprotokollierung (v2) weiterhin die Standardprotokollierung (Legacy) für Ihre Distribution verwenden. Weitere Informationen finden Sie unter Konfigurieren der Standardprotokollierung (Legacy).

- Wenn Sie die Standardprotokollierung (Legacy) bereits aktiviert haben und die Standardprotokollierung (v2) für Amazon S3 aktivieren möchten, empfehlen wir Ihnen, einen anderen Amazon-S3-Bucket anzugeben oder einen separaten Pfad in demselben Bucket zu verwenden (z. B. ein Protokollpräfix oder eine Partitionierung zu verwenden). Auf diese Weise behalten Sie den Überblick darüber, welche Protokolldateien mit welcher Distribution verknüpft sind, und es wird verhindert, dass sich Protokolldateien gegenseitig überschreiben.

Berechtigungen

CloudFront verwendet CloudWatch bereitgestellte Protokolle zur Bereitstellung von Zugriffsprotokollen. Dazu benötigen Sie Berechtigungen für die angegebenen Daten, AWS-Service sodass Sie die Übermittlung von Protokollen aktivieren können.

Um die erforderlichen Berechtigungen für jedes Protokollierungsziel zu sehen, wählen Sie eines der folgenden Themen im Amazon CloudWatch Logs-Benutzerhandbuch.

- CloudWatch Protokolle
- Firehose
- Amazon S3

Nachdem Sie die Berechtigungen für Ihr Protokollierungsziel eingerichtet haben, können Sie die Standardprotokollierung für Ihre Distribution aktivieren.

> **Note**
>
> CloudFront unterstützt das Senden von Zugriffsprotokollen an verschiedene AWS-Konten (kontenübergreifende). Um die kontoübergreifende Bereitstellung zu ermöglichen, müssen beide Konten (Ihr Konto und das Empfängerkonto) über die erforderlichen Berechtigungen verfügen. Weitere Informationen finden Sie im Aktivieren der Standardprotokollierung für die kontoübergreifende Bereitstellung Abschnitt oder im Cross-account Lieferbeispiel im Amazon CloudWatch Logs-Benutzerhandbuch.

Aktivieren der Standardprotokollierung

Um die Standardprotokollierung zu aktivieren, können Sie die CloudFront Konsole oder die CloudWatch API verwenden.

Inhalt

- Aktivieren Sie die Standardprotokollierung (CloudFront Konsole)
- Aktivieren Sie die Standardprotokollierung (CloudWatch API)

Aktivieren Sie die Standardprotokollierung (CloudFront Konsole)

Um die Standardprotokollierung für eine CloudFront Distribution zu aktivieren (Konsole)

1. Verwenden Sie die CloudFront Konsole, um eine bestehende Distribution zu aktualisieren.
2. Wählen Sie die Registerkarte Logging (Protokollierung) aus.
3. Wählen Sie Hinzufügen und anschließend den Service aus, der Ihre Protokolle erhalten soll:
   - CloudWatch Logs
   - Firehose
   - Amazon S3
4. Wählen Sie als Ziel die Ressource für Ihren Service aus. Wenn Sie Ihre Ressource noch nicht erstellt haben, können Sie Erstellen auswählen oder sich die folgende Dokumentation ansehen.
   - Geben Sie für CloudWatch Logs den Namen der Protokollgruppe ein.
   - Geben Sie für Firehose den Firehose-Bereitstellungsdatenstrom ein.
   - Geben Sie für Amazon S3 den Bucket-Namen ein.

   > **Tip**
   >
   > Um ein Präfix anzugeben, geben Sie das Präfix nach dem Bucket-Namen ein, z. B. `amzn-s3-demo-bucket.s3.amazonaws.com/MyLogPrefix`. Wenn Sie kein Präfix angeben, CloudFront wird automatisch eines für Sie hinzugefügt. Weitere Informationen finden Sie unter Senden von Protokollen an Amazon S3.

5. Für Zusätzliche Einstellungen – optional können Sie die folgenden Optionen angeben:

a. Wählen Sie unter Feldauswahl die Namen der Protokollfelder aus, die Sie an Ihr Ziel senden möchten. Sie können Zugriffsprotokollfelder und eine Teilmenge von Echtzeitzugriffsprotokollfeldern auswählen.

b. (Nur Amazon S3) Geben Sie für die Partitionierung den Pfad zur Partitionierung Ihrer Protokolldateidaten an.

c. (Nur Amazon S3) Für das Hive-compatible Dateiformat können Sie das Kontrollkästchen aktivieren, um Hive-compatible S3-Pfade zu verwenden. Dies vereinfacht das Laden neuer Daten in Ihre Hive-compatible Tools.

d. Geben Sie unter Ausgabeformat Ihr bevorzugtes Format an.

> **Note**
>
> Wenn Sie sich für Parquet entscheiden, fallen für diese Option CloudWatch Gebühren für die Konvertierung Ihrer Zugriffsprotokolle nach Apache Parquet an. Weitere Informationen zur Preisgestaltung finden Sie im Abschnitt Vending Logs. CloudWatch

e. Geben Sie unter Feldtrennzeichen an, wie Protokollfelder getrennt werden sollen.

6. Führen Sie die Schritte zum Aktualisieren oder Erstellen Ihrer Distribution aus.
7. Um ein weiteres Ziel hinzuzufügen, wiederholen Sie die Schritte 3–6.
8. Vergewissern Sie sich auf der Seite Protokolle, dass der Status der Standardprotokolle neben der Distribution aktiviert lautet.
9. (Optional) Um die Cookie-Protokollierung zu aktivieren, wählen Sie Verwalten und dann Einstellungen aus, aktivieren Sie die Cookie-Protokollierung und bestätigen Sie Ihre Auswahl anschließend mit Änderungen speichern.

> **Tip**
>
> Die Cookie-Protokollierung ist eine globale Einstellung, die für die gesamte Standardprotokollierung für Ihre Distribution gilt. Sie können diese Einstellung nicht für separate Bereitstellungsziele überschreiben.

Weitere Informationen zur standardmäßigen Bereitstellung der Protokollierung und den Protokollfeldern finden Sie in der Referenz zur Standardprotokollierung.

Aktivieren Sie die Standardprotokollierung (CloudWatch API)

Sie können die CloudWatch API auch verwenden, um die Standardprotokollierung für Ihre Distributionen zu aktivieren.

Hinweise

- Wenn Sie die CloudWatch API aufrufen, um die Standardprotokollierung zu aktivieren, müssen Sie die Region USA Ost (Nord-Virginia) (`us-east-1`) angeben, auch wenn Sie die regionsübergreifende Zustellung an ein anderes Ziel aktivieren möchten. Wenn Sie beispielsweise Ihre Zugriffsprotokolle an einen S3-Bucket in der Region Europa (Irland) () senden möchten, verwenden Sie die CloudWatch API in der `us-east-1` Region. `eu-west-1`
- Es gibt eine zusätzliche Option, um Cookies in die Standardprotokollierung aufzunehmen. In der CloudFront API ist dies der `IncludeCookies` Parameter. Wenn Sie die Zugriffsprotokollierung mithilfe der CloudWatch API konfigurieren und angeben, dass Sie Cookies einbeziehen möchten, müssen Sie die CloudFront Konsole oder CloudFront API verwenden, um Ihre Distribution so zu aktualisieren, dass sie Cookies enthält. Andernfalls CloudFront können keine Cookies an Ihr Protokollziel gesendet werden. Weitere Informationen finden Sie unter Protokollierung von Cookies.

Um die Standardprotokollierung für eine Distribution (CloudWatch API) zu aktivieren

1. Nachdem Sie eine Distribution erstellt haben, rufen Sie den Amazon-Ressourcennamen (ARN) ab.

   Sie finden den ARN auf der Verteilungsseite in der CloudFront Konsole oder Sie können den GetDistributionAPI-Vorgang verwenden. Ein Distributions-ARN folgt dem Format: `arn:aws:cloudfront::123456789012:distribution/d111111abcdef8`
2. Verwenden Sie als Nächstes den CloudWatch PutDeliverySourceAPI-Vorgang, um eine Lieferquelle für die Verteilung zu erstellen.

   a. Geben Sie einen Namen für die Bereitstellungsquelle ein.

   b. Übergeben Sie den `resourceArn` der Distribution.

   c. Geben Sie unter `logType` als Typ der erfassten Protokolle `ACCESS_LOGS` an.

d. Example Beispiel AWS CLI Befehl put-delivery-source

Im Folgenden finden Sie ein Beispiel für die Konfiguration einer Bereitstellungsquelle für eine Distribution.

```
aws logs put-delivery-source --name S3-delivery --resource-arn
 arn:aws:cloudfront::123456789012:distribution/d111111abcdef8 --log-type
 ACCESS_LOGS
```

Ausgabe

```
{
 "deliverySource": {
 "name": "S3-delivery",
 "arn": "arn:aws:logs:us-east-1:123456789012:delivery-source:S3-delivery",
 "resourceArns": [
 "arn:aws:cloudfront::123456789012:distribution/d111111abcdef8"
 ],
 "service": "cloudfront",
 "logType": "ACCESS_LOGS"
 }
}
```

3. Verwenden Sie den PutDeliveryDestinationAPI-Vorgang, um zu konfigurieren, wo Ihre Protokolle gespeichert werden sollen.

a. Geben Sie unter `destinationResourceArn` den ARN des Ziels an. Dies kann eine CloudWatch Logs-Protokollgruppe, ein Firehose-Lieferstream oder ein Amazon S3 S3-Bucket sein.

b. Geben Sie unter `outputFormat` das Ausgabeformat für Ihre Protokolle an.

c. Example Beispiel AWS CLI Befehl put-delivery-destination

Im Folgenden finden Sie ein Beispiel für die Konfiguration eines Bereitstellungsziels für einen Amazon-S3-Bucket.

```
aws logs put-delivery-destination --name S3-destination --delivery-destination-
configuration destinationResourceArn=arn:aws:s3:::amzn-s3-demo-bucket
```

Ausgabe

```
{
    "name": "S3-destination",
    "arn": "arn:aws:logs:us-east-1:123456789012:delivery-destination:S3-
destination",
    "deliveryDestinationType": "S3",
    "deliveryDestinationConfiguration": {
        "destinationResourceArn": "arn:aws:s3:::amzn-s3-demo-bucket"
    }
}
```

> **Note**
>
> Wenn Sie Protokolle kontoübergreifend bereitstellen, müssen Sie den PutDeliveryDestinationPolicyAPI-Vorgang verwenden, um dem Zielkonto eine AWS Identity and Access Management (IAM-) Richtlinie zuzuweisen. Die IAM-Richtlinie ermöglicht die Bereitstellung von einem Konto in einem anderen Konto.

4. Verwenden Sie den CreateDeliveryAPI-Vorgang, um die Lieferquelle mit dem Ziel zu verknüpfen, das Sie in den vorherigen Schritten erstellt haben. Diese API-Operation verknüpft die Bereitstellungsquelle mit dem Endziel.

   a. Geben Sie unter `deliverySourceName` den Quellnamen an.

   b. Geben Sie unter `deliveryDestinationArn` den ARN für das Bereitstellungsziel an.

   c. Geben Sie unter `fieldDelimiter` die Zeichenfolge zur Trennung der Protokollfelder an.

   d. Geben Sie unter `recordFields` die gewünschten Protokollfelder an.

   e. Wenn Sie S3 verwenden, geben Sie an, ob `enableHiveCompatiblePath` und `suffixPath` verwendet werden sollen.

   Example Beispiel AWS CLI Befehl create-delivery

   Im Folgenden finden Sie ein Beispiel für die Erstellung einer Bereitstellung:

   ```
   aws logs create-delivery --delivery-source-name cf-delivery --delivery-destination-
   arn arn:aws:logs:us-east-1:123456789012:delivery-destination:S3-destination
   ```

   Ausgabe

```
{
    "id": "abcNegnBoTR123",
    "arn": "arn:aws:logs:us-east-1:123456789012:delivery:abcNegnBoTR123",
    "deliverySourceName": "cf-delivery",
    "deliveryDestinationArn": "arn:aws:logs:us-east-1:123456789012:delivery-
destination:S3-destination",
    "deliveryDestinationType": "S3",
    "recordFields": [
        "date",
        "time",
        "x-edge-location",
        "sc-bytes",
        "c-ip",
        "cs-method",
        "cs(Host)",
        "cs-uri-stem",
        "sc-status",
        "cs(Referer)",
        "cs(User-Agent)",
        "cs-uri-query",
        "cs(Cookie)",
        "x-edge-result-type",
        "x-edge-request-id",
        "x-host-header",
        "cs-protocol",
        "cs-bytes",
        "time-taken",
        "x-forwarded-for",
        "ssl-protocol",
        "ssl-cipher",
        "x-edge-response-result-type",
        "cs-protocol-version",
        "fle-status",
        "fle-encrypted-fields",
        "c-port",
        "time-to-first-byte",
        "x-edge-detailed-result-type",
        "sc-content-type",
        "sc-content-len",
        "sc-range-start",
        "sc-range-end",
        "c-country",
        "cache-behavior-path-pattern"
```

```
    ],
     "fieldDelimiter": ""
}
```

5. Vergewissern Sie sich in der CloudFront Konsole auf der Seite Logs, dass der Status der Standardprotokolle neben der Distribution auf Aktiviert steht.

   Weitere Informationen zur standardmäßigen Bereitstellung der Protokollierung und den Protokollfeldern finden Sie in der Referenz zur Standardprotokollierung.

> **Note**
>
> Um die Standardprotokollierung (v2) für die CloudFront Verwendung zu aktivieren AWS CloudFormation, können Sie die folgenden CloudWatch Protokolleigenschaften verwenden:
>
> - Delivery
> - DeliveryDestination
> - DeliverySource
>
> Das `ResourceArn` ist die CloudFront Distribution und `LogType` muss `ACCESS_LOGS` der unterstützte Protokolltyp sein.

Aktivieren der Standardprotokollierung für die kontoübergreifende Bereitstellung

Wenn Sie die Standardprotokollierung für Ihr Konto aktivieren AWS-Konto und Ihre Zugriffsprotokolle an ein anderes Konto weitergeben möchten, stellen Sie sicher, dass Sie das Quellkonto und das Zielkonto korrekt konfigurieren. Das Quellkonto mit der CloudFront Verteilung sendet seine Zugriffsprotokolle an das Zielkonto.

In diesem Beispielverfahren sendet das Quellkonto *111111111111* () seine Zugriffsprotokolle an einen Amazon S3 S3-Bucket im Zielkonto (*222222222222*). Um Zugriffsprotokolle an einen Amazon-S3-Bucket im Zielkonto zu senden, verwenden Sie die AWS CLI.

Konfigurieren des Zielkontos

Führen Sie für das Zielkonto das folgende Verfahren aus.

So konfigurieren Sie das Zielkonto

1. Um das Ziel für die Protokollbereitstellung zu erstellen, können Sie den folgenden AWS CLI -Befehl eingeben. In diesem Beispiel wird die Zeichenfolge *MyLogPrefix* verwendet, um ein Präfix für Ihre Zugriffsprotokolle zu erstellen.

```
aws logs put-delivery-destination --name cloudfront-delivery-destination --
delivery-destination-configuration "destinationResourceArn=arn:aws:s3:::amzn-s3-
demo-bucket-cloudfront-logs/MyLogPrefix"
```

Ausgabe

```
{
    "deliveryDestination": {
        "name": "cloudfront-delivery-destination",
        "arn": "arn:aws:logs:us-east-1:222222222222:delivery-
destination:cloudfront-delivery-destination",
        "deliveryDestinationType": "S3",
        "deliveryDestinationConfiguration": {"destinationResourceArn":
 "arn:aws:s3:::amzn-s3-demo-bucket-cloudfront-logs/MyLogPrefix"}
    }
}
```

> **Note**
>
> Wenn Sie einen S3-Bucket ohne Präfix angeben, CloudFront wird automatisch das `AWSLogs/<account-ID>/CloudFront` als Präfix angehängt, das `suffixPath` im S3-Lieferziel erscheint. Weitere Informationen finden Sie unter S3DeliveryConfiguration.

2. Fügen Sie die Ressourcenrichtlinie für das Protokollbereitstellungsziel hinzu, um dem Quellkonto die Protokollbereitstellung zu ermöglichen.

   Ersetzen Sie in der folgenden Richtlinie *111111111111* durch die Quellkonto-ID und geben Sie den Lieferziel-ARN aus der Ausgabe in Schritt 1 an.

   JSON

```
{
    "Version":"2012-10-17",
```

```
    "Statement": [
        {
            "Sid": "AllowCreateDelivery",
            "Effect": "Allow",
            "Principal": {"AWS": "111111111111"},
            "Action": ["logs:CreateDelivery"],
            "Resource": "arn:aws:logs:us-east-1:222222222222:delivery-
destination:cloudfront-delivery-destination"
        }
    ]
}
```

3. Speichern Sie die Datei, z. B. `deliverypolicy.json`.
4. Geben Sie den folgenden AWS CLI Befehl ein, um die vorherige Richtlinie an das Lieferziel anzuhängen.

```
aws logs put-delivery-destination-policy --delivery-destination-name cloudfront-
delivery-destination --delivery-destination-policy file://deliverypolicy.json
```

5. Fügen Sie die folgende Erklärung zur Richtlinie des Ziel-Buckets in Amazon S3 hinzu und ersetzen Sie dabei den Ressourcen-ARN und die Quellkonto-ID. Diese Richtlinie ermöglicht es dem Service-Prinzipal `delivery.logs.amazonaws.com`, die Aktion `s3:PutObject` auszuführen.

```
{
    "Sid": "AWSLogsDeliveryWrite",
    "Effect": "Allow",
    "Principal": {"Service": "delivery.logs.amazonaws.com"},
    "Action": "s3:PutObject",
    "Resource": "arn:aws:s3:::amzn-s3-demo-bucket-cloudfront-logs/*",
    "Condition": {
        "StringEquals": {
            "s3:x-amz-acl": "bucket-owner-full-control",
            "aws:SourceAccount": "111111111111"
        },
        "ArnLike": {"aws:SourceArn": "arn:aws:logs:us-east-1:111111111111:delivery-
source:*"}
    }
}
```

6. Wenn Sie dies AWS KMS für Ihren Bucket verwenden, fügen Sie der KMS-Schlüsselrichtlinie die folgende Anweisung hinzu, um dem `delivery.logs.amazonaws.com` Dienstprinzipal Berechtigungen zu erteilen.

```
{
    "Sid": "Allow Logs Delivery to use the key",
    "Effect": "Allow",
    "Principal": {"Service": "delivery.logs.amazonaws.com"},
    "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
        "kms:DescribeKey"
    ],
    "Resource": "*",
    "Condition": {
        "StringEquals": {"aws:SourceAccount": "111111111111"},
        "ArnLike": {"aws:SourceArn": "arn:aws:logs:us-east-1:111111111111:delivery-
source:*"}
    }
}
```

Konfigurieren des Quellkontos

Gehen Sie nach der Konfiguration des Zielkontos wie folgt vor, um die Bereitstellungsquelle zu erstellen und die Protokollierung für die Distribution im Quellkonto zu aktivieren.

So konfigurieren Sie das Quellkonto

1. Erstellen Sie eine Zustellungsquelle für die CloudFront Standardprotokollierung, sodass Sie Protokolldateien an CloudWatch Logs senden können.

   Sie können den folgenden AWS CLI Befehl eingeben und dabei den Namen und den ARN Ihrer Distribution ersetzen.

```
aws logs put-delivery-source --name s3-cf-delivery --resource-arn
 arn:aws:cloudfront::111111111111:distribution/E1TR1RHV123ABC --log-type
 ACCESS_LOGS
```

Ausgabe

```
{
    "deliverySource": {
        "name": "s3-cf-delivery",
        "arn": "arn:aws:logs:us-east-1:111111111111:delivery-source:s3-cf-
delivery",
        "resourceArns":
 ["arn:aws:cloudfront::111111111111:distribution/E1TR1RHV123ABC"],
        "service": "cloudfront",
        "logType": "ACCESS_LOGS"
    }
}
```

2. Erstellen Sie eine Bereitstellung, um die Protokollbereitstellungsquelle des Quellkontos dem Protokollbereitstellungsziel des Zielkontos zuzuordnen.

   Geben Sie im folgenden AWS CLI Befehl den Lieferziel-ARN aus der Ausgabe in Schritt 1: Zielkonto konfigurieren an.

   ```
   aws logs create-delivery --delivery-source-name s3-cf-delivery --
   delivery-destination-arn arn:aws:logs:us-east-1:222222222222:delivery-
   destination:cloudfront-delivery-destination
   ```

   Ausgabe

   ```
   {
       "delivery": {
           "id": "OPmOpLahVzhx1234",
           "arn": "arn:aws:logs:us-east-1:111111111111:delivery:OPmOpLahVzhx1234",
           "deliverySourceName": "s3-cf-delivery",
           "deliveryDestinationArn": "arn:aws:logs:us-east-1:222222222222:delivery-
   destination:cloudfront-delivery-destination",
           "deliveryDestinationType": "S3",
           "recordFields": [
               "date",
               "time",
               "x-edge-location",
               "sc-bytes",
               "c-ip",
               "cs-method",
   ```

```
            "cs(Host)",
            "cs-uri-stem",
            "sc-status",
            "cs(Referer)",
            "cs(User-Agent)",
            "cs-uri-query",
            "cs(Cookie)",
            "x-edge-result-type",
            "x-edge-request-id",
            "x-host-header",
            "cs-protocol",
            "cs-bytes",
            "time-taken",
            "x-forwarded-for",
            "ssl-protocol",
            "ssl-cipher",
            "x-edge-response-result-type",
            "cs-protocol-version",
            "fle-status",
            "fle-encrypted-fields",
            "c-port",
            "time-to-first-byte",
            "x-edge-detailed-result-type",
            "sc-content-type",
            "sc-content-len",
            "sc-range-start",
            "sc-range-end",
            "c-country",
            "cache-behavior-path-pattern"
        ],
        "fieldDelimiter": "\t"
    }
}
```

3. Überprüfen Sie, ob die kontoübergreifende Bereitstellung erfolgreich war.

    a. Melden Sie sich *source* über das Konto bei der CloudFront Konsole an und wählen Sie Ihre Distribution aus. Auf der Registerkarte Protokollierung finden Sie unter Typ einen Eintrag, der für die kontoübergreifende S3-Protokollbereitstellung erstellt wurde.

    b. Melden Sie sich *destination* über das Konto bei der Amazon S3-Konsole an und wählen Sie Ihren Amazon S3 S3-Bucket aus. Sie werden das Präfix *MyLogPrefix* im Bucket-Namen und in allen Zugriffsprotokollen sehen, die an diesen Ordner übermittelt wurden.

Ausgabedateiformat

Je nachdem, welches Bereitstellungsziel Sie auswählen, können Sie eines der folgenden Formate für die Protokolldateien angeben.

- JSON
- Plain
- w3c
- Raw
- Parquet (nur Amazon S3)

> **Note**
>
> Sie können das Ausgabeformat nur festlegen, wenn Sie das Bereitstellungsziel zum ersten Mal erstellen. Diese kann später nicht mehr aktualisiert werden. Um das Ausgabeformat zu ändern, löschen Sie die Bereitstellung und erstellen Sie eine neue.

Weitere Informationen finden Sie PutDeliveryDestinationin der Amazon CloudWatch Logs API-Referenz.

Bearbeiten der Standardprotokollierungseinstellungen

Sie können die Protokollierung mithilfe der CloudFront Konsole oder der CloudWatch API aktivieren oder deaktivieren und andere Protokolleinstellungen aktualisieren. Ihre Änderungen der Protokollierungseinstellungen werden innerhalb von 12 Stunden wirksam.

Weitere Informationen finden Sie unter den folgenden Themen:

- Informationen zum Aktualisieren einer Distribution mithilfe der CloudFront Konsole finden Sie unterEine Verteilung aktualisieren.
- Informationen zum Aktualisieren einer Distribution mithilfe der CloudFront API finden Sie UpdateDistributionin der Amazon CloudFront API-Referenz.
- Weitere Informationen zu CloudWatch Logs-API-Vorgängen finden Sie in der Amazon CloudWatch Logs API-Referenz.

Zugriffsprotokollfelder

Sie können dieselben Protokollfelder auswählen, die die Standardprotokollierung (Legacy) unterstützt. Weitere Informationen finden Sie unter Protokolldateifelder.

Darüber hinaus können Sie die folgenden Protokollfelder für den Echtzeitzugriff auswählen.

1. **`timestamp(ms)`** – der Zeitstempel in Millisekunden
2. **`origin-fbl`**— Die Anzahl der Sekunden der Latenz im ersten Byte zwischen CloudFront und Ihrem Ursprung.
3. **`origin-lbl`**— Die Anzahl der Sekunden der Latenz im letzten Byte zwischen CloudFront und Ihrem Ursprung.
4. **`asn`** – die autonome Systemnummer (ASN) des Viewers
5. **`c-country`** – ein Ländercode, der den geografischen Standort des Viewers darstellt und durch die IP-Adresse des Viewers festgelegt wird. Die Liste der Ländercodes finden Sie unter ISO 3166-1 alpha-2.
6. **`cache-behavior-path-pattern`** – das Pfadmuster, das das Cacheverhalten identifiziert, das der Viewer-Anforderung entspricht
7. **`viewer-request-log-data`**— Enthält die benutzerdefinierten Daten, die Ihre CloudFront Viewer-Anforderungsfunktion mithilfe der `cf.logCustomData()` Hilfsmethode sendet. CloudFront URL-encodes den Wert und kürzt ihn auf ein Maximum von 800 Byte. Wenn dem Viewer-Anforderungsereignis keine CloudFront Funktion zugeordnet ist oder wenn die Funktion nicht aufgerufen wird`cf.logCustomData()`, gilt dies für dieses Feld. - Weitere Informationen finden Sie unter Allgemeine Hilfsmethoden.
8. **`viewer-response-log-data`**— Enthält die benutzerdefinierten Daten, die Ihre CloudFront Viewer-Antwortfunktion mithilfe der `cf.logCustomData()` Hilfsmethode sendet. CloudFront URL-encodes den Wert und kürzt ihn auf ein Maximum von 800 Byte. Wenn dem Antwortereignis des Betrachters keine CloudFront Funktion zugeordnet ist oder die Funktion nicht aufgerufen wird`cf.logCustomData()`, gilt dies für dieses Feld. - Weitere Informationen finden Sie unter Allgemeine Hilfsmethoden.

Protokolle an CloudWatch Logs senden

Um Protokolle an Logs zu CloudWatch senden, erstellen oder verwenden Sie eine bestehende CloudWatch Logs-Protokollgruppe. Weitere Informationen zur Konfiguration einer CloudWatch Logs-Log-Gruppe finden Sie unter Arbeiten mit Log-Gruppen und Log-Streams.

Nach dem Erstellen Ihrer Protokollgruppe benötigen Sie die erforderlichen Berechtigungen zur Ausführung der Standardprotokollierung. Weitere Informationen zu den erforderlichen Berechtigungen finden Sie unter An Logs sent to CloudWatch Logs im Amazon CloudWatch Logs-Benutzerhandbuch.

Hinweise

- Wenn Sie den Namen der CloudWatch Logs-Protokollgruppe angeben, verwenden Sie nur das Regex-Muster. `[\w-]` Weitere Informationen zur PutDeliveryDestinationAPI-Operation finden Sie in der Amazon CloudWatch Logs API-Referenz.
- Stellen Sie sicher, dass Ihre Ressourcenrichtlinie für die Protokollgruppe die Größenbeschränkung nicht überschreitet. Weitere Informationen finden Sie im Abschnitt Überlegungen zur Größenbeschränkung der Ressourcenrichtlinie für Protokollgruppen im Thema CloudWatch Protokolle.

Beispiel für ein Zugriffsprotokoll, das an CloudWatch Logs gesendet wurde

```
{
"date": "2024-11-14",
"time": "21:34:06",
"x-edge-location": "SOF50-P2",
"asn": "16509",
"timestamp(ms)": "1731620046814",
"origin-fbl": "0.251",
"origin-lbl": "0.251",
"x-host-header": "d111111abcdef8.cloudfront.net",
"cs(Cookie)": "examplecookie=value"
}
```

Senden von Protokollen an Firehose

Um Protokolle an Firehose zu senden, erstellen Sie einen Firehose-Bereitstellungsdatenstrom oder verwenden Sie einen bereits vorhandenen. Geben Sie dann den Firehose-Bereitstellungsdatenstrom als Protokollbereitstellungsziel an. Sie müssen einen Firehose-Bereitstellungsdatenstrom für die Region USA Ost (Nord-Virginia) us-east-1 angeben.

Weitere Informationen zum Erstellen eines Bereitstellungsdatenstroms finden Sie unter Erstellen eines Bereitstellungsdatenstroms für Amazon Data Firehose.

Nach dem Erstellen Ihres Bereitstellungsdatenstroms benötigen Sie die erforderlichen Berechtigungen zur Ausführung der Standardprotokollierung. Weitere Informationen finden Sie unter An Firehose gesendete Logs im Amazon CloudWatch Logs-Benutzerhandbuch.

> **Note**
>
> Wenn Sie den Namen des Firehose-Datenstroms angeben, verwenden Sie nur das Regex-Muster `[\w-]`. Weitere Informationen zur PutDeliveryDestinationAPI-Operation finden Sie in der Amazon CloudWatch Logs API-Referenz.

Beispiel für ein an Firehose gesendetes Zugriffsprotokoll

```
{"date":"2024-11-15","time":"19:45:51","x-edge-location":"SOF50-
P2","asn":"16509","timestamp(ms)":"1731699951183","origin-fbl":"0.254","origin-
lbl":"0.254","x-host-
header":"d111111abcdef8.cloudfront.net","cs(Cookie)":"examplecookie=value"}
{"date":"2024-11-15","time":"19:45:52","x-edge-location":"SOF50-
P2","asn":"16509","timestamp(ms)":"1731699952950","origin-fbl":"0.125","origin-
lbl":"0.125","x-host-
header":"d111111abcdef8.cloudfront.net","cs(Cookie)":"examplecookie=value"}
```

Senden von Protokollen an Amazon S3

Um Ihre Zugriffsprotokolle an Amazon S3 zu senden, erstellen Sie einen S3-Bucket oder verwenden Sie einen bereits vorhandenen. Wenn Sie die Anmeldung aktivieren CloudFront, geben Sie den Bucket-Namen an. Weitere Informationen zum Erstellen eines Buckets finden Sie unter Erstellen von Buckets im Benutzerhandbuch für Amazon Simple Storage Service.

Nach dem Erstellen Ihres Buckets benötigen Sie die erforderlichen Berechtigungen zur Ausführung der Standardprotokollierung. Weitere Informationen finden Sie unter An Amazon S3 gesendete Logs im Amazon CloudWatch Logs-Benutzerhandbuch.

- Nachdem Sie die Protokollierung aktiviert haben, AWS werden automatisch die erforderlichen Bucket-Richtlinien für Sie hinzugefügt.
- Sie können auch S3-Buckets in den Opt-In- AWS-Regionen verwenden.

> **Note**
>
> Wenn Sie die Standardprotokollierung (Legacy) bereits aktiviert haben und die Standardprotokollierung (v2) für Amazon S3 aktivieren möchten, empfehlen wir Ihnen, einen anderen Amazon-S3-Bucket anzugeben oder einen separaten Pfad in demselben Bucket zu verwenden (z. B. ein Protokollpräfix oder eine Partitionierung zu verwenden). Auf diese Weise behalten Sie den Überblick darüber, welche Protokolldateien mit welcher Distribution verknüpft sind, und es wird verhindert, dass sich Protokolldateien gegenseitig überschreiben.

Themen

- Angeben eines S3-Buckets
- Partitionierung
- Hive-compatible Format des Dateinamens
- Beispielpfade für Zugriffsprotokolle
- Beispiel eines an Amazon S3 gesendeten Zugriffsprotokolls

Angeben eines S3-Buckets

Beachten Sie Folgendes, wenn Sie einen S3-Bucket als Bereitstellungsziel angeben.

Der S3-Bucket-Name kann nur das Regex-Muster `[\w-]` verwenden. Weitere Informationen zur PutDeliveryDestinationAPI-Operation finden Sie in der Amazon CloudWatch Logs API-Referenz.

Wenn Sie ein Präfix für Ihren S3-Bucket angegeben haben, werden Ihre Protokolle unter diesem Pfad angezeigt. Wenn Sie kein Präfix angeben, CloudFront wird das `AWSLogs/{account-id}/CloudFront` Präfix automatisch für Sie angehängt.

Weitere Informationen finden Sie unter Beispielpfade für Zugriffsprotokolle.

Partitionierung

Sie können die Partitionierung verwenden, um Ihre Zugriffsprotokolle zu organisieren, wenn sie CloudFront an Ihren S3-Bucket gesendet werden. Dies hilft Ihnen, Ihre Zugriffsprotokolle auf der Grundlage des von Ihnen gewünschten Pfads zu organisieren und zu finden.

Sie können die folgenden Variablen verwenden, um einen Ordnerpfad zu erstellen.

- {DistributionId} oder {distributionid}
- {yyyy}
- {MM}
- {dd}
- {HH}
- {accountid}

Sie können eine beliebige Anzahl von Variablen verwenden und Ordnernamen in Ihrem Pfad angeben. CloudFrontverwendet dann diesen Pfad, um eine Ordnerstruktur für Sie im S3-Bucket zu erstellen.

Beispiele

- *my_distribution_log_data*/{DistributionId}/*logs*
- /*cloudfront*/{DistributionId}/*my_distribution_log_data*/{yyyy}/{MM}/{dd}/{HH}/*logs*

> **Note**
>
> Sie können eine der beiden Variablen für die Distributions-ID im Suffixpfad verwenden. Wenn Sie jedoch Zugriffsprotokolle an senden, müssen Sie die {distributionid} Variable verwenden AWS Glue, da AWS Glue erwartet wird, dass Partitionsnamen in Kleinbuchstaben geschrieben werden. Aktualisieren Sie Ihre bestehende Protokollkonfiguration, um CloudFront sie durch zu {DistributionId} ersetzen{distributionid}.

Hive-compatible Format des Dateinamens

Sie können diese Option verwenden, damit S3-Objekte, die bereitgestellte Zugriffsprotokolle enthalten, eine Präfixstruktur verwenden, die die Integration mit Apache Hive ermöglicht. Weitere Informationen finden Sie unter CreateDelivery-API-Operation.

Example Beispiel

```
/cloudfront/DistributionId={DistributionId}/my_distribution_log_data/year={yyyy}/
month={MM}/day={dd}/hour={HH}/logs
```

Weitere Informationen zur Partitionierung und den Hive-compatible Optionen finden Sie in dem S3DeliveryConfigurationElement in der Amazon CloudWatch Logs API-Referenz.

Beispielpfade für Zugriffsprotokolle

Wenn Sie einen S3-Bucket als Ziel angeben, können Sie die folgenden Optionen verwenden, um den Pfad zu Ihren Zugriffsprotokollen zu erstellen:

- Ein Amazon-S3-Bucket mit oder ohne Präfix
- Partitionierung, indem Sie eine CloudFront bereitgestellte Variable verwenden oder eine eigene eingeben
- Aktivierung der Option Hive-compatible

Die folgenden Tabellen zeigen, wie Ihre Zugriffsprotokolle in Ihrem Bucket angezeigt werden, abhängig von den ausgewählten Optionen.

Amazon-S3-Bucket mit Präfix

Name des Amazon-S3-Buckets	Im Suffixpfad angegebene Partition	Aktualisierter Suffixpfad	Hive-compatible aktiviert?	Zugriffsprotokolle werden gesendet an
`amzn-s3-demo-bucket/MyLogPrefix`	Keine	Keine	Nein	`amzn-s3-demo-bucket/MyLogPrefix/`
`amzn-s3-demo-bucket/MyLogPrefix`	`myFolderA/`	`myFolderA/`	Nein	`amzn-s3-demo-bucket/MyLogPrefix/myFolderA/`
`amzn-s3-demo-bucket/MyLogPrefix`	`myFolderA/{yyyy}`	`myFolderA/{yyyy}`	Ja	`amzn-s3-demo-bucket/MyLogPrefix/myFo`

Name des Amazon-S3-Buckets	Im Suffixpfad angegebene Partition	Aktualisierter Suffixpfad	Hive-compatible aktiviert?	Zugriffsprotokolle werden gesendet an
				`lderA/yea r=2025`

Amazon-S3-Bucket ohne Präfix

Name des Amazon-S3-Buckets	Im Suffixpfad angegebene Partition	Aktualisierter Suffixpfad	Hive-compatible aktiviert?	Zugriffsprotokolle werden gesendet an
`amzn-s3-d emo-bucket`	Keine	`AWSLogs/{ account-i d}/CloudF ront/`	Nein	`amzn-s3-d emo-bucke t/AWSLogs / <your-acc ount-ID> / CloudFront/`
`amzn-s3-d emo-bucket`	`myFolderA/`	`AWSLogs/{ account-i d}/CloudF ront/myFo lderA/`	Nein	`amzn-s3-d emo-bucke t/AWSLogs / <your-acc ount-ID> / CloudFront/ myFolderA/`
`amzn-s3-d emo-bucket`	`myFolderA/`	`AWSLogs/{ account-i d}/CloudF ront/myFo lderA/`	Ja	`amzn-s3-d emo-bucke t/AWSLogs /aws- account- id=<your-acc ount-ID> /`

Name des Amazon-S3-Buckets	Im Suffixpfad angegebene Partition	Aktualisierter Suffixpfad	Hive-compatible aktiviert?	Zugriffsprotokolle werden gesendet an
				`CloudFront/myFolderA/`
`amzn-s3-demo-bucket`	`myFolderA/{yyyy}`	`AWSLogs/{account-id}/CloudFront/myFolderA/{yyyy}`	Ja	`amzn-s3-demo-bucket/AWSLogs/aws-account-id=`*`<your-account-ID>`* `/CloudFront/myFolderA/year=2025`

AWS-Konto ID als Partition

Name des Amazon-S3-Buckets	Im Suffixpfad angegebene Partition	Aktualisierter Suffixpfad	Hive-compatible aktiviert?	Zugriffsprotokolle werden gesendet an
`amzn-s3-demo-bucket`	Keine	`AWSLogs/{account-id}/CloudFront/`	Ja	`amzn-s3-demo-bucket/AWSLogs/aws-account-id=`*`<your-account-ID>`* `/CloudFront/`
`amzn-s3-demo-bucket`	`myFolderA/{accountid}`	`AWSLogs/{account-id}/CloudFront/myFo`	Ja	`amzn-s3-demo-bucket/AWSLogs/aws-`

Name des Amazon-S3-Buckets	Im Suffixpfad angegebene Partition	Aktualisierter Suffixpfad	Hive-compatible aktiviert?	Zugriffsprotokolle werden gesendet an
		`lderA/{accountid}`		`account-id=<your-account-ID>/CloudFront/myFolderA/accountid= <your-account-ID>`

Hinweise

- Die `{account-id}` Variable ist reserviert für CloudFront. CloudFrontfügt diese Variable automatisch zu Ihrem Suffixpfad hinzu, wenn Sie einen Amazon S3 S3-Bucket ohne Präfix angeben. Wenn Ihre Logs es sind Hive-compatible, wird diese Variable als `aws-account-id` angezeigt.
- Sie können die `{accountid}` Variable verwenden, um Ihre CloudFront Konto-ID zum Suffixpfad hinzuzufügen. Wenn Ihre Logs es sind Hive-compatible, wird diese Variable als `accountid` angezeigt.
- Weitere Hinweise zum Suffixpfad finden Sie unter S3DeliveryConfiguration.

Beispiel eines an Amazon S3 gesendeten Zugriffsprotokolls

```
#Fields: date time x-edge-location asn timestamp(ms) x-host-header cs(Cookie)
2024-11-14    22:30:25    SOF50-P2    16509    1731623425421
d111111abcdef8.cloudfront.net    examplecookie=value2
```

Deaktivieren der Standardprotokollierung

Sie können die Standardprotokollierung für Ihre Distribution deaktivieren, wenn Sie sie nicht mehr benötigen.

So deaktivieren Sie die Standardprotokollierung

1. Melden Sie sich in der CloudFront -Konsole an.
2. Wählen Sie Distribution und dann Ihre Distributions-ID aus.
3. Wählen Sie Protokollierung und dann unter Zugriffsprotokollziele das Ziel aus.
4. Wählen Sie Verwalten und dann Löschen aus.
5. Wiederholen Sie den vorherigen Schritt, wenn Sie mehr als eine Standardprotokollierung haben.

> **Note**
>
> Wenn Sie die Standardprotokollierung aus der CloudFront Konsole löschen, werden durch diese Aktion nur die Lieferung und das Übermittlungsziel gelöscht. Die Lieferquelle wird dadurch nicht aus Ihrer AWS-Konto gelöscht. Um eine Bereitstellungsquelle zu löschen, geben Sie den Namen der Bereitstellungsquelle im Befehl `aws logs delete-delivery-source --name DeliverySourceName` an. Weitere Informationen finden Sie DeleteDeliverySourcein der Amazon CloudWatch Logs API-Referenz.

Fehlerbehebung

Verwenden Sie die folgenden Informationen, um häufig auftretende Probleme zu beheben, wenn Sie mit der CloudFront Standardprotokollierung (v2) arbeiten.

Bereitstellungsquelle bereits vorhanden

Wenn Sie die Standardprotokollierung für eine Distribution aktivieren, erstellen Sie eine Bereitstellungsquelle. Anschließend verwenden Sie diese Versandquelle, um Lieferungen an den gewünschten Zieltyp zu erstellen: CloudWatch Logs, Firehose, Amazon S3. Derzeit ist nur eine Bereitstellungsquelle pro Distribution möglich. Wenn Sie versuchen, eine weitere Bereitstellungsquelle für dieselbe Distribution zu erstellen, wird die folgende Fehlermeldung angezeigt.

```
This ResourceId has already been used in another Delivery Source in this account
```

Um eine weitere Bereitstellungsquelle zu erstellen, löschen Sie zuerst die vorhandene. Weitere Informationen finden Sie DeleteDeliverySourcein der Amazon CloudWatch Logs API-Referenz.

Ich habe den Suffixpfad geändert und der Amazon-S3-Bucket kann meine Protokolle nicht empfangen

Wenn Sie die Standardprotokollierung (v2) aktiviert haben und einen Bucket-ARN ohne Präfix angeben, CloudFront wird der folgende Standard an den Suffixpfad angehängt:. `AWSLogs/{account-id}/CloudFront` Wenn Sie die CloudFront Konsole oder den UpdateDeliveryConfigurationAPI-Vorgang verwenden, um einen anderen Suffixpfad anzugeben, müssen Sie die Amazon S3 S3-Bucket-Richtlinie aktualisieren, sodass sie denselben Pfad verwendet.

Example Beispiel: Aktualisieren des Suffixpfads

1. Ihr Standard-Suffixpfad ist `AWSLogs/{account-id}/CloudFront` und Sie ersetzen ihn durch `myFolderA`.
2. Da sich Ihr neuer Suffixpfad von dem in der Richtlinie für den Amazon-S3-Bucket angegebenen Pfad unterscheidet, werden Ihre Zugriffsprotokolle nicht übermittelt.
3. Sie können einen der folgenden Schritte ausführen:
   - Aktualisieren Sie die Berechtigung des Amazon-S3-Buckets von `amzn-s3-demo-bucket/AWSLogs/<your-account-ID>/CloudFront/*` auf `amzn-s3-demo-bucket/myFolderA/*`.
   - Aktualisieren Sie Ihre Protokollierungskonfiguration, um wieder das Standardsuffix `AWSLogs/{account-id}/CloudFront` zu verwenden.

Weitere Informationen finden Sie unter Berechtigungen.

Löschen von Protokolldateien

CloudFront löscht Protokolldateien nicht automatisch von Ihrem Ziel. Weitere Informationen zum Löschen von Protokolldateien finden Sie in den folgenden Themen:

Amazon S3

- Löschen von Objekten im Konsolen-Benutzerhandbuch für Amazon Simple Storage Service

CloudWatch Logs

- Arbeiten mit Protokollgruppen und Protokollstreams im Amazon CloudWatch Logs-Benutzerhandbuch

- DeleteLogGroupin der Amazon CloudWatch Logs API-Referenz

Firehose

- DeleteDeliveryStreamin der Amazon Data Firehose API-Referenz

Preisgestaltung

CloudFront berechnet keine Gebühren für die Aktivierung von Standardprotokollen. Je nach dem von Ihnen ausgewählten Ziel für die Protokollbereitstellung können jedoch Gebühren für die Bereitstellung, Erfassung, Speicherung oder den Zugriff anfallen. Weitere Informationen finden Sie unter Amazon CloudWatch Logs Pricing. Wählen Sie unter Kostenpflichtiges Kontingent die Registerkarte Protokolle und konsultieren Sie die Informationen zum jeweiligen Bereitstellungsziel unter Verkaufte Protokolle.

Weitere Informationen zu den jeweiligen AWS-Service Preisen finden Sie in den folgenden Themen:

- Amazon CloudWatch Logs — Preise
- Amazon Data Firehose – Preise
- Amazon S3 – Preise

> **Note**
>
> Für die Übertragung von Protokolldateien an Amazon S3 fallen keine zusätzlichen Gebühren an, obwohl Ihnen Amazon S3 Gebühren für die Speicherung und den Zugriff auf die Protokolldateien berechnet. Wenn Sie die Parquet-Option aktivieren, um Ihre Zugriffsprotokolle in Apache Parquet zu konvertieren, fallen für diese Option CloudWatch Gebühren an. Weitere Informationen zur Preisgestaltung finden Sie im Abschnitt Vending Logs. CloudWatch

Konfigurieren der Standardprotokollierung (Legacy)

Hinweise

- Dieses Thema bezieht sich auf die vorherige Version der Standardprotokollierung. Informationen zur neuesten Version finden Sie unter Konfigurieren der Standardprotokollierung (v2).
- Wenn Sie die Standardprotokollierung (Legacy) bereits aktiviert haben und die Standardprotokollierung (v2) für Amazon S3 aktivieren möchten, empfehlen wir Ihnen, einen anderen Amazon-S3-Bucket anzugeben oder einen separaten Pfad in demselben Bucket zu verwenden (z. B. ein Protokollpräfix oder eine Partitionierung zu verwenden). Auf diese Weise behalten Sie den Überblick darüber, welche Protokolldateien mit welcher Distribution verknüpft sind, und es wird verhindert, dass sich Protokolldateien gegenseitig überschreiben.

Führen Sie für den Einstieg in die Standardprotokollierung (Legacy) die folgenden Schritte aus:

1. Wählen Sie einen Amazon-S3-Bucket aus, der Ihre Protokolle empfängt, und fügen Sie die erforderlichen Berechtigungen hinzu.
2. Konfigurieren Sie die Standardprotokollierung (Legacy) CloudFront über die Konsole oder die CloudFront API. Sie können nur einen Amazon-S3-Bucket zum Empfang Ihrer Protokolle auswählen.
3. Zeigen Sie Ihre Zugriffsprotokolle an.

Auswählen eines Amazon-S3-Buckets für Standardprotokolle

Wenn Sie die Protokollierung für eine Distribution aktivieren, geben Sie den Amazon S3-Bucket CloudFront an, in dem Sie Protokolldateien speichern möchten. Wenn Sie Amazon S3 als Ursprung verwenden, empfehlen wir, einen separaten Bucket für Ihre Protokolldateien zu verwenden.

Geben Sie den Amazon S3-Bucket CloudFront an, in dem Sie Zugriffsprotokolle speichern möchten, `amzn-s3-demo-bucket.s3.amazonaws.com` z. B.

Sie können Protokolldateien für mehrere Verteilungen in demselben Bucket speichern. Wenn Sie die Protokollierung aktivieren, können Sie ein optionales Präfix für den Dateinamen festlegen; so behalten Sie den Überblick darüber, welche Protokolldateien welchen Verteilungen zugeordnet sind.

Informationen zur Auswahl eines S3-Buckets

- Für Ihren Bucket muss die Zugriffssteuerungsliste (ACL) aktiviert sein. Wenn Sie in der CloudFront Konsole einen Bucket ohne aktivierte ACL auswählen, wird eine Fehlermeldung angezeigt. Siehe Berechtigungen.
- Wählen Sie keinen Amazon-S3-Bucket, wenn S3-Objekteigentümerschaft auf Bucket-Eigentümer erzwungen festgelegt ist. Diese Einstellung deaktiviert ACLs für den Bucket und die darin enthaltenen Objekte, wodurch CloudFront verhindert wird, dass Protokolldateien an den Bucket gesendet werden.
- Legacy-Logging unterstützt Amazon S3-Buckets in Opt-In-Regionen nicht. Bitte wählen Sie eine Region, die standardmäßig aktiviert ist, oder verwenden Sie Standard Logging V2, das Opt-in-Regionen und zusätzliche Funktionen unterstützt. Eine Liste der Standard- und Opt-in-Regionen finden Sie unter. AWS-Regionen

Berechtigungen

Important

Ab April 2023 müssen Sie S3-ACLs für neue S3-Buckets aktivieren, die für Standardprotokolle verwendet werden. CloudFront Sie können ACLs aktivieren, wenn Sie einen Bucket erstellen, oder ACLs für einen vorhandenen Bucket aktivieren. Weitere Informationen zu den Änderungen finden Sie unter Häufig gestellte Fragen zu den Standardeinstellungen für neue S3-Buckets im Amazon Simple Storage Service-Benutzerhandbuch und im News-Blog Heads-Up: Amazon S3-Sicherheitsänderungen kommen im AWS April 2023.

Sie AWS-Konto müssen über die folgenden Berechtigungen für den Bucket verfügen, den Sie für Protokolldateien angeben:

- Die ACL muss Ihnen `FULL_CONTROL` für den Bucket gewähren. Wenn Sie der Bucket-Eigentümer sind, verfügt Ihr Konto standardmäßig über diese Berechtigung. Sind Sie das nicht, muss der Bucket-Eigentümer die ACL für den Bucket aktualisieren.
- `s3:GetBucketAcl`
- `s3:PutBucketAcl`

ACL für den Bucket

Wenn Sie eine Distribution erstellen oder aktualisieren und die Protokollierung aktivieren, CloudFront verwendet es diese Berechtigungen, um die ACL für den Bucket zu aktualisieren und dem `awslogsdelivery` Konto die `FULL_CONTROL` Berechtigung zu erteilen. Das `awslogsdelivery`-Konto schreibt Protokolldateien in den Bucket. Wenn Ihr Konto nicht über die erforderlichen Berechtigungen zum Aktualisieren der ACL verfügt, schlägt das Erstellen oder Aktualisieren der Verteilung fehl.

Wenn Sie programmgesteuert eine Anfrage senden, um einen Bucket zu erstellen, aber ein Bucket mit dem angegebenen Namen bereits vorhanden ist, setzt S3 in einigen Fällen die Berechtigungen für den Bucket auf den Standardwert zurück. Wenn Sie so konfiguriert haben CloudFront , dass Zugriffsprotokolle in einem S3-Bucket gespeichert werden und Sie keine Logs mehr in diesem Bucket abrufen, überprüfen Sie die Berechtigungen für den Bucket, um sicherzustellen, dass dieser CloudFront über die erforderlichen Berechtigungen verfügt.

Wiederherstellung der ACL für den Bucket

Wenn Sie Berechtigungen für das `awslogsdelivery`-Konto aufheben, kann CloudFront keine Protokolle in dem S3-Bucket speichern. Damit Sie wieder mit CloudFront dem Speichern von Protokollen für Ihre Distribution beginnen können, stellen Sie die ACL-Berechtigung wieder her, indem Sie einen der folgenden Schritte ausführen:

- Deaktivieren Sie die Anmeldung für Ihre Distribution und aktivieren Sie sie dann erneut. CloudFront Weitere Informationen finden Sie unter Standardprotokollierung.
- Fügen Sie die ACL-Berechtigung für `awslogsdelivery` manuell hinzu, indem Sie in der Amazon S3-Konsole zu dem S3-Bucket gehen und die Berechtigung hinzufügen. Um die ACL für `awslogsdelivery` hinzuzufügen, müssen Sie die kanonische ID für das Konto angeben, nämlich:

  ```
  c4c1ede66af53448b93c283ce9448c4ba468c9432aa01d700d3878632f77d2d0
  ```

  Weitere Informationen zum Hinzufügen von ACLs zu S3-Buckets finden Sie unter Konfigurieren von ACLs im Benutzerhandbuch für Amazon Simple Storage Service.

ACL für jede Protokolldatei

Neben der ACL für den Bucket gibt es auch ACLs für jede einzelne Protokolldatei. Der Bucket-Eigentümer verfügt für jede Protokolldatei über die Berechtigung `FULL_CONTROL`, der Eigentümer

der Verteilung (wenn dieser vom Bucket-Eigentümer abweicht) hat keine Berechtigung und das `awslogsdelivery`-Konto verfügt über Lese- und Schreibberechtigungen.

Deaktivieren der Protokollierung

Wenn Sie die Protokollierung deaktivieren, CloudFront werden die ACLs weder für den Bucket noch für die Protokolldateien gelöscht. Sie können die ACLs bei Bedarf löschen.

Erforderliche Schlüsselrichtlinie für Buckets SSE-KMS

Wenn der S3-Bucket für Ihre Standardprotokolle die serverseitige Verschlüsselung mit AWS KMS keys (SSE-KMS) unter Verwendung eines vom Kunden verwalteten Schlüssels verwendet, müssen Sie der Schlüsselrichtlinie für Ihren vom Kunden verwalteten Schlüssel die folgende Anweisung hinzufügen. Dies ermöglicht CloudFront das Schreiben von Protokolldateien in den Bucket. Sie können es nicht SSE-KMS mit dem verwenden Von AWS verwalteter Schlüssel , da CloudFront Sie dann keine Protokolldateien in den Bucket schreiben können.

```
{
    "Sid": "Allow CloudFront to use the key to deliver logs",
    "Effect": "Allow",
    "Principal": {
        "Service": "delivery.logs.amazonaws.com"
    },
    "Action": "kms:GenerateDataKey*",
    "Resource": "*"
}
```

Wenn der S3-Bucket für Ihre Standardprotokolle SSE-KMS mit einem S3-Bucket-Schlüssel verwendet wird, müssen Sie die `kms:Decrypt` Berechtigung auch der Richtlinienerklärung hinzufügen. In diesem Fall sieht die vollständige Richtlinienanweisung wie folgt aus.

```
{
    "Sid": "Allow CloudFront to use the key to deliver logs",
    "Effect": "Allow",
    "Principal": {
        "Service": "delivery.logs.amazonaws.com"
    },
    "Action": [
        "kms:GenerateDataKey*",
        "kms:Decrypt"
    ],
```

```
    "Resource": "*"
}
```

> **Note**
>
> Wenn Sie die Option SSE-KMS für Ihren S3-Bucket aktivieren, geben Sie den vollständigen ARN für den vom Kunden verwalteten Schlüssel an. Weitere Informationen finden Sie unter Angeben der serverseitigen Verschlüsselung mit AWS KMS keys (SSE-KMS) im Amazon Simple Storage Service-Benutzerhandbuch.

Aktivieren der Standardprotokollierung (Legacy)

Verwenden Sie die CloudFront Konsole oder die CloudFront API, um Standardprotokolle zu aktivieren.

Inhalt

- Aktivieren Sie die Standardprotokollierung (Legacy) (CloudFront Konsole)
- Aktivieren Sie die Standardprotokollierung (Legacy) (CloudFront API)

Aktivieren Sie die Standardprotokollierung (Legacy) (CloudFront Konsole)

So aktivieren Sie Standardprotokolle für eine CloudFront Distribution (Konsole)

1. Verwenden Sie die CloudFront Konsole, um eine neue Distribution zu erstellen oder eine bestehende zu aktualisieren.
2. Wählen Sie im Bereich Standardprotokollierung für die Protokollbereitstellung die Option Ein aus.
3. (Optional) Wählen Sie für die Cookie-Protokollierung die Option Ein, wenn Sie Cookies in Ihre Protokolle aufnehmen möchten. Weitere Informationen finden Sie unter Protokollierung von Cookies.

   > **Tip**
   >
   > Die Cookie-Protokollierung ist eine globale Einstellung, die für alle Standardprotokolle Ihrer Distribution gilt. Sie können diese Einstellung nicht für separate Bereitstellungsziele überschreiben.

4. Im Abschnitt Bereitstellen an geben Sie Amazon S3 (Legacy) an.

5. Wählen Sie Ihren Amazon-S3-Bucket aus. Wenn Sie noch keinen haben, können Sie Erstellen auswählen oder die Dokumentation zum Erstellen eines Buckets konsultieren.
6. (Optional) Geben Sie unter Protokollpräfix die Zeichenfolge an, falls vorhanden, die Sie CloudFront den Namen der Zugriffsprotokolldateien für diese Distribution voranstellen möchten, `exampleprefix/` z. B. Der abschließende Schrägstrich (/) ist optional, jedoch empfohlen, um das Durchsuchen Ihrer Protokolldateien zu vereinfachen. Weitere Informationen finden Sie unter Protokollpräfix.
7. Führen Sie die Schritte zum Aktualisieren oder Erstellen Ihrer Distribution aus.
8. Vergewissern Sie sich auf der Seite Protokolle, dass der Status der Standardprotokolle neben der Distribution aktiviert lautet.

   Weitere Informationen zur standardmäßigen Bereitstellung der Protokollierung und den Protokollfeldern finden Sie in der Referenz zur Standardprotokollierung.

Aktivieren Sie die Standardprotokollierung (Legacy) (CloudFront API)

Sie können die CloudFront API auch verwenden, um Standardprotokolle für Ihre Distributionen zu aktivieren.

Um Standardprotokolle für eine Distribution (CloudFront API) zu aktivieren

- Verwenden Sie die Operation CreateDistribution oder UpdateDistribution API und konfigurieren Sie das LoggingConfig Objekt.

Bearbeiten der Standardprotokollierungseinstellungen

Sie können die Protokollierung aktivieren oder deaktivieren, den Amazon S3-Bucket, in dem Ihre Logs gespeichert werden, ändern und das Präfix für Protokolldateien ändern, indem Sie die CloudFront Konsole oder die CloudFront API verwenden. Ihre Änderungen der Protokollierungseinstellungen werden innerhalb von 12 Stunden wirksam.

Weitere Informationen finden Sie unter den folgenden Themen:

- Informationen zum Aktualisieren einer Distribution mithilfe der CloudFront Konsole finden Sie unterEine Verteilung aktualisieren.
- Informationen zum Aktualisieren einer Distribution mithilfe der CloudFront API finden Sie UpdateDistribution in der CloudFront Amazon-API-Referenz.

Senden von Protokollen an Amazon S3

Wenn Sie Ihre Protokolle an Amazon S3 senden, werden sie im folgenden Format angezeigt.

Dateinamenformat

Der Name jeder Protokolldatei, die in Ihrem Amazon S3-Bucket CloudFront gespeichert wird, verwendet das folgende Dateinamenformat:

<optional prefix>/*<distribution ID>*.*YYYY*-*MM*-*DD*-*HH*.*unique-ID*.gz

Datum und Uhrzeit entsprechen der Zeitzone UTC (Coordinated Universal Time).

Wenn Sie beispielsweise `example-prefix` als Präfix verwenden und Ihre Verteilungs-ID `EMLARXS9EXAMPLE` lautet, sehen Ihre Dateinamen folgendermaßen aus:

```
example-prefix/EMLARXS9EXAMPLE.2019-11-14-20.RT4KCN4SGK9.gz
```

Wenn Sie die Protokollierung für eine Verteilung aktivieren, können Sie ein optionales Präfix für den Dateinamen festlegen; so behalten Sie den Überblick darüber, welche Protokolldateien welchen Verteilungen zugeordnet sind. Wenn Sie einen Wert für das Protokolldateipräfix angeben und Ihr Präfix nicht mit einem Schrägstrich (`/`) CloudFront endet, wird automatisch einer angehängt. Wenn Ihr Präfix mit einem Schrägstrich endet, fügen Sie CloudFront keinen weiteren hinzu.

Das `.gz` am Ende des Dateinamens gibt an, dass die Protokolldatei mit gzip komprimiert CloudFront wurde.

Dateiformat des Standardprotokolls

Jeder Eintrag in einer Protokolldatei enthält detaillierte Informationen zu den einzelnen Viewer-Anfragen. Die Protokolldateien weisen folgende Merkmale auf:

- Sie verwenden das erweiterte W3C-Format für Protokolldateien.
- Sie enthalten tabulatorgetrennte Werte.
- Sie enthalten Datensätze in nicht unbedingt chronologischer Reihenfolge.
- Sie enthalten zwei Headerzeilen: eine mit der Version des Dateiformats und eine andere mit den W3C-Feldern für jeden einzelnen Datensatz.
- Enthält URL-encoded Entsprechungen für Leerzeichen und bestimmte andere Zeichen in Feldwerten.

  URL-encoded Entsprechungen werden für die folgenden Zeichen verwendet:

- ASCII-Zeichencodes 0 bis 32, inklusive
- ASCII-Zeichencodes 127 und höher
- Alle Zeichen in der folgenden Tabelle

Der URL-Codierungsstandard ist in RFC 1738 definiert.

URL-Encoded Wert	Zeichen
%3C	<
%3E	>
%22	"
%23	#
%25	%
%7B	{
%7D	}
%7C	|
%5C	\
%5E	^
%7E	~
%5B	[
%5D	]
%60	`
%27	'
%20	Leerzeichen

Löschen von Protokolldateien

CloudFront löscht nicht automatisch Protokolldateien aus Ihrem Amazon S3-Bucket. Informationen zum Löschen von Protokolldateien aus einem Amazon-S3-Bucket finden Sie unter Löschen von Objekten im Konsolen-Benutzerhandbuch für Amazon Simple Storage Service.

Preisgestaltung

Die Standardprotokollierung ist eine optionale Funktion von CloudFront. CloudFront erhebt keine Gebühren für die Aktivierung von Standardprotokollen. Sie müssen jedoch die normalen Amazon-S3-Gebühren für das Speichern und den Zugriff auf die Dateien in Amazon S3 entrichten. Sie können sie jederzeit löschen.

Weitere Informationen zu Preisen finden Sie unter Amazon S3-Preise.

Weitere Informationen zur CloudFront Preisgestaltung finden Sie unter CloudFront Preisgestaltung.

Referenz zur Standardprotokollierung

Die folgenden Abschnitte gelten sowohl für die Standardprotokollierung (v2) als auch für die Standardprotokollierung (Legacy).

Themen

- Zeitrahmen für die Übermittlung von Protokolldateien
- Protokollierung von Anforderungen, wenn Anforderungs-URL oder Header die maximale Größe überschreiten
- Felder der Protokolldatei
- Analysieren von Protokollen

Zeitrahmen für die Übermittlung von Protokolldateien

CloudFront liefert Protokolle für eine Verteilung bis zu mehrmals pro Stunde. Im Allgemeinen enthält eine Protokolldatei Informationen über die Anfragen, die während eines bestimmten Zeitraums CloudFront eingegangen sind. CloudFront übermittelt die Protokolldatei für diesen Zeitraum in der Regel innerhalb einer Stunde nach Auftreten der Ereignisse im Protokoll an Ihr Ziel. Beachten Sie jedoch, dass einige oder auch alle Protokolldateieinträge für einen bestimmten Zeitraum manchmal mit einer Verzögerung von bis zu 24 Stunden übermittelt werden können. Wenn Protokolleinträge verzögert werden, werden sie in einer Protokolldatei CloudFront gespeichert, deren Dateiname das

Datum und die Uhrzeit des Zeitraums enthält, in dem die Anforderungen aufgetreten sind, nicht das Datum und die Uhrzeit der Übermittlung der Datei.

Wenn Sie eine Protokolldatei erstellen, CloudFront werden Informationen für Ihre Verteilung von allen Edge-Standorten zusammengefasst, die während des Zeitraums, den die Protokolldatei abdeckt, Anfragen für Ihre Objekte erhalten haben.

CloudFront kann mehr als eine Datei für einen bestimmten Zeitraum speichern, je nachdem, wie viele Anfragen für CloudFront die mit einer Verteilung verknüpften Objekte eingehen.

CloudFront beginnt etwa vier Stunden, nachdem Sie die Protokollierung aktiviert haben, mit der zuverlässigen Bereitstellung von Zugriffsprotokollen. Möglicherweisen erhalten Sie ein paar Zugriffsprotokolle auch schon vorher.

> **Note**
>
> Wenn während eines bestimmten Zeitraums Ihre Objekte nicht von Benutzern angefordert werden, erhalten Sie keine Protokolldateien für diesen Zeitraum.

Protokollierung von Anforderungen, wenn Anforderungs-URL oder Header die maximale Größe überschreiten

Wenn die Gesamtgröße aller Anfrage-Header, einschließlich Cookies, 20 KB überschreitet oder wenn die URL die Größe von 8.192 Byte überschreitet, kann CloudFront die Anforderung nicht vollständig analysieren und die Anforderung nicht protokollieren. Da die Anfrage nicht protokolliert wird, wird in den Protokolldateien der HTTP-Fehler-Statuscode nicht zurückgegeben.

Wenn der Anfragetext die maximale Größe überschreitet, wird die Anfrage einschließlich des HTTP-Fehlerstatuscodes protokolliert.

Felder der Protokolldatei

Die Protokolldatei für eine Verteilung enthält 33 Felder. Die folgende Liste enthält jeden Feldnamen in der Reihenfolge sowie eine Beschreibung der Informationen in diesem Feld.

1. **`date`**

   Das Datum, an dem das Ereignis aufgetreten ist, im Format `YYYY-MM-DD`. Beispiel, `2019-06-30`. Datum und Uhrzeit entsprechen der Zeitzone UTC (Coordinated Universal Time). Bei WebSocket Verbindungen ist dies das Datum, an dem die Verbindung geschlossen wurde.

2. **`time`**

   Die Uhrzeit, zu der der CloudFront Server die Antwort auf die Anfrage beendet hat (in UTC), `01:42:39` z. B. Bei WebSocket Verbindungen ist dies der Zeitpunkt, zu dem die Verbindung geschlossen wird.

3. **`x-edge-location`**

   Der Edge-Standort, an dem die Anfrage verarbeitet wurde. Jeder Edge-Standort wird anhand eines Codes aus drei Buchstaben und einer willkürlich zugewiesenen Zahl identifiziert, z. B. DFW3. Der Code aus drei Buchstaben entspricht in der Regel dem Code der International Air Transport Association (IATA) für einen Flughafen in der Nähe des Edge-Standorts. (Diese Abkürzungen ändern sich möglicherweise in der Zukunft.)

4. **`sc-bytes`**

   Die Gesamtzahl der Bytes, die der Server als Antwort auf die Anforderung an den Viewer übermittelt hat, einschließlich Headern. Bei WebSocket und gRPC-Verbindungen ist dies die Gesamtzahl der Byte, die vom Server über die Verbindung an den Client gesendet wurden.

5. **`c-ip`**

   Die IP-Adresse des Betrachters, die der Anfrage gestellt hat, z. B. `192.0.2.183` oder `2001:0db8:85a3::8a2e:0370:7334`. Wenn der Viewer einen HTTP-Proxy oder eine Load Balancer verwendet hat, um die Anforderung zu senden, entspricht der Wert dieses Feldes der IP-Adresse des Proxys bzw. des Load Balancers. Siehe auch das Feld `x-forwarded-for`.

6. **`cs-method`**

   Die vom Viewer empfangene HTTP-Anforderungsmethode.

7. **`cs(Host)`**

   Der Domänenname der CloudFront Distribution (z. B. d111111abcdef8.cloudfront.net).

8. **`cs-uri-stem`**

   Der Teil der Anforderungs-URL, der den Pfad und das Objekt identifiziert (z. B, `/images/cat.jpg`). Fragezeichen (?) in URLs und Abfragezeichenfolgen sind nicht im Protokoll enthalten.

9. **`sc-status`**

   Enthält einen der folgenden Werte:

   - Den HTTP-Statuscode der Antwort des Servers (z. B. `200`).

- `000`, was anzeigt, dass der Viewer die Verbindung geschlossen hat, bevor der Server auf die Anforderung antworten konnte. Wenn der Viewer die Verbindung schließt, nachdem der Server mit dem Senden der Antwort begonnen hat, enthält dieses Feld den HTTP-Statuscode der Antwort, mit deren Senden der Server begonnen hatte.

10 **`cs(Referer)`**

Der Wert für den `Referer`-Header in der Anfrage. Der Name der Domäne, von der die Anforderung ausgegangen ist. Häufig vorkommende Referrer sind Suchmaschinen, andere Websites, die direkt auf Ihre Objekte verlinken, und Ihre eigene Website.

11 **`cs(User-Agent)`**

Der Wert für den `User-Agent`-Header in der Anfrage. Der `User-Agent`-Header bezeichnet die Quelle für die Anforderung, z. B. den Gerätetyp und den Browser, der die Anforderung abgesendet hat, oder die Suchmaschine, wenn die Anforderung von einer Suchmaschine stammt.

12 **`cs-uri-query`**

Der Teil der Anforderungs-URL mit der Abfragezeichenfolge, wenn vorhanden.

Wenn eine URL keine Abfragezeichenfolge enthält, ist der Wert dieses Felds ein Bindestrich (-). Weitere Informationen finden Sie unter Zwischenspeichern von Inhalten auf der Grundlage von Abfragezeichenfolgeparametern.

13 **`cs(Cookie)`**

Der `Cookie`-Header in der Anforderung einschließlich der Name-Wert-Paaren und zugehörigen Attributen.

Wenn Sie die Cookie-Protokollierung aktivieren, werden die Cookies in allen Anfragen CloudFront protokolliert, unabhängig davon, welche Cookies Sie an den Ursprung weiterleiten möchten. Wenn eine Anforderung keinen Cookie-Header enthält, ist der Wert dieses Felds ein Bindestrich (-). Weitere Informationen zu Cookies finden Sie unter Zwischenspeichern von Inhalten auf der Grundlage von Cookies.

14 **`x-edge-result-type`**

Art der Klassifizierung der Antwort durch den Server, nachdem das letzte Byte den Server verlassen hat. Manchmal wird der Ergebnistyp zwischen dem Zeitpunkt, an dem Server zum Senden der Antwort bereit ist, und dem Zeitpunkt, an dem er das Senden der Antwort abgeschlossen hat, geändert. Siehe auch das Feld `x-edge-response-result-type`.

Angenommen, im HTTP-Streaming findet der Server ein Segment des Streams im Cache. In diesem Szenario würde der Wert dieses Feldes normalerweise sei `Hit`. Wenn der Viewer jedoch die Verbindung schließt, bevor der Server das ganze Segment übermittelt hat, ist der endgültige Ergebnistyp (und damit der Wert dieses Felds) `Error`.

WebSocket und gRPC-Verbindungen haben `Miss` für dieses Feld den Wert von, da der Inhalt nicht zwischenspeicherbar ist und direkt an den Ursprung weitergeleitet wird.

Mögliche Werte sind:

- `Hit` – Der Server hat das Objekt aus dem Cache für den Betrachter bereitgestellt.
- `RefreshHit` – Der Server hat das Objekt im Edge-Zwischenspeicher gefunden, es war jedoch abgelaufen. Daher nahm der Server Kontakt mit dem Ursprung auf, um zu überprüfen, ob der Zwischenspeicher die neueste Version des Objekts enthalten hatte.
- `Miss` – Die Anforderung konnte nicht durch ein Objekt im Zwischenspeicher bedient werden. Daher hat der Server die Anforderung an den Ursprung weitergeleitet und das Ergebnis an den Betrachter ausgegeben.
- `LimitExceeded`— Die Anfrage wurde abgelehnt, weil ein CloudFront Kontingent (früher als Limit bezeichnet) überschritten wurde.
- `CapacityExceeded` – Der Server hat den HTTP-Statuscode 503 zurückgegeben, da er zum Zeitpunkt der Anforderung nicht über genügend Kapazitäten für die Bereitstellung des Objekts verfügte.
- `Error` – In der Regel bedeutet dies, dass die Anforderung zu einem Client-Fehler (d. h. der Wert des Felds `sc-status` liegt im `4xx`-Bereich) oder zu einem Serverfehler geführt hat (d. h. der Wert des Felds `sc-status` liegt im `5xx`-Bereich). Wenn der Wert des Felds `sc-status` `200` oder wenn der Wert dieses Felds `Error` ist und der Wert des Felds `x-edge-response-result-type` nicht `Error` ist, war die HTTP-Anforderung erfolgreich. Die Client-Verbindung wurde jedoch getrennt, bevor alle Bytes empfangen wurden.
- `Redirect` – Der Server hat den Betrachter entsprechend den Verteilungseinstellungen von HTTP zu HTTPS umgeleitet.
- `LambdaExecutionError`— Die mit der Distribution verknüpfte Lambda @Edge -Funktion wurde aufgrund einer fehlerhaften Zuordnung, eines Funktions-Timeouts, eines AWS Abhängigkeitsproblems oder eines anderen allgemeinen Verfügbarkeitsproblems nicht abgeschlossen.

15**`x-edge-request-id`**

Eine undurchsichtige Zeichenfolge, die eine Anfrage eindeutig identifiziert. CloudFront sendet diese Zeichenfolge auch im `x-amz-cf-id` Antwortheader.

16 **`x-host-header`**

Der Wert, den der Viewer in den `Host`-Header der Anforderung eingefügt hat. Wenn Sie den CloudFront Domainnamen in Ihren Objekt-URLs verwenden (z. B. d111111abcdef8.cloudfront.net), enthält dieses Feld diesen Domainnamen. Wenn Sie alternative Domänennamen (CNAMEs) in Ihren Objekt-URLs verwenden (z. B. www.example.com), enthält dieses Feld den alternativen Domänennamen.

Wenn Sie alternative Domain-Namen verwenden, finden Sie unter `cs(Host)` in Feld 7 den Namen der Domain, die Ihrer Verteilung zugewiesen ist.

17 **`cs-protocol`**

Das Protokoll der Viewer-Anforderung (`http`, `https`, `grpcs`, `ws` oder `wss`).

18 **`cs-bytes`**

Die Gesamtzahl der Bytes an Daten, die in der Anforderung des Viewers einschließlich Headern enthalten sind. Bei WebSocket und gRPC-Verbindungen ist dies die Gesamtzahl der Byte, die über die Verbindung vom Client an den Server gesendet wurden.

19 **`time-taken`**

Die Anzahl der Sekunden (auf die Tausendstelsekunde genau, z. B. 0,082) ab dem Zeitpunkt, an dem der Server die Anforderung des Viewers empfängt, bis zu dem Zeitpunkt, an dem der Server das letzte Byte der Antwort in die Ausgabewarteschlange schreibt, wie auf dem Server gemessen. Aus der Perspektive der Viewers vergeht bis zum Empfang der gesamten Antwort aufgrund der Netzwerklatenz und des TCP-Puffervorgangs insgesamt mehr Zeit, als mit diesem Wert angegeben wird.

20 **`x-forwarded-for`**

Wenn der Viewer einen HTTP-Proxy oder Load Balancer verwendet hat, um die Anforderung zu senden, entspricht der Wert des Felds `c-ip` der IP-Adresse des Proxys oder Load Balancers. In diesem Fall stellt dieses Feld die IP-Adresse des Viewers dar, von dem die Anfrage stammt. Dieses Feld kann mehrere durch Kommas getrennte IP-Adressen enthalten. Jede IP-Adresse kann eine IPv4-Adresse (z. B. `192.0.2.183`) oder eine IPv6-Adresse (z. B. `2001:0db8:85a3::8a2e:0370:7334`) sein.

Wenn der Viewer keinen HTTP-Proxy oder Load Balancer verwendet hat, ist der Wert dieses Feldes ein Bindestrich (-).

21 **`ssl-protocol`**

Wenn für die Anfrage HTTPS verwendet wurde, enthält dieses Feld das SSL/TLS Protokoll, das der Viewer und der Server für die Übertragung der Anfrage und Antwort ausgehandelt haben. Eine Liste der möglichen Werte finden Sie unter den unterstützten SSL/TLS ProtokollenUnterstützte Protokolle und Chiffren zwischen Viewern und CloudFront.

Wenn als `cs-protocol` in Feld 17 `http` angegeben ist, ist der Wert für dieses Feld ein Bindestrich (-).

22 **`ssl-cipher`**

Wenn für die Anfrage HTTPS verwendet wurde, enthält dieses Feld die SSL/TLS Chiffre, die der Viewer und der Server für die Verschlüsselung der Anfrage und Antwort ausgehandelt haben. Eine Liste der möglichen Werte finden Sie unter den unterstützten SSL/TLS Chiffren. Unterstützte Protokolle und Chiffren zwischen Viewern und CloudFront

Wenn als `cs-protocol` in Feld 17 `http` angegeben ist, ist der Wert für dieses Feld ein Bindestrich (-).

23 **`x-edge-response-result-type`**

Die Art, wie der Server die Antwort direkt vor der Rücksendung der Antwort an den Viewer klassifiziert hat. Siehe auch das Feld `x-edge-result-type`. Mögliche Werte sind:

- `Hit` – Der Server hat das Objekt aus dem Cache für den Betrachter bereitgestellt.
- `RefreshHit` – Der Server hat das Objekt im Edge-Zwischenspeicher gefunden, es war jedoch abgelaufen. Daher nahm der Server Kontakt mit dem Ursprung auf, um zu überprüfen, ob der Zwischenspeicher die neueste Version des Objekts enthalten hatte.
- `Miss` – Die Anforderung konnte nicht durch ein Objekt im Zwischenspeicher bedient werden. Daher hat der Server die Anforderung an den Ursprungs-Server weitergeleitet und das Ergebnis an den Betrachter ausgegeben.
- `LimitExceeded`— Die Anfrage wurde abgelehnt, weil ein CloudFront Kontingent (früher als Limit bezeichnet) überschritten wurde.
- `CapacityExceeded` – Der Server hat den Fehler 503 zurückgegeben, da er zum Zeitpunkt der Anforderung nicht über genügend Kapazitäten verfügte, um das Objekt bereitzustellen.

- `Error` – In der Regel bedeutet dies, dass die Anforderung zu einem Client-Fehler (d. h. der Wert des Felds `sc-status` liegt im `4xx`-Bereich) oder zu einem Serverfehler geführt hat (d. h. der Wert des Felds `sc-status` liegt im `5xx`-Bereich).

  Wenn der Wert des Felds `x-edge-result-type` `Error` ist und der Wert dieses Felds nicht `Error` ist, wurde die Verbindung vor dem Abschluss des Downloads durch den Client getrennt.
- `Redirect` – Der Server hat den Betrachter entsprechend den Verteilungseinstellungen von HTTP zu HTTPS umgeleitet.
- `LambdaExecutionError`— Die mit der Distribution verknüpfte Lambda @Edge -Funktion wurde aufgrund einer fehlerhaften Zuordnung, eines Funktions-Timeouts, eines AWS Abhängigkeitsproblems oder eines anderen allgemeinen Verfügbarkeitsproblems nicht abgeschlossen.

24. **`cs-protocol-version`**

    Die HTTP-Version, die der Viewer in der Anfrage angegeben hat: Mögliche Werte sind `HTTP/0.9`, `HTTP/1.0`, `HTTP/1.1`, `HTTP/2.0` und `HTTP/3.0`.

25. **`fle-status`**

    Wenn die Verschlüsselung auf Feldebene für eine Verteilung konfiguriert ist, enthält dieses Feld einen Code, der angibt, ob der Anforderungstext erfolgreich verarbeitet wurde. Wenn der Server den Anforderungstext erfolgreich verarbeitet, Werte in den angegebenen Feldern verschlüsselt und die Anforderung an den Ursprung weiterleitet, ist der Wert dieses Felds `Processed`. Der Wert von `x-edge-result-type` kann in diesem Fall immer noch auf einen clientseitigen oder serverseitigen Fehler hinweisen.

    Mögliche Werte für dieses Feld sind:

    - `ForwardedByContentType` – Der Server hat die Anforderung ohne Parsing oder Verschlüsselung an den Ursprung weitergeleitet, da kein Content-Typ konfiguriert wurde.
    - `ForwardedByQueryArgs` – Der Server hat die Anforderung ohne Parsing oder Verschlüsselung an den Ursprung weitergeleitet, da die Anforderung ein Abfrageargument enthält, das nicht in der Konfiguration für die Verschlüsselung auf Feldebene enthalten war.
    - `ForwardedDueToNoProfile` – Der Server hat die Anforderung ohne Parsing oder Verschlüsselung an den Ursprung weitergeleitet, da in der Konfiguration für die Verschlüsselung auf Feldebene kein Profil angegeben wurde.

- `MalformedContentTypeClientError` – Der Server hat die Anforderung zurückgewiesen und einen HTTP-400-Statuscode an den Betrachter zurückgegeben, da der Wert des `Content-Type`-Headers ein ungültiges Format hatte.
- `MalformedInputClientError` – Der Server hat die Anforderung zurückgewiesen und einen HTTP 400-Statuscode an den Betrachter zurückgegeben, da der Anforderungstext ein ungültiges Format hatte.
- `MalformedQueryArgsClientError` – Der Server hat die Anforderung zurückgewiesen und einen HTTP-400-Statuscode an den Betrachter zurückgegeben, weil ein Abfrageargument leer war oder ein ungültiges Format hatte.
- `RejectedByContentType` – Der Server hat die Anforderung zurückgewiesen und einen HTTP-400-Statuscode an den Betrachter zurückgegeben, da in der Konfiguration für die Verschlüsselung auf Feldebene kein Content-Typ angegeben wurde.
- `RejectedByQueryArgs` – Der Server hat die Anforderung zurückgewiesen und einen HTTP-400-Statuscode an den Betrachter zurückgegeben, da in der Konfiguration für die Verschlüsselung auf Feldebene kein Abfrageargument angegeben wurde.
- `ServerError` – Der Ursprungs-Server hat einen Fehler zurückgegeben.

Wenn die Anforderung das Kontingent für die Verschlüsselung auf Feldebene überschreitet (früher als Limit bezeichnet), enthält dieses Feld einen der folgenden Fehlercodes und der Server gibt dem Viewer den HTTP-Statuscode 400 zurück. Eine Liste der aktuellen Kontingente für die Verschlüsselung auf Feldebene finden Sie unter Kontingente für Verschlüsselung auf Feldebene.

- `FieldLengthLimitClientError` – Ein Feld, das für die Verschlüsselung konfiguriert ist, hat die maximal zulässige Länge überschritten.
- `FieldNumberLimitClientError` – Eine Anforderung, die die Verteilung verschlüsseln soll, enthält mehr Felder als zulässig.
- `RequestLengthLimitClientError` – Die Länge des Anfragetexts hat die maximal zulässige Länge, wenn die Verschlüsselung auf Feldebene konfiguriert ist, überschritten.

Wenn die Verschlüsselung auf Feldebene nicht für die Verteilung konfiguriert ist, ist der Wert dieses Feldes ein Bindestrich (-).

26. **`fle-encrypted-fields`**

Die Anzahl der Verschlüsselungsfelder auf Feldebene, die der Server verschlüsselt und an den Ursprung weitergeleitet hat. CloudFront Server streamen die verarbeitete Anfrage an den

Ursprung, während sie Daten verschlüsseln, sodass dieses Feld einen Wert haben kann, auch wenn der Wert von ein Fehler `fle-status` ist.

Wenn die Verschlüsselung auf Feldebene nicht für die Verteilung konfiguriert ist, ist der Wert dieses Feldes ein Bindestrich (-).

27. **`c-port`**

Die Portnummer der Anforderung des Viewers.

28. **`time-to-first-byte`**

Die Anzahl der Sekunden zwischen dem Empfangen der Anforderung und dem Schreiben des ersten Bytes der Antwort, gemessen auf dem Server.

29. **`x-edge-detailed-result-type`**

Dieses Feld enthält den gleichen Wert wie der Wert für das `x-edge-result-type`-Feld, außer in den folgenden Fällen:

- Wenn das Objekt dem Viewer aus der Origin-Shield-Ebene bereitgestellt wurde, enthält dieses Feld `OriginShieldHit`.
- Wenn sich das Objekt nicht im CloudFront Cache befand und die Antwort von einer Lambda @Edge -Funktion der Ursprungsanfrage generiert wurde, enthält dieses Feld Folgendes: `MissGeneratedResponse`
- Wenn der Wert des `x-edge-result-type`-Felds `Error` lautet, enthält dieses Feld einen der folgenden Werte mit weiteren Informationen zum Fehler:
  - `AbortedOrigin` – Der Server hat ein Problem mit dem Ursprung festgestellt.
  - `ClientCommError` – Die Antwort auf den Betrachter wurde aufgrund eines Kommunikationsproblems zwischen dem Server und dem Betrachter unterbrochen.
  - `ClientGeoBlocked` – Die Verteilung ist so konfiguriert, dass Anforderungen vom geografischen Standort des Viewers abgelehnt werden.
  - `ClientHungUpRequest` — Der Betrachter wurde beim Senden der Anfrage vorzeitig gestoppt.
  - `Error` – Es ist ein Fehler aufgetreten, dessen Fehlertyp zu keiner der anderen Kategorien passt. Dieser Fehlertyp kann auftreten, wenn der Server eine Fehlerantwort aus dem Cache ausgibt.

- `InternalError`— Ein interner CloudFront Fehler ist aufgetreten. Dies deutet in der Regel darauf hin, dass der Fehler von der CloudFront Infrastruktur herrührt, nicht vom Ursprung oder vom Betrachter.
- `InvalidRequest` – Der Server hat eine ungültige Anforderung vom Betrachter erhalten.
- `InvalidRequestBlocked` — Der Zugriff auf die angeforderte Ressource ist blockiert.
- `InvalidRequestCertificate`— Die Verteilung entspricht nicht dem SSL/TLS Zertifikat, für das die HTTPS-Verbindung hergestellt wurde.
- `InvalidRequestHeader` – Die Anforderung enthielt einen ungültigen Header.
- `InvalidRequestMethod` — Die Verteilung ist nicht für eine Verarbeitung der verwendeten HTTP-Anforderungsmethode konfiguriert. Dies kann vorkommen, wenn die Verteilung nur Anforderungen unterstützt, die zwischengespeichert werden können.
- `LambdaLimitExceeded`— Die Anfrage hat ein Lambda @Edge -Funktionslimit überschritten.
- `OriginCommError` – Bei der Anforderung ist eine Zeitüberschreitung aufgetreten, während eine Verbindung mit dem Ursprung hergestellt wurde oder Daten aus dem Ursprung gelesen wurden.
- `OriginConnectError` – Der Server konnte keine Verbindung zum Ursprung herstellen.
- `OriginContentRangeLengthError` – Der `Content-Length`-Header in der Antwort des Ursprungs stimmt nicht mit der Länge im `Content-Range`-Header überein.
- `OriginDnsError` – Der Server konnte den Domänennamen des Ursprungs nicht auflösen.
- `OriginError` — Der Ursprung gab eine falsche Antwort zurück.
- `OriginHeaderTooBigError` – Ein vom Ursprung zurückgegebener Header ist für eine Verarbeitung durch den Edge-Server zu groß.
- `OriginInvalidResponseError` — Der Ursprung gab eine ungültige Antwort zurück.
- `OriginReadError` – Der Server konnte nicht vom Ursprung lesen.
- `OriginWriteError` – Der Server konnte nicht in den Ursprung schreiben.
- `OriginZeroSizeObjectError` — Ein Nullgrößenobjekt, das vom Ursprung gesendet wurde, führte zu einem Fehler.
- `SlowReaderOriginError` — Der Betrachter hat die Nachricht, die den Ursprungsfehler verursacht hat, nur langsam gelesen.

30**`sc-content-type`**

Der Wert des HTTP `Content-Type`-Headers der Antwort.

31**`sc-content-len`**

Der Wert des HTTP `Content-Length`-Headers der Antwort.

32 **`sc-range-start`**

Wenn die Antwort den HTTP `Content-Range`-Header enthält, enthält dieses Feld den Bereichsstartwert.

33 **`sc-range-end`**

Wenn die Antwort den HTTP `Content-Range`-Header enthält, enthält dieses Feld den Bereichsendwert.

34 **`distribution-tenant-id`**

Die ID des Distributionsmandanten.

35 **`connection-id`**

Ein eindeutiger Bezeichner für die TLS-Verbindung.

Sie müssen mTLS für Ihre Distributionen aktivieren, bevor Sie Informationen für dieses Feld abrufen können. Weitere Informationen finden Sie unter Gegenseitige TLS-Authentifizierung mit CloudFront (Viewer mTLS).

Es folgt eine Beispielprotokolldatei für eine Distribution.

```
#Version: 1.0
#Fields: date time x-edge-location sc-bytes c-ip cs-method cs(Host) cs-uri-stem sc-
status cs(Referer) cs(User-Agent) cs-uri-query cs(Cookie) x-edge-result-type x-edge-
request-id x-host-header cs-protocol cs-bytes time-taken x-forwarded-for ssl-protocol
 ssl-cipher x-edge-response-result-type cs-protocol-version fle-status fle-encrypted-
fields c-port time-to-first-byte x-edge-detailed-result-type sc-content-type sc-
content-len sc-range-start sc-range-end
2019-12-04 21:02:31 LAX1 392 192.0.2.100 GET d111111abcdef8.cloudfront.net /
index.html 200 - Mozilla/5.0%20(Windows%20NT%2010.0;%20Win64;
%20x64)%20AppleWebKit/537.36%20(KHTML,%20like
%20Gecko)%20Chrome/78.0.3904.108%20Safari/537.36 - - Hit
 SOX4xwn4XV6Q4rgb7XiVGOHms_BGlTAC4KyHmureZmBNrjGdRLiNIQ== d111111abcdef8.cloudfront.net
 https 23 0.001 - TLSv1.2 ECDHE-RSA-AES128-GCM-SHA256 Hit HTTP/2.0 - - 11040 0.001 Hit
 text/html 78 - -
2019-12-04 21:02:31 LAX1 392 192.0.2.100 GET d111111abcdef8.cloudfront.net /
index.html 200 - Mozilla/5.0%20(Windows%20NT%2010.0;%20Win64;
%20x64)%20AppleWebKit/537.36%20(KHTML,%20like
```

```
%20Gecko)%20Chrome/78.0.3904.108%20Safari/537.36 - - Hit
 k6WGMNkEzR5BEM_SaF47gjtX9zBDO2m349OY2an0QPEaUum1ZOLrow== d111111abcdef8.cloudfront.net
 https 23 0.000 - TLSv1.2 ECDHE-RSA-AES128-GCM-SHA256 Hit HTTP/2.0 - - 11040 0.000 Hit
 text/html 78 - -
2019-12-04 21:02:31 LAX1 392 192.0.2.100 GET d111111abcdef8.cloudfront.net /
index.html 200 - Mozilla/5.0%20(Windows%20NT%2010.0;%20Win64;
%20x64)%20AppleWebKit/537.36%20(KHTML,%20like
%20Gecko)%20Chrome/78.0.3904.108%20Safari/537.36 - - Hit
 f37nTMVvnKvV2ZSvEsivup_c2kZ7VXzYdjC-GUQZ5qNs-89BlWazbw== d111111abcdef8.cloudfront.net
 https 23 0.001 - TLSv1.2 ECDHE-RSA-AES128-GCM-SHA256 Hit HTTP/2.0 - - 11040 0.001 Hit
 text/html 78 - -
2019-12-13 22:36:27 SEA19-C1 900 192.0.2.200 GET d111111abcdef8.cloudfront.net /
favicon.ico 502 http://www.example.com/ Mozilla/5.0%20(Windows
%20NT%2010.0;%20Win64;%20x64)%20AppleWebKit/537.36%20(KHTML,
%20like%20Gecko)%20Chrome/78.0.3904.108%20Safari/537.36 - - Error
 1pkpNfBQ39sYMnjjUQjmH2w1wdJnbHYTbag21o_3OfcQgPzdL2RSSQ== www.example.com http 675
 0.102 - - - Error HTTP/1.1 - - 25260 0.102 OriginDnsError text/html 507 - -
2019-12-13 22:36:26 SEA19-C1 900 192.0.2.200 GET d111111abcdef8.cloudfront.net / 502
 - Mozilla/5.0%20(Windows%20NT%2010.0;%20Win64;%20x64)%20AppleWebKit/537.36%20(KHTML,
%20like%20Gecko)%20Chrome/78.0.3904.108%20Safari/537.36 - - Error
 3AqrZGCnF_g0-5KOvfA7c9XLcf4YGvMFSeFdIetR1N_2y8jSis8Zxg== www.example.com http 735
 0.107 - - - Error HTTP/1.1 - - 3802 0.107 OriginDnsError text/html 507 - -
2019-12-13 22:37:02 SEA19-C2 900 192.0.2.200 GET d111111abcdef8.cloudfront.net / 502
 - curl/7.55.1 - - Error kBkDzGnceVtWHqSCqBUqtA_cEs2T3tFUBbnBNkB9El_uVRhHgcZfcw==
 www.example.com http 387 0.103 - - - Error HTTP/1.1 - - 12644 0.103 OriginDnsError
 text/html 507 - -
```

Analysieren von Protokollen

Da Sie mehrere Zugriffsprotokolle pro Stunde erhalten können, empfehlen wir, alle für einen bestimmten Zeitraum erhaltenen Protokolldateien in einer Datei zusammenzufassen. Sie können die Daten für diesen Zeitraum dann genauer und vollständig analysieren.

Eine Möglichkeit, Ihre Zugriffsprotokolle zu analysieren, ist die Verwendung von Amazon Athena;. Athena ist ein interaktiver Abfragedienst, mit dem Sie Daten für AWS Dienste analysieren können, darunter. CloudFront Weitere Informationen finden Sie unter Abfragen von CloudFront Amazon-Protokollen im Amazon Athena-Benutzerhandbuch.

Darüber hinaus werden in den folgenden AWS Blogbeiträgen einige Möglichkeiten zur Analyse von Zugriffsprotokollen erörtert.

- Amazon CloudFront Request Logging (für Inhalte, die über HTTP bereitgestellt werden)

- Verbesserte CloudFront Protokolle, jetzt mit Abfragezeichenfolgen

Verwenden Sie Zugriffsprotokolle in Echtzeit

Mit CloudFront Echtzeit-Zugriffsprotokollen können Sie Informationen über Anfragen an eine Distribution in Echtzeit abrufen (Protokolle werden innerhalb von Sekunden nach Eingang der Anfragen zugestellt). Mithilfe von Zugriffsprotokollen in Echtzeit können Sie die Leistung der Inhaltsbereitstellung überwachen, analysieren und Maßnahmen ergreifen.

CloudFront Echtzeit-Zugriffsprotokolle sind konfigurierbar. Sie können wählen:

- Die Stichprobenrate für Ihre Echtzeitprotokolle, d. h. der Prozentsatz der Anfragen, für die Sie Zugriffsprotokolldatensätze in Echtzeit erhalten möchten.
- Die spezifischen Felder, die Sie in den Protokolldatensätzen empfangen möchten.
- Das spezifische Cache-Verhalten (Pfadmuster), für das Sie Echtzeit-Protokolle erhalten möchten.

CloudFront Echtzeit-Zugriffsprotokolle werden an den Datenstream Ihrer Wahl in Amazon Kinesis Data Streams übermittelt. Sie können Ihren eigenen Kinesis Data Stream Consumer erstellen oder Amazon Data Firehose verwenden, um die Protokolldaten an Amazon Simple Storage Service (Amazon S3), Amazon Redshift, Amazon OpenSearch Service (Service) oder einen OpenSearch Protokollverarbeitungsservice eines Drittanbieters zu senden.

CloudFront Gebühren für Echtzeit-Zugriffsprotokolle, zusätzlich zu den Gebühren, die Ihnen für die Nutzung von Kinesis Data Streams entstehen. Weitere Informationen zu den Preisen finden Sie unter CloudFront Amazon-Preise und Amazon Kinesis Data Streams-Preise.

> **Important**
>
> Wir empfehlen Ihnen, die Protokolle zu verwenden, um die Art der Anfragen nach Ihren Inhalten zu verstehen, und nicht, um alle Anfragen vollständig zu erfassen. CloudFront liefert Zugriffsprotokolle in Echtzeit nach bestem Wissen und Gewissen. Der Protokolleintrag für eine bestimmte Anfrage wird möglicherweise viel später übermittelt, als die Anfrage tatsächlich verarbeitet wurde; in seltenen Fällen kann es auch sein, dass ein Protokolleintrag gar nicht übermittelt wird. Wenn ein Protokolleintrag in den Echtzeit-Zugriffsprotokollen weggelassen wird, entspricht die Anzahl der Einträge in den Echtzeit-Zugriffsprotokollen nicht der Nutzung, die in den AWS Abrechnungs- und Nutzungsberichten angegeben ist.

Themen

- Erstellen und verwenden Sie Konfigurationen für Echtzeit-Zugriffsprotokolle
- Machen Sie sich mit den Konfigurationen der Echtzeit-Zugriffsprotokolle vertraut
- Erstellen eines Kinesis-Datenstromverbrauchers
- Beheben Sie Probleme mit Zugriffsprotokollen in Echtzeit

Erstellen und verwenden Sie Konfigurationen für Echtzeit-Zugriffsprotokolle

Um Informationen über Anfragen an eine Distribution in Echtzeit zu erhalten, können Sie Zugriffs-Log-Konfigurationen in Echtzeit verwenden. Die Protokolle werden innerhalb von Sekunden nach Eingang der Anforderungen zugestellt. Sie können eine Konfiguration für das Echtzeit-Zugriffsprotokoll in der CloudFront Konsole, mit AWS Command Line Interface (AWS CLI) oder mit der CloudFront API erstellen.

Um eine Konfiguration für das Echtzeit-Zugriffsprotokoll zu verwenden, fügen Sie sie einem oder mehreren Cache-Verhalten in einer CloudFront Distribution hinzu.

Console

Um eine Konfiguration für das Echtzeit-Zugriffsprotokoll zu erstellen

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Seite Logs in der CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home?#/logs.
2. Wählen Sie die Registerkarte „Real-time Konfigurationen“.
3. Wählen Sie Create configuration (Konfiguration erstellen).
4. Geben Sie unter Name einen Namen für die Konfiguration ein.
5. Geben Sie unter Abtastrate den Prozentsatz der Anforderungen ein, für die Sie Echtzeit-Protokolldatensätze erhalten möchten.
6. Wählen Sie unter Felder die Felder aus, die Sie in den Echtzeit-Zugriffsprotokollen erhalten möchten.
   - Um alle CMCD-Felder in Ihre Protokolle aufzunehmen, wählen Sie CMCD alle Schlüssel aus.
7. Wählen Sie für Endpoint einen oder mehrere Kinesis-Datenstreams aus, um Zugriffsprotokolle in Echtzeit zu erhalten.

> **Note**
>
> CloudFront Echtzeit-Zugriffsprotokolle werden an den Datenstream übermittelt, den Sie in Kinesis Data Streams angeben. Um Ihre Echtzeit-Zugriffsprotokolle zu lesen und zu analysieren, können Sie Ihren eigenen Kinesis Data Stream Consumer erstellen. Sie können Firehose auch verwenden, um die Protokolldaten an Amazon S3, Amazon Redshift, Amazon OpenSearch Service oder einen Protokollverarbeitungsdienst eines Drittanbieters zu senden.

8. Wählen Sie unter IAM-Rolle die Option Neue Servicerolle erstellen oder eine vorhandene Rolle aus. Sie müssen über die Berechtigung zum Erstellen von IAM-Rollen verfügen.
9. (Optional) Wählen Sie für CloudFrontVerteilung ein Verteilungs- und Cache-Verhalten aus, das an die Konfiguration des Echtzeit-Zugriffsprotokolls angehängt werden soll.
10. Wählen Sie Create configuration (Konfiguration erstellen).

Bei Erfolg zeigt die Konsole die Details der Konfiguration des Echtzeit-Zugriffsprotokolls an, die Sie gerade erstellt haben.

Weitere Informationen finden Sie unter Machen Sie sich mit den Konfigurationen der Echtzeit-Zugriffsprotokolle vertraut.

AWS CLI

Verwenden Sie den aws cloudfront create-realtime-log-config Befehl, um eine Konfiguration für das AWS CLI Echtzeit-Zugriffsprotokoll mit dem zu erstellen. Sie können die Eingabeparameter des Befehls in einer Eingabedatei bereitstellen, anstatt jeden einzelnen Parameter als Befehlszeileneingabe anzugeben.

Um eine Echtzeit-Zugriffsprotokollkonfiguration zu erstellen (CLI mit Eingabedatei)

1. Verwenden Sie den folgenden Befehl, um eine Datei mit dem Namen `rtl-config.yaml` zu erstellen, die alle Eingabeparameter für den create-realtime-log-config-Befehl enthält.

```
aws cloudfront create-realtime-log-config --generate-cli-skeleton yaml-input >
 rtl-config.yaml
```

2. Öffnen Sie die Datei mit dem Namen `rtl-config.yaml`, die Sie gerade erstellt haben. Bearbeiten Sie die Datei, um die gewünschten Konfigurationseinstellungen für das Echtzeit-Zugriffsprotokoll anzugeben, und speichern Sie dann die Datei. Beachten Sie Folgendes:

   - Für `StreamType` ist der einzige gültige Wert `Kinesis`.

   Weitere Informationen zu den Echtzeit-Konfigurationseinstellungen im Langformat finden Sie unter Machen Sie sich mit den Konfigurationen der Echtzeit-Zugriffsprotokolle vertraut.

3. Verwenden Sie den folgenden Befehl, um die Konfiguration des Echtzeit-Zugriffsprotokolls mithilfe von Eingabeparametern aus der `rtl-config.yaml` Datei zu erstellen.

```
aws cloudfront create-realtime-log-config --cli-input-yaml file://rtl-
config.yaml
```

Bei Erfolg werden in der Ausgabe des Befehls die Details der Konfiguration des Echtzeit-Zugriffsprotokolls angezeigt, die Sie gerade erstellt haben.

Um eine Echtzeit-Zugriffsprotokollkonfiguration an eine bestehende Distribution anzuhängen (CLI mit Eingabedatei)

1. Verwenden Sie den folgenden Befehl, um die Verteilungskonfiguration für die CloudFront Distribution zu speichern, die Sie aktualisieren möchten. Ersetzen Sie *`distribution_ID`* durch die ID der Verteilung.

```
aws cloudfront get-distribution-config --id distribution_ID --output yaml >
 dist-config.yaml
```

2. Öffnen Sie die Datei mit dem Namen `dist-config.yaml`, die Sie gerade erstellt haben. Bearbeiten Sie die Datei und nehmen Sie die folgenden Änderungen an jedem Cache-Verhalten vor, das Sie aktualisieren, um eine Konfiguration für das Echtzeit-Zugriffsprotokoll zu verwenden.

   - Fügen Sie in der Cache-Verhaltensweise ein Feld mit dem Namen `RealtimeLogConfigArn` hinzu. Verwenden Sie für den Feldwert den ARN der Echtzeit-Zugriffs-Log-Konfiguration, die Sie an dieses Cache-Verhalten anhängen möchten.

- Benennen Sie das Feld `ETag` in `IfMatch` um, ändern Sie jedoch nicht den Wert des Feldes.

Speichern Sie die Datei, wenn Sie fertig sind.

3. Verwenden Sie den folgenden Befehl, um die Verteilung so zu aktualisieren, dass sie die Konfiguration des Echtzeit-Zugriffsprotokolls verwendet. Ersetzen Sie *`distribution_ID`* durch die ID der Verteilung.

```
aws cloudfront update-distribution --id distribution_ID --cli-input-yaml file://
dist-config.yaml
```

Wenn dies erfolgreich ist, zeigt die Ausgabe des Befehls die Details der gerade aktualisierten Verteilung an.

API

Verwenden Sie den CloudFront API-Vorgang, um eine Konfiguration für das Echtzeit-Zugriffsprotokoll mit der CreateRealtimeLogConfigAPI zu erstellen. Weitere Informationen zu den Parametern, die Sie in diesem API-Aufruf angeben, finden Sie in Machen Sie sich mit den Konfigurationen der Echtzeit-Zugriffsprotokolle vertraut und in der API-Referenzdokumentation für Ihr AWS SDK oder einen anderen API-Client.

Nachdem Sie eine Konfiguration für das Echtzeit-Zugriffsprotokoll erstellt haben, können Sie sie mithilfe einer der folgenden API-Operationen an ein Cache-Verhalten anhängen:

- Um es an ein Cache-Verhalten in einer vorhandenen Distribution anzuhängen, verwenden Sie UpdateDistribution.
- Um es an ein Cache-Verhalten in einer neuen Distribution anzuhängen, verwenden Sie CreateDistribution.

Geben Sie für diese beiden API-Operationen den ARN der Echtzeit-Zugriffsprotokollkonfiguration vor `RealtimeLogConfigArn` Ort innerhalb eines Cache-Verhaltens an. Weitere Informationen zu den anderen Feldern, die Sie in diesen API-Aufrufen angeben, finden Sie in Referenz für alle Distributionseinstellungen und in der API-Referenzdokumentation für Ihr AWS SDK oder einen anderen API-Client.

Machen Sie sich mit den Konfigurationen der Echtzeit-Zugriffsprotokolle vertraut

Um CloudFront Echtzeit-Zugriffsprotokolle zu verwenden, müssen Sie zunächst eine Konfiguration für Echtzeit-Zugriffsprotokolle erstellen. Die Konfiguration des Echtzeit-Zugriffsprotokolls enthält Informationen darüber, welche Protokollfelder Sie empfangen möchten, die Samplingrate für Protokolldatensätze und den Kinesis-Datenstream, an den Sie die Protokolle liefern möchten.

Insbesondere enthält eine Konfiguration für das Echtzeit-Zugriffsprotokoll die folgenden Einstellungen:

Inhalt

- Name
- Abtastrate
- Felder
- Endpunkt (Kinesis Data Streams)
- IAM role (IAM-Rolle)

Name

Ein Name zur Identifizierung der Konfiguration des Echtzeit-Zugriffsprotokolls.

Abtastrate

Die Abtastrate ist eine ganze Zahl zwischen 1 und 100 (einschließlich), die den Prozentsatz der Viewer-Anfragen bestimmt, die an Kinesis Data Streams als Echtzeit-Protokolldatensätze gesendet werden. Um jede Viewer-Anforderung in Ihre Echtzeit-Protokolle aufzunehmen, geben Sie als Abtastrate 100 an. Sie können eine niedrigere Abtastrate wählen, um die Kosten zu senken, während Sie dennoch eine repräsentative Stichprobe von Anforderungsdaten in Ihren Echtzeit-Protokollen erhalten.

Felder

Eine Liste der Felder, die in jedem Datensatz des Echtzeitzugriffsprotokolls enthalten sind. Jeder Protokolldatensatz kann bis zu 40 Felder enthalten und Sie können auswählen, ob alle verfügbaren Felder empfangen werden, oder nur die Felder, die Sie für die Überwachung und Analyse der Leistung benötigen.

Die folgende Liste enthält jeden Feldnamen und eine Beschreibung der Informationen in diesem Feld. Die Felder werden in der Reihenfolge aufgelistet, in der sie in den Protokolldatensätzen angezeigt werden, die an Kinesis Data Streams übermittelt werden.

Die Felder 46–63 sind Common Media Client Data (CMCD), die Media-Player-Clients bei jeder Anforderung an CDNs senden können. Sie können mithilfe dieser Daten jede Anforderung besser verstehen, z. B. den Medientyp (Audio, Video), die Wiedergabegeschwindigkeit und die Streaming-Länge. Diese Felder werden nur dann in Ihren Echtzeit-Zugriffsprotokollen angezeigt, wenn sie an gesendet werden CloudFront.

1. **`timestamp`**

   Die Angabe zu Datum und Uhrzeit, an der der Edge-Server die Reaktion auf die Anforderung abgeschlossen hat.

2. **`c-ip`**

   Die IP-Adresse des Betrachters, die der Anfrage gestellt hat, z. B. `192.0.2.183` oder `2001:0db8:85a3::8a2e:0370:7334`. Wenn der Viewer einen HTTP-Proxy oder eine Load Balancer verwendet hat, um die Anforderung zu senden, entspricht der Wert dieses Feldes der IP-Adresse des Proxys bzw. des Load Balancers. Siehe auch das Feld `x-forwarded-for`.

3. **`s-ip`**

   Die IP-Adresse des CloudFront Servers, der die Anfrage bearbeitet hat, z. B. `192.0.2.183` oder`2001:0db8:85a3::8a2e:0370:7334`.

4. **`time-to-first-byte`**

   Die Anzahl der Sekunden zwischen dem Empfangen der Anforderung und dem Schreiben des ersten Bytes der Antwort, gemessen auf dem Server.

5. **`sc-status`**

   Den HTTP-Statuscode der Antwort des Servers (z. B. `200`).

6. **`sc-bytes`**

   Die Gesamtzahl der Bytes, die der Server als Antwort auf die Anforderung an den Viewer übermittelt hat, einschließlich Headern. Für WebSocket und gRPC-Verbindungen ist dies die Gesamtzahl der Byte, die vom Server über die Verbindung an den Client gesendet werden.

7. **`cs-method`**

Die vom Viewer empfangene HTTP-Anforderungsmethode.

8. **`cs-protocol`**

   Das Protokoll der Viewer-Anforderung (`http`, `https`, `grpcs`, `ws` oder `wss`).

9. **`cs-host`**

   Der Wert, den der Viewer in den `Host`-Header der Anforderung eingefügt hat. Wenn Sie den CloudFront Domainnamen in Ihren Objekt-URLs verwenden (z. B. d111111abcdef8.cloudfront.net), enthält dieses Feld diesen Domainnamen. Wenn Sie alternative Domänennamen (CNAMEs) in Ihren Objekt-URLs verwenden (z. B. www.example.com), enthält dieses Feld den alternativen Domänennamen.

10. **`cs-uri-stem`**

    Die gesamte Anforderungs-URL, einschließlich der Abfragezeichenfolge (falls vorhanden), jedoch ohne den Domänennamen. Beispiel, `/images/cat.jpg?mobile=true`.

    > **Note**
    >
    > In Standardprotokollen enthält der `cs-uri-stem`-Wert nicht die Abfragezeichenfolge.

11. **`cs-bytes`**

    Die Gesamtzahl der Bytes an Daten, die in der Anforderung des Viewers einschließlich Headern enthalten sind. Für WebSocket und gRPC-Verbindungen ist dies die Gesamtzahl der Byte, die vom Client an den Server über die Verbindung gesendet wurden.

12. **`x-edge-location`**

    Der Edge-Standort, an dem die Anfrage verarbeitet wurde. Jeder Edge-Standort wird anhand eines Codes aus drei Buchstaben und einer willkürlich zugewiesenen Zahl identifiziert, z. B. DFW3. Der Code aus drei Buchstaben entspricht in der Regel dem Code der International Air Transport Association (IATA) für einen Flughafen in der Nähe des Edge-Standorts. (Diese Abkürzungen ändern sich möglicherweise in der Zukunft.)

13. **`x-edge-request-id`**

    Eine undurchsichtige Zeichenfolge, die eine Anfrage eindeutig identifiziert. CloudFront sendet diese Zeichenfolge auch im `x-amz-cf-id` Antwort-Header.

14. **`x-host-header`**

Der Domainname der CloudFront Distribution (z. B. d111111abcdef8.cloudfront.net).

15. **`time-taken`**

    Die Anzahl der Sekunden (auf die Tausendstelsekunde genau, z. B. 0,082) ab dem Zeitpunkt, an dem der Server die Anforderung des Viewers empfängt, bis zu dem Zeitpunkt, an dem der Server das letzte Byte der Antwort in die Ausgabewarteschlange schreibt, wie auf dem Server gemessen. Aus der Perspektive der Viewers vergeht bis zum Empfang der gesamten Antwort aufgrund der Netzwerklatenz und des TCP-Puffervorgangs insgesamt mehr Zeit, als mit diesem Wert angegeben wird.

16. **`cs-protocol-version`**

    Die HTTP-Version, die der Viewer in der Anfrage angegeben hat: Mögliche Werte sind `HTTP/0.9`, `HTTP/1.0`, `HTTP/1.1`, `HTTP/2.0` und `HTTP/3.0`.

17. **`c-ip-version`**

    Die IP-Version der Anforderung (IPv4 oder IPv6).

18. **`cs-user-agent`**

    Der Wert für den `User-Agent`-Header in der Anfrage. Der `User-Agent`-Header bezeichnet die Quelle für die Anforderung, z. B. den Gerätetyp und den Browser, der die Anforderung abgesendet hat, oder die Suchmaschine, wenn die Anforderung von einer Suchmaschine stammt.

19. **`cs-referer`**

    Der Wert für den `Referer`-Header in der Anfrage. Der Name der Domäne, von der die Anforderung ausgegangen ist. Häufig vorkommende Referrer sind Suchmaschinen, andere Websites, die direkt auf Ihre Objekte verlinken, und Ihre eigene Website.

20. **`cs-cookie`**

    Der `Cookie`-Header in der Anforderung einschließlich der Name-Wert-Paaren und zugehörigen Attributen.

    > **Note**
    >
    > Dieses Feld wird auf 800 Bytes abgeschnitten.

21. **`cs-uri-query`**

Der Teil der Anforderungs-URL mit der Abfragezeichenfolge, wenn vorhanden.

22 **`x-edge-response-result-type`**

Die Art, wie der Server die Antwort direkt vor der Rücksendung der Antwort an den Viewer klassifiziert hat. Siehe auch das Feld `x-edge-result-type`. Mögliche Werte sind:

- `Hit` – Der Server hat das Objekt aus dem Cache für den Betrachter bereitgestellt.
- `RefreshHit` – Der Server hat das Objekt im Edge-Zwischenspeicher gefunden, es war jedoch abgelaufen. Daher nahm der Server Kontakt mit dem Ursprung auf, um zu überprüfen, ob der Zwischenspeicher die neueste Version des Objekts enthalten hatte.
- `Miss` – Die Anforderung konnte nicht durch ein Objekt im Zwischenspeicher bedient werden. Daher hat der Server die Anforderung an den Ursprungs-Server weitergeleitet und das Ergebnis an den Betrachter ausgegeben.
- `LimitExceeded`— Die Anfrage wurde abgelehnt, weil ein CloudFront Kontingent (früher als Limit bezeichnet) überschritten wurde.
- `CapacityExceeded` – Der Server hat den Fehler 503 zurückgegeben, da er zum Zeitpunkt der Anforderung nicht über genügend Kapazitäten verfügte, um das Objekt bereitzustellen.
- `Error` – In der Regel bedeutet dies, dass die Anforderung zu einem Client-Fehler (d. h. der Wert des Felds `sc-status` liegt im `4xx`-Bereich) oder zu einem Serverfehler geführt hat (d. h. der Wert des Felds `sc-status` liegt im `5xx`-Bereich).

  Wenn der Wert des Felds `x-edge-result-type` `Error` ist und der Wert dieses Felds nicht `Error` ist, wurde die Verbindung vor dem Abschluss des Downloads durch den Client getrennt.
- `Redirect` – Der Server hat den Betrachter entsprechend den Verteilungseinstellungen von HTTP zu HTTPS umgeleitet.
- `LambdaExecutionError`— Die der Distribution zugeordnete Lambda @Edge -Funktion wurde aufgrund einer fehlerhaften Zuordnung, eines Funktionstimeouts, eines AWS Abhängigkeitsproblems oder eines anderen allgemeinen Verfügbarkeitsproblems nicht abgeschlossen.

23 **`x-forwarded-for`**

Wenn der Viewer einen HTTP-Proxy oder Load Balancer verwendet hat, um die Anforderung zu senden, entspricht der Wert des Felds `c-ip` der IP-Adresse des Proxys oder Load Balancers. In diesem Fall stellt dieses Feld die IP-Adresse des Viewers dar, von dem die Anfrage stammt. Dieses Feld kann mehrere durch Kommas getrennte IP-Adressen enthalten. Jede

IP-Adresse kann eine IPv4-Adresse (z. B. `192.0.2.183`) oder eine IPv6-Adresse (z. B. `2001:0db8:85a3::8a2e:0370:7334`) sein.

24 **`ssl-protocol`**

Wenn für die Anfrage HTTPS verwendet wurde, enthält dieses Feld SSL/TLS das Protokoll, das der Betrachter und der Server für die Übertragung der Anfrage und Antwort ausgehandelt haben. Eine Liste der möglichen Werte finden Sie unter den unterstützten SSL/TLS ProtokollenUnterstützte Protokolle und Chiffren zwischen Viewern und CloudFront.

25 **`ssl-cipher`**

Wenn für die Anfrage HTTPS verwendet wurde, enthält dieses Feld die SSL/TLS Chiffre, die der Betrachter und der Server für die Verschlüsselung der Anfrage und Antwort ausgehandelt haben. Eine Liste der möglichen Werte finden Sie unter den unterstützten SSL/TLS Verschlüsselungen. Unterstützte Protokolle und Chiffren zwischen Viewern und CloudFront

26 **`x-edge-result-type`**

Art der Klassifizierung der Antwort durch den Server, nachdem das letzte Byte den Server verlassen hat. Manchmal wird der Ergebnistyp zwischen dem Zeitpunkt, an dem Server zum Senden der Antwort bereit ist, und dem Zeitpunkt, an dem er das Senden der Antwort abgeschlossen hat, geändert. Siehe auch das Feld `x-edge-response-result-type`.

Angenommen, im HTTP-Streaming findet der Server ein Segment des Streams im Cache. In diesem Szenario würde der Wert dieses Feldes normalerweise sei `Hit`. Wenn der Viewer jedoch die Verbindung schließt, bevor der Server das ganze Segment übermittelt hat, ist der endgültige Ergebnistyp (und damit der Wert dieses Felds) `Error`.

WebSocket und gRPC-Verbindungen haben `Miss` für dieses Feld den Wert von, da der Inhalt nicht zwischengespeichert werden kann und direkt an den Ursprung weitergeleitet wird.

Mögliche Werte sind:

- `Hit` – Der Server hat das Objekt aus dem Cache für den Betrachter bereitgestellt.
- `RefreshHit` – Der Server hat das Objekt im Edge-Zwischenspeicher gefunden, es war jedoch abgelaufen. Daher nahm der Server Kontakt mit dem Ursprung auf, um zu überprüfen, ob der Zwischenspeicher die neueste Version des Objekts enthalten hatte.
- `Miss` – Die Anforderung konnte nicht durch ein Objekt im Zwischenspeicher bedient werden. Daher hat der Server die Anforderung an den Ursprung weitergeleitet und das Ergebnis an den Betrachter ausgegeben.

- `LimitExceeded`— Die Anfrage wurde abgelehnt, weil ein CloudFront Kontingent (früher als Limit bezeichnet) überschritten wurde.
- `CapacityExceeded` – Der Server hat den HTTP-Statuscode 503 zurückgegeben, da er zum Zeitpunkt der Anforderung nicht über genügend Kapazitäten für die Bereitstellung des Objekts verfügte.
- `Error` – In der Regel bedeutet dies, dass die Anforderung zu einem Client-Fehler (d. h. der Wert des Felds `sc-status` liegt im `4xx`-Bereich) oder zu einem Serverfehler geführt hat (d. h. der Wert des Felds `sc-status` liegt im `5xx`-Bereich). Wenn der Wert des Felds `sc-status` `200` oder wenn der Wert dieses Felds `Error` ist und der Wert des Felds `x-edge-response-result-type` nicht `Error` ist, war die HTTP-Anforderung erfolgreich. Die Client-Verbindung wurde jedoch getrennt, bevor alle Bytes empfangen wurden.
- `Redirect` – Der Server hat den Betrachter entsprechend den Verteilungseinstellungen von HTTP zu HTTPS umgeleitet.
- `LambdaExecutionError`— Die der Distribution zugeordnete Lambda @Edge -Funktion wurde aufgrund einer fehlerhaften Zuordnung, eines Funktionstimeouts, eines AWS Abhängigkeitsproblems oder eines anderen allgemeinen Verfügbarkeitsproblems nicht abgeschlossen.

27. **`fle-encrypted-fields`**

    Die Anzahl der Felder für die Verschlüsselung auf Feldebene, die der Server verschlüsselt und an den Ursprung weitergeleitet hat. CloudFront Server streamen die verarbeitete Anfrage an den Ursprung, während sie Daten verschlüsseln, sodass dieses Feld auch dann einen Wert haben kann, wenn der Wert von ein Fehler `fle-status` ist.

28. **`fle-status`**

    Wenn die Verschlüsselung auf Feldebene für eine Verteilung konfiguriert ist, enthält dieses Feld einen Code, der angibt, ob der Anforderungstext erfolgreich verarbeitet wurde. Wenn der Server den Anforderungstext erfolgreich verarbeitet, Werte in den angegebenen Feldern verschlüsselt und die Anforderung an den Ursprung weiterleitet, ist der Wert dieses Felds `Processed`. Der Wert von `x-edge-result-type` kann in diesem Fall immer noch auf einen clientseitigen oder serverseitigen Fehler hinweisen.

    Mögliche Werte für dieses Feld sind:

    - `ForwardedByContentType` – Der Server hat die Anforderung ohne Parsing oder Verschlüsselung an den Ursprung weitergeleitet, da kein Content-Typ konfiguriert wurde.

- `ForwardedByQueryArgs` – Der Server hat die Anforderung ohne Parsing oder Verschlüsselung an den Ursprung weitergeleitet, da die Anforderung ein Abfrageargument enthält, das nicht in der Konfiguration für die Verschlüsselung auf Feldebene enthalten war.
- `ForwardedDueToNoProfile` – Der Server hat die Anforderung ohne Parsing oder Verschlüsselung an den Ursprung weitergeleitet, da in der Konfiguration für die Verschlüsselung auf Feldebene kein Profil angegeben wurde.
- `MalformedContentTypeClientError` – Der Server hat die Anforderung zurückgewiesen und einen HTTP-400-Statuscode an den Betrachter zurückgegeben, da der Wert des `Content-Type`-Headers ein ungültiges Format hatte.
- `MalformedInputClientError` – Der Server hat die Anforderung zurückgewiesen und einen HTTP 400-Statuscode an den Betrachter zurückgegeben, da der Anforderungstext ein ungültiges Format hatte.
- `MalformedQueryArgsClientError` – Der Server hat die Anforderung zurückgewiesen und einen HTTP-400-Statuscode an den Betrachter zurückgegeben, weil ein Abfrageargument leer war oder ein ungültiges Format hatte.
- `RejectedByContentType` – Der Server hat die Anforderung zurückgewiesen und einen HTTP-400-Statuscode an den Betrachter zurückgegeben, da in der Konfiguration für die Verschlüsselung auf Feldebene kein Content-Typ angegeben wurde.
- `RejectedByQueryArgs` – Der Server hat die Anforderung zurückgewiesen und einen HTTP-400-Statuscode an den Betrachter zurückgegeben, da in der Konfiguration für die Verschlüsselung auf Feldebene kein Abfrageargument angegeben wurde.
- `ServerError` – Der Ursprungs-Server hat einen Fehler zurückgegeben.

Wenn die Anforderung das Kontingent für die Verschlüsselung auf Feldebene überschreitet (früher als Limit bezeichnet), enthält dieses Feld einen der folgenden Fehlercodes und der Server gibt dem Viewer den HTTP-Statuscode 400 zurück. Eine Liste der aktuellen Kontingente für die Verschlüsselung auf Feldebene finden Sie unter Kontingente für Verschlüsselung auf Feldebene.

- `FieldLengthLimitClientError` – Ein Feld, das für die Verschlüsselung konfiguriert ist, hat die maximal zulässige Länge überschritten.
- `FieldNumberLimitClientError` – Eine Anforderung, die die Verteilung verschlüsseln soll, enthält mehr Felder als zulässig.
- `RequestLengthLimitClientError` – Die Länge des Anfragetexts hat die maximal zulässige Länge, wenn die Verschlüsselung auf Feldebene konfiguriert ist, überschritten.

29 **`sc-content-type`**

Der Wert des HTTP `Content-Type`-Headers der Antwort.

30 **`sc-content-len`**

Der Wert des HTTP `Content-Length`-Headers der Antwort.

31 **`sc-range-start`**

Wenn die Antwort den HTTP `Content-Range`-Header enthält, enthält dieses Feld den Bereichsstartwert.

32 **`sc-range-end`**

Wenn die Antwort den HTTP `Content-Range`-Header enthält, enthält dieses Feld den Bereichsendwert.

33 **`c-port`**

Die Portnummer der Anforderung des Viewers.

34 **`x-edge-detailed-result-type`**

Dieses Feld enthält den gleichen Wert wie der Wert für das `x-edge-result-type`-Feld, außer in den folgenden Fällen:

- Wenn das Objekt dem Viewer aus der Origin-Shield-Ebene bereitgestellt wurde, enthält dieses Feld `OriginShieldHit`.
- Wenn sich das Objekt nicht im CloudFront Cache befand und die Antwort von einer Lambda @Edge -Funktion für die ursprüngliche Anfrage generiert wurde, enthält `MissGeneratedResponse` dieses Feld.
- Wenn der Wert des `x-edge-result-type`-Felds `Error` lautet, enthält dieses Feld einen der folgenden Werte mit weiteren Informationen zum Fehler:
  - `AbortedOrigin` – Der Server hat ein Problem mit dem Ursprung festgestellt.
  - `ClientCommError` – Die Antwort auf den Betrachter wurde aufgrund eines Kommunikationsproblems zwischen dem Server und dem Betrachter unterbrochen.
  - `ClientGeoBlocked` – Die Verteilung ist so konfiguriert, dass Anforderungen vom geografischen Standort des Viewers abgelehnt werden.
  - `ClientHungUpRequest` — Der Betrachter wurde beim Senden der Anfrage vorzeitig gestoppt.

- `Error` – Es ist ein Fehler aufgetreten, dessen Fehlertyp zu keiner der anderen Kategorien passt. Dieser Fehlertyp kann auftreten, wenn der Server eine Fehlerantwort aus dem Cache ausgibt.
- `InternalError`— Ein interner CloudFront Fehler ist aufgetreten. Dies weist in der Regel darauf hin, dass der Fehler von der CloudFront Infrastruktur herrührte, nicht vom Ursprung oder vom Betrachter.
- `InvalidRequest` – Der Server hat eine ungültige Anforderung vom Betrachter erhalten.
- `InvalidRequestBlocked` — Der Zugriff auf die angeforderte Ressource ist blockiert.
- `InvalidRequestCertificate`— Die Verteilung stimmt nicht mit dem SSL/TLS Zertifikat überein, für das die HTTPS-Verbindung hergestellt wurde.
- `InvalidRequestHeader` – Die Anforderung enthielt einen ungültigen Header.
- `InvalidRequestMethod` — Die Verteilung ist nicht für eine Verarbeitung der verwendeten HTTP-Anforderungsmethode konfiguriert. Dies kann vorkommen, wenn die Verteilung nur Anforderungen unterstützt, die zwischengespeichert werden können.
- `LambdaLimitExceeded`— Die Anfrage hat ein Lambda @Edge -Funktionslimit überschritten.
- `OriginCommError` – Bei der Anforderung ist eine Zeitüberschreitung aufgetreten, während eine Verbindung mit dem Ursprung hergestellt wurde oder Daten aus dem Ursprung gelesen wurden.
- `OriginConnectError` – Der Server konnte keine Verbindung zum Ursprung herstellen.
- `OriginContentRangeLengthError` – Der `Content-Length`-Header in der Antwort des Ursprungs stimmt nicht mit der Länge im `Content-Range`-Header überein.
- `OriginDnsError` – Der Server konnte den Domänennamen des Ursprungs nicht auflösen.
- `OriginError` — Der Ursprung gab eine falsche Antwort zurück.
- `OriginHeaderTooBigError` – Ein vom Ursprung zurückgegebener Header ist für eine Verarbeitung durch den Edge-Server zu groß.
- `OriginInvalidResponseError` — Der Ursprung gab eine ungültige Antwort zurück.
- `OriginReadError` – Der Server konnte nicht vom Ursprung lesen.
- `OriginWriteError` – Der Server konnte nicht in den Ursprung schreiben.
- `OriginZeroSizeObjectError` — Ein Nullgrößenobjekt, das vom Ursprung gesendet wurde, führte zu einem Fehler.
- `SlowReaderOriginError` — Der Betrachter hat die Nachricht, die den Ursprungsfehler verursacht hat, nur langsam gelesen.

35. **`c-country`**

    Ein Ländercode, der den geografischen Standort des Viewers darstellt, der von der IP-Adresse des Viewers festgelegt wird. Die Liste der Ländercodes finden Sie unter ISO 3166-1 alpha-2.

36. **`cs-accept-encoding`**

    Der Wert für den `Accept-Encoding`-Header in der Viewer-Anforderung.

37. **`cs-accept`**

    Der Wert für den `Accept`-Header in der Viewer-Anforderung.

38. **`cache-behavior-path-pattern`**

    Das Pfadmuster, das das Cache-Verhalten identifiziert, das der Viewer-Anforderung entspricht.

39. **`cs-headers`**

    Die HTTP-Header (Namen und Werte) in der Viewer-Anforderung.

    > **Note**
    >
    > Dieses Feld wird auf 800 Bytes abgeschnitten.

40. **`cs-header-names`**

    Die Namen der HTTP-Header (keine Werte) in der Viewer-Anforderung.

    > **Note**
    >
    > Dieses Feld wird auf 800 Bytes abgeschnitten.

41. **`cs-headers-count`**

    Die Anzahl der HTTP-Header in der Viewer-Anforderung.

42. **`primary-distribution-id`**

    Wenn die kontinuierliche Bereitstellung aktiviert ist, identifiziert diese ID, welche Distribution in der aktuellen Distribution die primäre ist.

43. **`primary-distribution-dns-name`**

Wenn Continuous Deployment aktiviert ist, zeigt dieser Wert den primären Domainnamen an, der sich auf die aktuelle CloudFront Distribution bezieht (z. B. d111111abcdef8.cloudfront.net).

44. **`origin-fbl`**

    Die Anzahl der Sekunden der First-Byte-Latenz zwischen und Ihrem Ursprung. CloudFront

45. **`origin-lbl`**

    Die Anzahl der Sekunden der Latenz im letzten Byte zwischen CloudFront und Ihrem Ursprung.

46. **`asn`**

    Die autonome Systemnummer (ASN) des Viewers.

47.

    > **CMCD-Felder in Echtzeit-Zugriffsprotokollen**
    >
    > Weitere Informationen zu diesen Feldern finden Sie im Dokument CTA Specification Web Application Video Ecosystem — Common Media Client Data. CTA-5004

48. **`cmcd-encoded-bitrate`**

    Die kodierte Bitrate des angeforderten Audio- oder Videoobjekts.

49. **`cmcd-buffer-length`**

    Die Pufferlänge des angeforderten Medienobjekts.

50. **`cmcd-buffer-starvation`**

    Ob der Puffer irgendwann zwischen der vorherigen Anforderung und der Objektanforderung verbraucht wurde. Dies kann dazu führen, dass sich der Player in einem Neupufferungsstatus befindet, was die Video- oder Audiowiedergabe stoppen kann.

51. **`cmcd-content-id`**

    Eine eindeutige Zeichenfolge, die den aktuellen Inhalt identifiziert.

52. **`cmcd-object-duration`**

    Die Wiedergabedauer des angeforderten Objekts (in Millisekunden).

53. **`cmcd-deadline`**

Die Frist ab dem Zeitpunkt der Anforderung, zu der das erste Beispiel dieses Objekts verfügbar sein muss, damit ein Unterlauf des Puffers oder andere Wiedergabeprobleme vermieden werden.

54. **`cmcd-measured-throughput`**

    Der vom Client gemessene Durchsatz zwischen Client und Server.

55. **`cmcd-next-object-request`**

    Der relative Pfad des nächsten angeforderten Objekts.

56. **`cmcd-next-range-request`**

    Wenn es sich bei der nächsten Anforderung um eine teilweise Objektanforderung handelt, gibt diese Zeichenfolge den Bytebereich an, der angefordert werden soll.

57. **`cmcd-object-type`**

    Der Medientyp des aktuell angeforderten Objekts.

58. **`cmcd-playback-rate`**

    1 bei Echtzeit, 2 bei doppelter Geschwindigkeit, 0 bei Nichtwiedergabe.

59. **`cmcd-requested-maximum-throughput`**

    Der angeforderte maximale Durchsatz, den der Client für die Bereitstellung von Ressourcen für ausreichend hält.

60. **`cmcd-streaming-format`**

    Das Streaming-Format, das die aktuelle Anforderung definiert.

61. **`cmcd-session-id`**

    Eine GUID, die die aktuelle Wiedergabesitzung identifiziert.

62. **`cmcd-stream-type`**

    Token, das die Verfügbarkeit von Segmenten identifiziert. `v` = alle Segmente sind verfügbar. `l` = Segmente werden im Laufe der Zeit verfügbar.

63. **`cmcd-startup`**

    Der Schlüssel wird ohne Wert angegeben, wenn das Objekt beim Start, bei der Suche oder bei der Wiederherstellung nach einem Ereignis mit leerem Puffer dringend benötigt wird.

64. **`cmcd-top-bitrate`**

Die Wiedergabe mit der höchsten Bitrate, die der Client abspielen kann.

65 **`cmcd-version`**

Die Version dieser Spezifikation, die für die Interpretation der definierten Schlüsselnamen und Werte verwendet wird. Wenn dieser Schlüssel weggelassen wird, müssen der Client und der Server die Werte so interpretieren, als ob sie in Version 1 definiert sind.

66 **`r-host`**

Dieses Feld wird für Ursprungsanforderungen gesendet und gibt die Domain des Ursprungsservers an, der für die Bereitstellung des Objekts verwendet wird. Wenn Fehler auftreten, können Sie dieses Feld verwenden, um den Ursprung des letzten Verbindungsversuchs zu finden, zum Beispiel: *`cd8jhdejh6a`*`.mediapackagev2.us-east-1.amazonaws.com`.

67 **`sr-reason`**

Dieses Feld gibt einen Grund an, warum der Ursprung ausgewählt wurde. Es ist leer, wenn eine Anforderung an den primären Ursprung erfolgreich ist.

Wenn ein Ursprungs-Failover auftritt, enthält das Feld den HTTP-Fehlercode, der zum Failover geführt hat, z. B. `Failover:403` oder `Failover:502`. Wenn im Fall eines Ursprungs-Failovers auch die wiederholte Anforderung fehlschlägt und Sie keine benutzerdefinierten Fehlerseiten konfiguriert haben, zeigt `r-status` die Antwort des zweiten Ursprungs an. Wenn Sie jedoch bei der Konfiguration des Ursprungs-Failovers benutzerdefinierte Fehlerseiten konfiguriert haben, enthält dieses Feld die Antwort des zweiten Ursprungs, wenn die Anforderung fehlgeschlagen ist und stattdessen eine benutzerdefinierte Fehlerseite zurückgegeben wurde.

Wenn kein Ursprungs-Failover, sondern eine Auswahl des Ursprungs anhand von Media Quality-Aware Resilience (MQAR) erfolgt, wird dies als `MediaQuality` protokolliert. Weitere Informationen finden Sie unter Media Quality-Aware Resiliency.

68 **`x-edge-mqcs`**

Dieses Feld gibt den Media Quality Confidence Score (MQCS) (Bereich: 0 — 100) für Mediensegmente an, die in den CMSD-Antwort-Headern von v2 CloudFront abgerufen wurden. MediaPackage Dieses Feld ist für Anfragen verfügbar, die einem Cache-Verhalten entsprechen, das über eine Ursprungsgruppe verfügt. MQAR-enabled CloudFront protokolliert dieses Feld für Mediensegmente, die zusätzlich zu den ursprünglichen Anfragen auch aus dem Cache bedient werden. Weitere Informationen finden Sie unter Media Quality-Aware Resiliency.

69 **`distribution-tenant-id`**

Die ID des Distributionsmandanten.

70. **`connection-id`**

    Eine eindeutige Kennung für die TLS-Verbindung.

    Sie müssen mTLS für Ihre Distributionen aktivieren, bevor Sie Informationen für dieses Feld abrufen können. Weitere Informationen finden Sie unter Gegenseitige TLS-Authentifizierung mit CloudFront (Viewer mTLS).

71. **`viewer-request-log-data`**

    Benutzerdefinierte Daten, die Ihre CloudFront Viewer-Anforderungsfunktion mithilfe der `cf.logCustomData()` Hilfsmethode sendet. Dieses Feld enthält den Zeichenfolgenwert, den Sie während des Viewer-Anforderungsereignisses an die Methode übergeben haben. CloudFront URL-encodes den Wert und kürzt ihn auf ein Maximum von 800 Byte. Wenn dem Viewer-Anforderungsereignis keine CloudFront Funktion zugeordnet ist oder wenn die Funktion nicht aufgerufen wird`cf.logCustomData()`, gilt dies für dieses Feld. - Weitere Informationen finden Sie unter Allgemeine Hilfsmethoden.

72. **`viewer-response-log-data`**

    Benutzerdefinierte Daten, die Ihre CloudFront Viewer-Antwortfunktion mithilfe der `cf.logCustomData()` Hilfsmethode sendet. Dieses Feld enthält den Zeichenkettenwert, den Sie während des Zuschauer-Antwortereignisses an die Methode übergeben haben. CloudFront URL-encodes den Wert und kürzt ihn auf ein Maximum von 800 Byte. Wenn dem Antwortereignis des Betrachters keine CloudFront Funktion zugeordnet ist oder wenn die Funktion nicht aufgerufen wird`cf.logCustomData()`, gilt dies für dieses Feld. - Weitere Informationen finden Sie unter Allgemeine Hilfsmethoden.

Endpunkt (Kinesis Data Streams)

Der Endpunkt enthält Informationen über die Kinesis-Datenströme, an die Sie Echtzeit-Protokolle senden möchten. Sie stellen den Amazon-Ressourcennamen (ARN) des Streams oder der Funktion bereit.

Weitere Informationen zum Erstellen eines Kinesis-Datenstroms finden Sie in den folgenden Themen im Entwicklerhandbuch für Amazon Kinesis Data Streams.

- Erstellen und Verwalten von Datenströmen

- Führen Sie grundlegende Kinesis Data Streams Streams-Operationen mit dem AWS CLI
- Einen Stream erstellen (verwendet den AWS SDK für Java)

Wenn Sie einen Datensteam erstellen, müssen Sie die Anzahl der Shards angeben. Verwenden Sie die folgenden Informationen, um die Anzahl der benötigten Shards zu schätzen.

So schätzen Sie die Anzahl der Shards für Ihren Kinesis-Datenstream:

1. Berechnen (oder schätzen) Sie die Anzahl der Anforderungen pro Sekunde, die Ihre CloudFront-Verteilung erhält.

   Sie können die CloudFrontNutzungsberichte (in der CloudFront Konsole) und die CloudFront Metriken (in den CloudWatch Konsolen CloudFront und Amazon) verwenden, um Ihre Anfragen pro Sekunde zu berechnen.
2. Ermitteln Sie die typische Größe eines einzelnen Echtzeit-Zugriffsprotokolldatensatzes.

   Im Allgemeinen ist ein einzelner Protokolldatensatz etwa 500 Byte groß. Ein typischer Datensatz, der alle verfügbaren Felder enthält, ist in der Regel etwa 1 KB groß.

   Wenn Sie Ihre Protokoll-Datensatzgröße nicht genau kennen, können Sie Echtzeitprotokolle mit einer niedrigen Abtastrate (z. B. 1 %) aktivieren und dann die durchschnittliche Datensatzgröße anhand von Überwachungsdaten in Kinesis Data Streams berechnen (Gesamtzahl der eingehenden Bytes dividiert durch die Gesamtzahl der Datensätze).
3. Wählen Sie auf der Preisseite für Amazon Kinesis Data Streams unter AWS -PreisrechnerJetzt Ihren individuellen Kostenvoranschlag erstellen aus.

   - Geben Sie im Rechner die Anzahl der Anforderungen (Datensätze) pro Sekunde ein.
   - Geben Sie die durchschnittliche Datensatzgröße eines einzelnen Protokolldatensatzes ein.
   - Wählen Sie dann Berechnungen anzeigen aus.

   Der Preisrechner zeigt Ihnen die Anzahl der benötigten Shards und die geschätzten Kosten an.

IAM role (IAM-Rolle)

Die AWS Identity and Access Management (IAM-) Rolle, die die CloudFront Erlaubnis erteilt, Echtzeit-Zugriffsprotokolle für Ihren Kinesis-Datenstream bereitzustellen.

Wenn Sie mit der CloudFront Konsole eine Konfiguration für das Echtzeit-Zugriffsprotokoll erstellen, können Sie Neue Servicerolle erstellen auswählen, damit die Konsole die IAM-Rolle für Sie erstellt.

Wenn Sie eine Echtzeit-Zugriffsprotokollkonfiguration mit AWS CloudFormation oder der CloudFront API (AWS CLI oder dem SDK) erstellen, müssen Sie die IAM-Rolle selbst erstellen und den Rollen-ARN angeben. Verwenden Sie die folgenden Richtlinien, um die IAM-Rolle selbst zu erstellen:

IAM-Rollen-Vertrauensrichtlinie

Um die folgende Vertrauensrichtlinie für IAM-Rollen zu verwenden, *111122223333* ersetzen Sie sie durch Ihre AWS-Konto Nummer. Das `Condition` Element in dieser Richtlinie trägt dazu bei, das Problem des verwirrten Stellvertreters zu vermeiden, da CloudFront Sie diese Rolle nur im Namen einer Abteilung in Ihrem AWS-Konto Unternehmen übernehmen können.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "cloudfront.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "111122223333"
                }
            }
        }
    ]
}
```

IAM-Rollen-Berechtigungsrichtlinie für einen unverschlüsselten Daten-Stream

Um die folgende Richtlinie zu verwenden, *arn:aws:kinesis:us-east-2:123456789012:stream/StreamName* ersetzen Sie sie durch den ARN Ihres Kinesis-Datenstroms.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kinesis:DescribeStreamSummary",
                "kinesis:DescribeStream",
                "kinesis:PutRecord",
                "kinesis:PutRecords"
            ],
            "Resource": [
                "arn:aws:kinesis:us-east-2:123456789012:stream/StreamName"
            ]
        }
    ]
}
```

IAM-Rollen-Berechtigungsrichtlinie für einen verschlüsselten Daten-Stream

Um die folgende Richtlinie zu verwenden, *arn:aws:kinesis:us-east-2:123456789012:stream/StreamName* ersetzen Sie sie durch den ARN Ihres Kinesis-Datenstreams und *arn:aws:kms:us-east-2:123456789012:key/e58a3d0b-fe4f-4047-a495-ae03cc73d486* durch den ARN Ihres AWS KMS key.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kinesis:DescribeStreamSummary",
                "kinesis:DescribeStream",
                "kinesis:PutRecord",
                "kinesis:PutRecords"
            ],
```

```
            "Resource": [
                "arn:aws:kinesis:us-east-2:123456789012:stream/StreamName"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:GenerateDataKey"
            ],
            "Resource": [
                "arn:aws:kms:us-east-2:123456789012:key/e58a3d0b-fe4f-4047-a495-
ae03cc73d486"
            ]
        }
    ]
}
```

Erstellen eines Kinesis-Datenstromverbrauchers

Um Ihre Echtzeit-Zugriffsprotokolle zu lesen und zu analysieren, erstellen oder verwenden Sie einen Kinesis Data Streams Streams-Consumer. Wenn Sie einen Consumer für CloudFront Echtzeit-Logs erstellen, ist es wichtig zu wissen, dass die Felder in jedem Echtzeit-Zugriffsprotokolldatensatz immer in derselben Reihenfolge übermittelt werden, wie im Felder Abschnitt aufgeführt. Stellen Sie sicher, dass Sie Ihren Verbraucher für diese feste Reihenfolge erstellen.

Stellen Sie sich zum Beispiel eine Konfiguration für ein Echtzeit-Zugriffsprotokoll vor, die nur diese drei Felder enthält: `time-to-first-bytesc-status`, und`c-country`. In diesem Szenario ist das letzte Feld, `c-country`, immer Feld Nummer 3 in jedem Protokolldatensatz. Wenn Sie jedoch später Felder zur Konfiguration des Echtzeit-Zugriffsprotokolls hinzufügen, kann sich die Platzierung der einzelnen Felder in einem Datensatz ändern.

Wenn Sie beispielsweise die Felder `sc-bytes` und `time-taken` zur Konfiguration des Echtzeit-Zugriffsprotokolls hinzufügen, werden diese Felder in jeden Protokolldatensatz in der im Felder Abschnitt angegebenen Reihenfolge eingefügt. Die resultierende Reihenfolge aller fünf Felder lautet dann `time-to-first-byte`, `sc-status`, `sc-bytes`, `time-taken` und `c-country`. Das Feld `c-country`hatte ursprünglich die Feldnummer 3, hat jetzt jedoch Feldnummer 5. Stellen Sie sicher, dass Ihre Verbraucheranwendung Felder verarbeiten kann, deren Position in

einem Protokolldatensatz geändert wird, falls Sie Felder zu Ihrer Konfiguration für das Echtzeit-Zugriffsprotokoll hinzufügen.

Beheben Sie Probleme mit Zugriffsprotokollen in Echtzeit

Nachdem Sie eine Konfiguration für das Echtzeit-Zugriffsprotokoll erstellt haben, stellen Sie möglicherweise fest, dass keine Datensätze (oder nicht alle Datensätze) an Kinesis Data Streams übermittelt werden. In diesem Fall sollten Sie zunächst überprüfen, ob Ihre CloudFront -Verteilung Viewer-Anforderungen empfängt. Wenn dies der Fall ist, können Sie die folgende Einstellung überprüfen, um die Fehlerbehebung fortzusetzen.

IAM-Rollenberechtigungen

: Um Echtzeit-Zugriffsprotokolldatensätze für Ihren Kinesis-Datenstrom bereitzustellen, CloudFront verwendet die IAM-Rolle in der Konfiguration des Echtzeit-Zugriffsprotokolls. Stellen Sie sicher, dass die Rollen-Vertrauensrichtlinie und die Rollen-Berechtigungsrichtlinie mit den in angezeigten Richtlinien übereinstimme IAM role (IAM-Rolle).

Kinesis Data Streams-Drosselung

: Wenn Echtzeit-Zugriffsprotokolldatensätze schneller in Ihren Kinesis-Datenstream CloudFront geschrieben werden, als der Stream verarbeiten kann, drosselt Kinesis Data Streams möglicherweise die Anfragen von. CloudFront In diesem Fall können Sie die Anzahl der Shards in Ihrem Kinesis-Datenstream erhöhen. Jeder Shard unterstützt Schreibvorgänge von bis zu 1.000 Datensätzen pro Sekunde bis zu einem maximalen Datenschreibvolumen von 1 MB pro Sekunde.

Protokolle für Edge-Funktionen

Sie können Amazon CloudWatch Logs verwenden, um Protokolle für Ihre Edge-Funktionen, sowohl Lambda @Edge als auch CloudFront Functions, abzurufen. Sie können über die CloudWatch Konsole oder die Logs-API auf die CloudWatch Protokolle zugreifen.

> **Important**
>
> Wir empfehlen Ihnen, die Protokolle zu verwenden, um die Art der Anfragen nach Ihren Inhalten zu verstehen, und nicht, um alle Anfragen vollständig zu erfassen. CloudFront liefert Edge-Funktionsprotokolle nach bestem Wissen und Gewissen. Der Protokolleintrag für eine bestimmte Anfrage wird möglicherweise viel später übermittelt, als die Anfrage tatsächlich verarbeitet wurde; in seltenen Fällen kann es auch sein, dass ein Protokolleintrag gar nicht

übermittelt wird. Wenn ein Protokolleintrag nicht in den Protokollen für Edge-Funktionen enthalten ist, stimmt die Anzahl der Einträge in den Protokollen für Edge-Funktionen nicht mit deren Anzahl in den Abrechnungs- und Nutzungsberichten für AWS überein.

Themen

- Lambda@Edge-Protokolle
- CloudFront Funktionen, Protokolle

Lambda@Edge-Protokolle

Lambda @Edge sendet automatisch CloudWatch Funktionsprotokolle an Logs und erstellt Protokollstreams dort, AWS-Regionen wo die Funktionen aufgerufen werden. Wenn Sie eine Funktion in erstellen oder ändern AWS Lambda, können Sie entweder den Standardnamen der CloudWatch Protokollgruppe verwenden oder ihn anpassen.

- Der Standardname der Protokollgruppe lautet `/aws/lambda/<FunctionName>`, wobei *`<FunctionName>`* der Name ist, den Sie beim Erstellen der Funktion angegeben haben. Beim Senden von CloudWatch Protokollen an fügt Lambda @Edge dem Funktionsnamen automatisch das `us-east-1` Präfix hinzu, sodass der Protokollgruppenname lautet`/aws/lambda/us-east-1.`*`<FunctionName>`*. Dieses Präfix entspricht dem AWS-Region Ort, an dem die Funktion erstellt wurde. Dieses Präfix bleibt Teil des Protokollgruppennamens, auch in anderen Regionen, in denen die Funktion aufgerufen wird.
- Wenn Sie einen benutzerdefinierten Protokollgruppennamen angeben, wie z. B. */MyLogGroup*, fügt Lambda@Edge das Regionspräfix nicht hinzu. Der Name der Protokollgruppe bleibt in allen anderen Regionen, in denen die Funktion aufgerufen wird, derselbe.

> **Note**
>
> Wenn Sie eine benutzerdefinierte Protokollgruppe erstellen und denselben Namen wie den standardmäßigen `/aws/lambda/`*`<FunctionName>`* angeben, fügt Lambda@Edge dem Funktionsnamen das Präfix `us-east-1` hinzu.

Neben der Anpassung des Protokollgruppennamens unterstützen Lambda@Edge-Funktionen auch JSON- und Klartext-Protokollformate sowie die Filterung auf Protokollebene. Weitere Informationen

finden Sie unter Konfigurieren erweiterter Protokollierungsoptionen für die Lambda-Funktion im Entwicklerhandbuch für AWS Lambda .

> **Note**
>
> Lambda@Edge drosselt Protokolle basierend auf dem angeforderten Volumen und der Größe der Protokolle.

Sie müssen die CloudWatch Protokolldateien in der richtigen Region überprüfen, um Ihre Lambda @Edge -Funktionsprotokolldateien zu sehen. Um die Regionen zu sehen, in denen Ihre Lambda @Edge -Funktion ausgeführt wird, sehen Sie sich Diagramme der Metriken für die Funktion in der CloudFront Konsole an. Metriken werden für jede -Region angezeigt. Auf derselben Seite können Sie eine Region auswählen und Protokolldateien für diese anzeigen, um Probleme zu untersuchen.

Weitere Informationen zur Verwendung von CloudWatch Logs mit Lambda @Edge -Funktionen finden Sie in den folgenden Themen:

- Weitere Informationen zum Anzeigen von Diagrammen im Bereich Überwachung der CloudFront Konsole finden Sie unterthe section called “Überwachen Sie CloudFront Metriken mit Amazon CloudWatch”.
- Informationen zu den Berechtigungen, die zum Senden von Daten an CloudWatch Logs erforderlich sind, finden Sie unterthe section called “Einrichten von IAM-Berechtigungen und -Rollen”.
- Weitere Informationen zum Hinzufügen der Protokollierung zu einer Lambda@Edge-Funktion finden Sie unter AWS Lambda -Funktionsprotokollierung in Node.js oder AWS Lambda -Funktionsprotokollierung in Python im AWS Lambda -Entwicklerhandbuch.
- Informationen zu CloudWatch Logs-Kontingenten (früher als Limits bezeichnet) finden Sie unter CloudWatch Logs-Kontingente im Amazon CloudWatch Logs-Benutzerhandbuch.

CloudFront Funktionen, Protokolle

Wenn der Code einer CloudFront Funktion `console.log()` Anweisungen enthält, sendet CloudFront Functions diese Protokollzeilen automatisch an CloudWatch Logs. Wenn es keine `console.log()` Anweisungen gibt, wird nichts an CloudWatch Logs gesendet.

CloudFront Functions erstellt immer Log-Streams in der Region USA Ost (Nord-Virginia) (`us-east-1`), unabhängig davon, an welchem Edge-Standort die Funktion ausgeführt wurde. Der Name des Protokollstreams ist im Format `YYYY/M/D/`*`UUID`*.

Der Name der Protokollgruppe verwendet das folgende Format:

- Für CloudFront Funktionen auf der Cache-Verhaltensebene lautet das Format `/aws/cloudfront/function/`*`<FunctionName>`*
- Für CloudFront Funktionen auf der Verteilungsebene (Verbindungsfunktionen) lautet das Format `/aws/cloudfront/connection-function/`*`<FunctionName>`*

Das *`<FunctionName>`* ist der Name, den Sie der Funktion gegeben haben, als Sie sie erstellt haben.

Example Anfragen des Betrachters

Im Folgenden sehen Sie ein Beispiel für eine Protokollnachricht, die an CloudWatch Logs gesendet wurde. Jede Zeile beginnt mit einer ID, die eine CloudFront Anfrage eindeutig identifiziert. Die Nachricht beginnt mit einer `START` Zeile, die die CloudFront Verteilungs-ID enthält, und endet mit einer `END` Zeile. Zwischen den Zeilen `END` und `START` befinden sich die durch `console.log()`-Anweisungen in der Funktion erzeugten Protokollzeilen.

```
U7b4hR_RaxMADupvKAvr8_m9gsGXvioUggLV5Oyq-vmAtH8HADpjhw== START DistributionID:
 E3E5D42GADAXZZ
U7b4hR_RaxMADupvKAvr8_m9gsGXvioUggLV5Oyq-vmAtH8HADpjhw== Example function log output
U7b4hR_RaxMADupvKAvr8_m9gsGXvioUggLV5Oyq-vmAtH8HADpjhw== END
```

Example Verbindungsanfragen

Im Folgenden sehen Sie ein Beispiel für eine Protokollnachricht, die an CloudWatch Logs gesendet wurde. Jede Zeile beginnt mit einer ID, die eine CloudFront Anfrage eindeutig identifiziert. Die Nachricht beginnt mit einer `START` Zeile, die die CloudFront Verteilungs-ID enthält, und endet mit einer `END` Zeile. Zwischen den Zeilen `END` und `START` befinden sich die durch `console.log()`-Anweisungen in der Funktion erzeugten Protokollzeilen.

```
U7b4hR_RaxMADupvKAvr8_m9gsGXvioUggLV5Oyq-vmAtH8HADpjhw== START DistributionID:
 E3E5D42GADA123
U7b4hR_RaxMADupvKAvr8_m9gsGXvioUggLV5Oyq-vmAtH8HADpjhw== 1.2.3.4
U7b4hR_RaxMADupvKAvr8_m9gsGXvioUggLV5Oyq-vmAtH8HADpjhw== END
```

> **Note**
>
> CloudFront Functions sendet CloudWatch nur Protokolle für Funktionen in der `LIVE` Phase, die als Reaktion auf Produktionsanfragen und -antworten ausgeführt werden. Wenn Sie eine Funktion testen, sendet CloudFront keine Protokolle an CloudWatch. Die Testausgabe enthält Informationen über Fehler, Rechenauslastung und Funktionsprotokolle (`console.log()`Anweisungen), aber diese Informationen werden nicht an gesendet CloudWatch.

CloudFront Functions verwendet eine dienstbezogene AWS Identity and Access Management (IAM-) Rolle, um Protokolle an Logs in Ihrem CloudWatch Konto zu senden. Eine serviceverknüpfte Rolle ist ein spezieller Typ von IAM-Rolle, die direkt mit einem AWS-Service verknüpft ist. Mit Diensten verknüpfte Rollen sind vom Dienst vordefiniert und enthalten alle Berechtigungen, die der Dienst benötigt, um andere AWS-Services Rollen für Sie aufzurufen. CloudFront Functions verwendet die AWSServiceRoleForCloudFrontLoggerdienstverknüpfte Rolle. Weitere Informationen zu dieser Rolle finden Sie unter the section called “Service-linked Rollen für Lambda @Edge” (Lambda@Edge verwendet dieselbe serviceverknüpfte Rolle).

Wenn eine Funktion aufgrund eines Überprüfungs- oder Ausführungsfehlers fehlschlägt, werden die Informationen in Standardprotokollen und Echtzeit-Zugriffsprotokollen protokolliert. Genauere Informationen zum Fehler finden Sie in den Feldern `x-edge-result-type`, `x-edge-response-result-type` und `x-edge-detailed-result-type`.

Protokollieren Amazon CloudFront Amazon-API-Aufrufen mit AWS CloudTrail

CloudFront ist integriert in AWS CloudTrail, ein Service, der eine Aufzeichnung der Aktionen bereitstellt, die von einem Benutzer, einer Rolle oder einem ausgeführt wurden AWS-Service. CloudTrail erfasst alle API-Aufrufe CloudFront als Ereignisse. Zu den erfassten Aufrufen gehören Aufrufe von der CloudFront Konsole und Codeaufrufen für die CloudFront API-Operationen. Anhand der von gesammelten Informationen können Sie die Anfrage CloudTrail, an die die Anfrage gestellt wurde CloudFront, die IP-Adresse, von der aus die Anfrage gestellt wurde, den Zeitpunkt der Anfrage und weitere Details ermitteln.

Jeder Ereignis- oder Protokolleintrag enthält Informationen zu dem Benutzer, der die Anforderung generiert hat. Die Identitätsinformationen unterstützen Sie bei der Ermittlung der folgenden Punkte:

- Ob die Anforderung mit Anmeldeinformationen des Root-Benutzers oder des Benutzers gestellt wurde.
- Die Anforderung wurde im Namen eines IAM-Identity-Center-Benutzers erstellt.
- Gibt an, ob die Anforderung mit temporären Sicherheitsanmeldeinformationen für eine Rolle oder einen Verbundbenutzer gesendet wurde.
- Ob die Anforderung aus einem anderen AWS-Service gesendet wurde.

CloudTrail ist in Ihrem aktiv AWS-Konto , wenn Sie das Konto erstellen, und Sie haben automatisch Zugriff auf den CloudTrail Eventverlauf. Der CloudTrail Ereignisverlauf bietet eine einsehbare, durchsuchbare, herunterladbare und unveränderliche Aufzeichnung der aufgezeichneten Verwaltungsereignisse der letzten 90 Tage in einem. AWS-Region Weitere Informationen finden Sie im AWS CloudTrail Benutzerhandbuch unter Arbeiten mit dem CloudTrail Ereignisverlauf. Für die Anzeige des Eventverlaufs CloudTrail fallen keine Gebühren an.

Für eine fortlaufende Aufzeichnung der Ereignisse in AWS-Konto den letzten 90 Tagen erstellen Sie einen Trail- oder CloudTrailLake-Event-Datenspeicher.

CloudTrail Pfade

Ein Trail ermöglicht CloudTrail die Übermittlung von Protokolldateien an einen Amazon S3 S3-Bucket. Alle mit dem erstellten Pfade AWS-Managementkonsole sind regionsübergreifend. Sie können mithilfe von AWS CLI einen Einzel-Region- oder einen Multi-Region-Trail erstellen. Es wird empfohlen, einen Trail mit mehreren Regionen zu erstellen, da Sie alle Aktivitäten AWS-Regionen in Ihrem Konto erfassen. Wenn Sie einen Einzel-Region-Trail erstellen, können Sie nur die Ereignisse anzeigen, die im AWS-Region des Trails protokolliert wurden. Weitere Informationen zu Trails finden Sie unter Erstellen eines Trails für Ihr AWS-Konto und Erstellen eines Trails für eine Organisation im AWS CloudTrail -Benutzerhandbuch.

Sie können eine Kopie Ihrer laufenden Verwaltungsereignisse kostenlos an Ihren Amazon S3 S3-Bucket senden, CloudTrail indem Sie einen Trail erstellen. Es fallen jedoch Amazon S3 S3-Speichergebühren an. Weitere Informationen zur CloudTrail Preisgestaltung finden Sie unter AWS CloudTrail Preise. Informationen zu Amazon-S3-Preisen finden Sie unter Amazon S3 – Preise.

CloudTrail Datenspeicher für Ereignisse in Lake

CloudTrail Mit Lake können Sie SQL-basierte Abfragen für Ihre Ereignisse ausführen. CloudTrail Lake konvertiert bestehende Ereignisse im zeilenbasierten JSON-Format in das Apache ORC-Format. ORC ist ein spaltenförmiges Speicherformat, das für den schnellen Abruf von Daten optimiert ist. Die Ereignisse werden in Ereignisdatenspeichern zusammengefasst, bei denen es

sich um unveränderliche Sammlungen von Ereignissen handelt, die auf Kriterien basieren, die Sie mit Hilfe von erweiterten Ereignisselektoren auswählen. Die Selektoren, die Sie auf einen Ereignisdatenspeicher anwenden, steuern, welche Ereignisse bestehen bleiben und für Sie zur Abfrage verfügbar sind. Weitere Informationen zu CloudTrail Lake finden Sie unter Arbeiten mit AWS CloudTrail Lake im AWS CloudTrail Benutzerhandbuch.

CloudTrail Für das Speichern und Abfragen von Ereignisdaten in Lake fallen Kosten an. Beim Erstellen eines Ereignisdatenspeichers wählen Sie die Preisoption aus, die für den Ereignisdatenspeicher genutzt werden soll. Die Preisoption bestimmt die Kosten für die Erfassung und Speicherung von Ereignissen sowie die standardmäßige und maximale Aufbewahrungsdauer für den Ereignisdatenspeicher. Weitere Informationen zur CloudTrail Preisgestaltung finden Sie unter AWS CloudTrail Preise.

> **Note**
>
> CloudFront ist ein globaler Service. CloudTrail zeichnet Ereignisse CloudFront in der Region USA Ost (Nord-Virginia) auf. Weitere Informationen finden Sie unter Globale Serviceereignisse im Benutzerhandbuch für AWS CloudTrail .
> Wenn Sie temporäre Sicherheitsanmeldeinformationen verwenden AWS -Security-Token-Service, werden Anrufe an regionale Endpunkte, z. B.`us-west-2`, in der CloudTrail entsprechenden Region angemeldet.
> Weitere Informationen zu CloudFront Endpunkten finden Sie unter CloudFront Endpunkte und Kontingente in der. Allgemeine AWS-Referenz

CloudFront Datenereignisse in CloudTrail

Datenereignisse liefern Informationen über die Ressourcenoperationen, die auf oder in einer Ressource ausgeführt werden (z. B. Lesen oder Schreiben in eine CloudFront Distribution). Sie werden auch als Vorgänge auf Datenebene bezeichnet. Datenereignisse sind oft Aktivitäten mit hohem Volume. Protokolliert standardmäßig CloudTrail keine Datenereignisse. Der CloudTrail Ereignisverlauf zeichnet keine Datenereignisse auf.

Für Datenereignisse werden zusätzliche Gebühren fällig. Weitere Informationen zur CloudTrail Preisgestaltung finden Sie unter AWS CloudTrail Preisgestaltung.

Sie können Datenereignisse für die CloudFront Ressourcentypen mithilfe der CloudTrail Konsole oder CloudTrail API-Operationen protokollieren. AWS CLI Weitere Informationen zum Protokollieren

von Datenereignissen finden Sie unter Protokollieren von Datenereignissen mit dem AWS-Managementkonsole und Protokollieren von Datenereignissen mit dem AWS Command Line Interface im AWS CloudTrail -Benutzerhandbuch.

In der folgenden Tabelle sind die CloudFront Ressourcentypen aufgeführt, für die Sie Datenereignisse protokollieren können. In der Spalte Datenereignistyp (Konsole) wird der Wert angezeigt, den Sie in der Liste Datenereignistyp auf der CloudTrail Konsole auswählen können. In der Wertspalte resources.type wird der `resources.type` Wert angezeigt, den Sie bei der Konfiguration erweiterter Event-Selektoren mithilfe von oder angeben würden. AWS CLI CloudTrail APIs In der CloudTrail Spalte APIs Protokollierte Daten werden die API-Aufrufe angezeigt, die CloudTrail für den Ressourcentyp protokolliert wurden.

Typ des Datenereignisses (Konsole)	resources.type-Wert	Daten, die APIs protokolliert wurden CloudTrail
CloudFront KeyValueStore	`AWS::CloudFront::KeyValueStore`	• DeleteKeys • DescribeKeyValueStore • GetKey • ListKeys • PutKeys • UpdateKeys

Sie können erweiterte Event-Selektoren so konfigurieren, dass sie nach den Feldern `eventName`, `readOnly` und `resources.ARN` filtern, sodass nur die Ereignisse protokolliert werden, die für Sie wichtig sind. Weitere Informationen zu diesen Kontingenten finden Sie unter AdvancedFieldSelector in derAWS CloudTrail -API-Referenz.

CloudFront Verwaltungsereignisse in CloudTrail

Verwaltungsereignisse bieten Informationen über Verwaltungsvorgänge, die an Ressourcen in Ihrem ausgeführt werden AWS-Konto. Sie werden auch als Vorgänge auf Steuerebene bezeichnet. CloudTrail Protokolliert standardmäßig Verwaltungsereignisse.

Amazon CloudFront protokolliert alle Operationen auf der CloudFront Kontrollebene als Managementereignisse. Eine Liste der Vorgänge auf der CloudFront Amazon-Kontrollebene, bei denen CloudFront sich angemeldet wird CloudTrail, finden Sie in der Amazon CloudFront API-Referenz.

CloudFront Beispiele für Ereignisse

Ein Ereignis stellt eine einzelne Anfrage aus einer beliebigen Quelle dar und enthält Informationen über den angeforderten API-Vorgang, Datum und Uhrzeit des Vorgangs, Anforderungsparameter usw. CloudTrail Protokolldateien sind kein geordneter Stack-Trace der öffentlichen API-Aufrufe, sodass Ereignisse nicht in einer bestimmten Reihenfolge angezeigt werden.

Inhalt

- Beispiel: UpdateDistribution
- Beispiel: UpdateKeys

Beispiel: UpdateDistribution

Das folgende Beispiel zeigt ein CloudTrail Ereignis, das den UpdateDistributionVorgang demonstriert.

Für CloudFront API-Aufrufe `eventSource` ist der`cloudfront.amazonaws.com`.


```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AIDACKCEVSQ6C2EXAMPLE:role-session-name",
        "arn": "arn:aws:sts::111122223333:assumed-role/Admin/role-session-name",
        "accountId": "111122223333",
        "accessKeyId": "ASIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AIDACKCEVSQ6C2EXAMPLE",
                "arn": "arn:aws:iam::111122223333:role/Admin",
                "accountId": "111122223333",
                "userName": "Admin"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2024-02-02T19:23:50Z",
                "mfaAuthenticated": "false"
            }
        }
    },
    "eventTime": "2024-02-02T19:26:01Z",
    "eventSource": "cloudfront.amazonaws.com",
```

```
    "eventName": "UpdateDistribution",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "52.94.133.137",
    "userAgent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
 like Gecko) Chrome/121.0.0.0 Safari/537.36",
    "requestParameters": {
        "distributionConfig": {
            "defaultRootObject": "",
            "aliases": {
                "quantity": 3,
                "items": [
                    "alejandro_rosalez.awsps.myinstance.com",
                    "cross-testing.alejandro_rosalez.awsps.myinstance.com",
                    "*.alejandro_rosalez.awsps.myinstance.com"
                ]
            },
            "cacheBehaviors": {
                "quantity": 0,
                "items": []
            },
            "httpVersion": "http2and3",
            "originGroups": {
                "quantity": 0,
                "items": []
            },
            "viewerCertificate": {
                "minimumProtocolVersion": "TLSv1.2_2021",
                "cloudFrontDefaultCertificate": false,
                "aCMCertificateArn": "arn:aws:acm:us-east-1:111122223333:certificate/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
                "sSLSupportMethod": "sni-only"
            },
            "webACLId": "arn:aws:wafv2:us-east-1:111122223333:global/webacl/testing-
acl/a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
            "customErrorResponses": {
                "quantity": 0,
                "items": []
            },
            "logging": {
                "includeCookies": false,
                "prefix": "",
                "enabled": false,
                "bucket": ""
            },
```

```
            "priceClass": "PriceClass_All",
            "restrictions": {
                "geoRestriction": {
                    "restrictionType": "none",
                    "quantity": 0,
                    "items": []
                }
            },
            "isIPV6Enabled": true,
            "callerReference": "1578329170895",
            "continuousDeploymentPolicyId": "",
            "enabled": true,
            "defaultCacheBehavior": {
                "targetOriginId": "d111111abcdef8",
                "minTTL": 0,
                "compress": false,
                "maxTTL": 31536000,
                "functionAssociations": {
                    "quantity": 0,
                    "items": []
                },
                "trustedKeyGroups": {
                    "quantity": 0,
                    "items": [],
                    "enabled": false
                },
                "smoothStreaming": false,
                "fieldLevelEncryptionId": "",
                "defaultTTL": 86400,
                "lambdaFunctionAssociations": {
                    "quantity": 0,
                    "items": []
                },
                "viewerProtocolPolicy": "redirect-to-https",
                "forwardedValues": {
                    "cookies": {"forward": "none"},
                    "queryStringCacheKeys": {
                        "quantity": 0,
                        "items": []
                    },
                    "queryString": false,
                    "headers": {
                        "quantity": 1,
                        "items": ["*"]
```

```
                    }
                },
                "trustedSigners": {
                    "items": [],
                    "enabled": false,
                    "quantity": 0
                },
                "allowedMethods": {
                    "quantity": 2,
                    "items": [
                        "HEAD",
                        "GET"
                    ],
                    "cachedMethods": {
                        "quantity": 2,
                        "items": [
                            "HEAD",
                            "GET"
                        ]
                    }
                }
            },
            "staging": false,
            "origins": {
                "quantity": 1,
                "items": [
                    {
                        "originPath": "",
                        "connectionTimeout": 10,
                        "customOriginConfig": {
                            "originReadTimeout": 30,
                            "hTTPSPort": 443,
                            "originProtocolPolicy": "https-only",
                            "originKeepaliveTimeout": 5,
                            "hTTPPort": 80,
                            "originSslProtocols": {
                                "quantity": 3,
                                "items": [
                                    "TLSv1",
                                    "TLSv1.1",
                                    "TLSv1.2"
                                ]
                            }
                        },
```

```
                        "id": "d111111abcdef8",
                        "domainName": "d111111abcdef8.cloudfront.net",
                        "connectionAttempts": 3,
                        "customHeaders": {
                            "quantity": 0,
                            "items": []
                        },
                        "originShield": {"enabled": false},
                        "originAccessControlId": ""
                    }
                ]
            },
            "comment": "HIDDEN_DUE_TO_SECURITY_REASONS"
        },
        "id": "EDFDVBD6EXAMPLE",
        "ifMatch": "E1RTLUR9YES76O"
    },
    "responseElements": {
        "distribution": {
            "activeTrustedSigners": {
                "quantity": 0,
                "enabled": false
            },
            "id": "EDFDVBD6EXAMPLE",
            "domainName": "d111111abcdef8.cloudfront.net",
            "distributionConfig": {
                "defaultRootObject": "",
                "aliases": {
                    "quantity": 3,
                    "items": [
                        "alejandro_rosalez.awsps.myinstance.com",
                        "cross-testing.alejandro_rosalez.awsps.myinstance.com",
                        "*.alejandro_rosalez.awsps.myinstance.com"
                    ]
                },
                "cacheBehaviors": {"quantity": 0},
                "httpVersion": "http2and3",
                "originGroups": {"quantity": 0},
                "viewerCertificate": {
                    "minimumProtocolVersion": "TLSv1.2_2021",
                    "cloudFrontDefaultCertificate": false,
                    "aCMCertificateArn": "arn:aws:acm:us-
east-1:111122223333:certificate/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
                    "sSLSupportMethod": "sni-only",
```

```
                    "certificateSource": "acm",
                    "certificate": "arn:aws:acm:us-east-1:111122223333:certificate/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
                },
                "webACLId": "arn:aws:wafv2:us-east-1:111122223333:global/webacl/
testing-acl/a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
                "customErrorResponses": {"quantity": 0},
                "logging": {
                    "includeCookies": false,
                    "prefix": "",
                    "enabled": false,
                    "bucket": ""
                },
                "priceClass": "PriceClass_All",
                "restrictions": {
                    "geoRestriction": {
                        "restrictionType": "none",
                        "quantity": 0
                    }
                },
                "isIPV6Enabled": true,
                "callerReference": "1578329170895",
                "continuousDeploymentPolicyId": "",
                "enabled": true,
                "defaultCacheBehavior": {
                    "targetOriginId": "d111111abcdef8",
                    "minTTL": 0,
                    "compress": false,
                    "maxTTL": 31536000,
                    "functionAssociations": {"quantity": 0},
                    "trustedKeyGroups": {
                        "quantity": 0,
                        "enabled": false
                    },
                    "smoothStreaming": false,
                    "fieldLevelEncryptionId": "",
                    "defaultTTL": 86400,
                    "lambdaFunctionAssociations": {"quantity": 0},
                    "viewerProtocolPolicy": "redirect-to-https",
                    "forwardedValues": {
                        "cookies": {"forward": "none"},
                        "queryStringCacheKeys": {"quantity": 0},
                        "queryString": false,
                        "headers": {
```

```
                        "quantity": 1,
                        "items": ["*"]
                    }
                },
                "trustedSigners": {
                    "enabled": false,
                    "quantity": 0
                },
                "allowedMethods": {
                    "quantity": 2,
                    "items": [
                        "HEAD",
                        "GET"
                    ],
                    "cachedMethods": {
                        "quantity": 2,
                        "items": [
                            "HEAD",
                            "GET"
                        ]
                    }
                }
            },
            "staging": false,
            "origins": {
                "quantity": 1,
                "items": [
                    {
                        "originPath": "",
                        "connectionTimeout": 10,
                        "customOriginConfig": {
                            "originReadTimeout": 30,
                            "hTTPSPort": 443,
                            "originProtocolPolicy": "https-only",
                            "originKeepaliveTimeout": 5,
                            "hTTPPort": 80,
                            "originSslProtocols": {
                                "quantity": 3,
                                "items": [
                                    "TLSv1",
                                    "TLSv1.1",
                                    "TLSv1.2"
                                ]
                            }
```

```
                        },
                        "id": "d111111abcdef8",
                        "domainName": "d111111abcdef8.cloudfront.net",
                        "connectionAttempts": 3,
                        "customHeaders": {"quantity": 0},
                        "originShield": {"enabled": false},
                        "originAccessControlId": ""
                    }
                ]
            },
            "comment": "HIDDEN_DUE_TO_SECURITY_REASONS"
        },
        "aliasICPRecordals": [
            {
                "cNAME": "alejandro_rosalez.awsps.myinstance.com",
                "iCPRecordalStatus": "APPROVED"
            },
            {
                "cNAME": "cross-testing.alejandro_rosalez.awsps.myinstance.com",
                "iCPRecordalStatus": "APPROVED"
            },
            {
                "cNAME": "*.alejandro_rosalez.awsps.myinstance.com",
                "iCPRecordalStatus": "APPROVED"
            }
        ],
        "aRN": "arn:aws:cloudfront::111122223333:distribution/EDFDVBD6EXAMPLE",
        "status": "InProgress",
        "lastModifiedTime": "Feb 2, 2024 7:26:01 PM",
        "activeTrustedKeyGroups": {
            "enabled": false,
            "quantity": 0
        },
        "inProgressInvalidationBatches": 0
    },
    "eTag": "E1YHBLAB2BJY1G"
},
"requestID": "4e6b66f9-d548-11e3-a8a9-73e33example",
"eventID": "5ab02562-0fc5-43d0-b7b6-90293example",
"readOnly": false,
"eventType": "AwsApiCall",
"apiVersion": "2020_05_31",
"managementEvent": true,
"recipientAccountId": "111122223333",
```

```
    "eventCategory": "Management",
    "tlsDetails": {
        "tlsVersion": "TLSv1.3",
        "cipherSuite": "TLS_AES_128_GCM_SHA256",
        "clientProvidedHostHeader": "cloudfront.amazonaws.com"
    },
    "sessionCredentialFromConsole": "true"
}
```

Beispiel: UpdateKeys

Das folgende Beispiel zeigt ein CloudTrail Ereignis, das den UpdateKeysVorgang demonstriert.

Bei CloudFront KeyValueStore API-Aufrufen `eventSource` ist `edgekeyvaluestore.amazonaws.com` stattcloudfront.amazonaws.com.

```
{
    "eventVersion": "1.09",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AIDACKCEVSQ6C2EXAMPLE:role-session-name",
        "arn": "arn:aws:sts::111122223333:assumed-role/Admin/role-session-name",
        "accountId": "111122223333",
        "accessKeyId": "ASIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AIDACKCEVSQ6C2EXAMPLE",
                "arn": "arn:aws:iam::111122223333:role/Admin",
                "accountId": "111122223333",
                "userName": "Admin"
            },
            "attributes": {
                "creationDate": "2023-11-01T23:41:14Z",
                "mfaAuthenticated": "false"
            }
        }
    },
    "eventTime": "2023-11-01T23:41:28Z",
    "eventSource": "edgekeyvaluestore.amazonaws.com",
    "eventName": "UpdateKeys",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "3.235.183.252",
```

```
    "userAgent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
 like Gecko) Chrome/121.0.0.0 Safari/537.36,
    "requestParameters": {
        "kvsARN": "arn:aws:cloudfront::111122223333:key-value-store/a1b2c3d4-5678-90ab-
cdef-EXAMPLE11111",
        "ifMatch": "KV3O6B1CX531EBP",
        "deletes": [
            {"key": "key1"}
        ]
    },
    "responseElements": {
        "itemCount": 0,
        "totalSizeInBytes": 0,
        "eTag": "KVDC9VEVZ71ZGO"
    },
    "requestID": "5ccf104c-acce-4ea1-b7fc-73e33example",
    "eventID": "a0b1b5c7-906c-439d-9925-90293example",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::CloudFront::KeyValueStore",
            "ARN": "arn:aws:cloudfront::111122223333:key-value-store/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": false,
    "recipientAccountId": "111122223333",
    "eventCategory": "Data",
    "tlsDetails": {
        "tlsVersion": "TLSv1.3",
        "cipherSuite": "TLS_AES_128_GCM_SHA256",
        "clientProvidedHostHeader": "111122223333.cloudfront-kvs.global.api.aws"
    }
}
```


Informationen zu CloudTrail Datensatzinhalten finden Sie im AWS CloudTrail Benutzerhandbuch unter CloudTrail Datensatzinhalte.

Verfolgen Sie Konfigurationsänderungen mit AWS Config

Um Konfigurationen Ihrer AWS Ressourcen aufzuzeichnen und auszuwerten, können Sie diese AWS Config Funktion verwenden, die Ihnen einen detaillierten Überblick über die Konfiguration Ihrer Distributionen bietet. Dazu gehört auch, wie die Ressourcen jeweils zueinander in Beziehung stehen und wie sie in der Vergangenheit konfiguriert wurden, damit Sie Änderungen im Laufe der Zeit einsehen können.

Sie können es auch verwenden AWS Config , um Konfigurationsänderungen an Ihren CloudFront Distributionseinstellungen aufzuzeichnen. Sie können Distributionsstatusänderungen sowie Änderungen an Preisklassen, Ursprüngen, geografischen Einschränkungseinstellungen und Lambda@Edge-Konfigurationen erfassen.

> **Note**
>
> AWS Config zeichnet keine Schlüssel-Wert-Tags für CloudFront Streaming-Verteilungen auf.

Inhalt

- Richten Sie ein mit AWS Config CloudFront
- CloudFront Konfigurationshistorie anzeigen
- Evaluieren Sie CloudFront Konfigurationen mit AWS Config Regeln

Richten Sie ein mit AWS Config CloudFront

Bei der Einrichtung können Sie wählen AWS Config, ob Sie alle unterstützten AWS Ressourcen oder nur einige bestimmte Ressourcen aufzeichnen möchten, z. B. CloudFront nur Änderungen für. Eine Liste der unterstützten CloudFront Ressourcen finden Sie im CloudFront Abschnitt Amazon des Themas Unterstützte Ressourcentypen im AWS Config Developer Guide.

> **Hinweise**
>
> - Um die Konfigurationsänderungen an Ihrer CloudFront Distribution nachzuverfolgen, müssen Sie sich bei der CloudFront Konsole im Osten der USA (Nord-Virginia) anmelden AWS-Region.

- Es kann zu einer Verzögerung bei der Aufzeichnung von Ressourcen mit kommen AWS Config. AWS Config zeichnet Ressourcen erst auf, nachdem sie erkannt wurden.

Console

Zur Einrichtung mit AWS Config CloudFront

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die AWS Config Konsole.
2. Wählen Sie Get Started Now.
3. Geben Sie auf der Seite Einstellungen unter Aufzuzeichnende Ressourcentypen die AWS Ressourcentypen an, die Sie aufzeichnen AWS Config möchten. Wenn Sie nur CloudFront Änderungen aufzeichnen möchten, wählen Sie Bestimmte Typen und dann unter die Verteilung oder Streaming-Verteilung aus CloudFront, für die Sie Änderungen verfolgen möchten.

   Wählen Sie nach der Ersteinrichtung zum Hinzufügen oder Ändern der nachzuverfolgenden Verteilungen Einstellungen auf der linken Seite aus.
4. Geben Sie zusätzliche erforderliche Optionen an für AWS Config: eine Benachrichtigung einrichten, einen Speicherort für die Konfigurationsinformationen angeben und Regeln für die Bewertung von Ressourcentypen hinzufügen.

Weitere Informationen finden Sie unter Einrichtung AWS Config mit der Konsole im AWS Config Entwicklerhandbuch.

AWS CLI

Informationen zur Einrichtung AWS Config CloudFront mit der AWS CLI finden Sie unter Einrichtung AWS Config mit der AWS CLI im AWS Config Entwicklerhandbuch.

AWS Config API

Informationen zur Einrichtung AWS Config mit der CloudFront Verwendung der AWS Config API finden Sie unter StartConfigurationRecorderAPI-Vorgang in der AWS Config API-Referenz.

CloudFront Konfigurationshistorie anzeigen

Nachdem Sie mit der Aufzeichnung von Konfigurationsänderungen an Ihren Distributionen AWS Config begonnen haben, können Sie den Konfigurationsverlauf jeder Distribution abrufen, für CloudFront die Sie konfiguriert haben.

Sie können die Konfigurationsverläufe auf die folgenden Arten anzeigen.

Console

Für jede aufgenommene Ressource können Sie eine Timeline-Seite anzeigen, die einen Verlauf der Konfigurationsdetails bietet. Wählen Sie zum Anzeigen dieser Seite das graue Symbol in der Spalte Config Timeline (Konfig.-Timeline) der Seite Dedicated Hosts.

Weitere Informationen finden Sie im AWS Config Entwicklerhandbuch unter Konfigurationsdetails in der AWS Config Konsole anzeigen.

AWS CLI

Um eine Liste all Ihrer Distributionen abzurufen, führen Sie den list-discovered-resourcesBefehl aus, wie im folgenden Beispiel gezeigt.

```
aws configservice list-discovered-resources --resource-type
 AWS::CloudFront::Distribution
```

Führen Sie den get-resource-config-historyBefehl aus, um die Konfigurationsdetails einer Distribution für ein bestimmtes Zeitintervall abzurufen.

Weitere Informationen finden Sie unter View Configuration Details Using the CLI (Anzeigen von Konfigurationsdetails mithilfe der Befehlszeilenschnittstelle (CLI)) im AWS Config -Developer-Handbuch.

AWS Config API

Verwenden Sie den ListDiscoveredResourcesAPI-Vorgang, um eine Liste all Ihrer Distributionen abzurufen.

Verwenden Sie den GetResourceConfigHistoryAPI-Vorgang, um die Konfigurationsdetails einer Distribution für ein bestimmtes Zeitintervall abzurufen. Weitere Informationen finden Sie in der AWS Config -API-Referenz.

Evaluieren Sie CloudFront Konfigurationen mit AWS Config Regeln

Mithilfe von AWS Config Regeln können Sie Konfigurationen anhand der gewünschten Konfigurationen auswerten. Mithilfe von AWS Config Regeln können Sie beispielsweise beurteilen, ob Ihre CloudFront Ressourcen den gängigen bewährten Sicherheitsmethoden entsprechen. Sie können verwaltete Regeln wie Viewer-Richtlinie HTTPS, SNI aktiviert, OAC aktiviert, Origin Failover aktiviert, AWS WAF WebACL oder AWS Shield Advanced Ressourcenrichtlinien auswählen, die ausgelöst werden sollen, wenn sich die Konfiguration ändert.

Mit verwalteten Regeln können in regelmäßigen Abständen Evaluierungen mit einer von Ihnen festgelegten Häufigkeit durchgeführt werden. AWS Firewall Manager stützt sich auf AWS Config automatische Warnmeldungen und Abhilfemaßnahmen. Weitere Informationen finden Sie unter Ressourcen anhand von AWS Config Regeln evaluieren und Liste AWS Config verwalteter Regeln im AWS Config Entwicklerhandbuch.

Sicherheit bei Amazon CloudFront

Cloud-Sicherheit AWS hat höchste Priorität. Als AWS Kunde profitieren Sie von einer Rechenzentrums- und Netzwerkarchitektur, die auf die Anforderungen der sicherheitsrelevantesten Organisationen zugeschnitten sind.

Sicherheit ist eine gemeinsame Verantwortung von Ihnen AWS und Ihnen. Das Modell der geteilten Verantwortung beschreibt dies als Sicherheit der Cloud und Sicherheit in der Cloud:

- Sicherheit der Cloud — AWS ist für den Schutz der Infrastruktur verantwortlich, auf der AWS Dienste in der AWS Cloud ausgeführt werden. AWS bietet Ihnen auch Dienste, die Sie sicher nutzen können. Third-party Auditoren testen und verifizieren regelmäßig die Wirksamkeit unserer Sicherheitsvorkehrungen im Rahmen der AWS Compliance-Programme. Weitere Informationen zu den Compliance-Programmen, die für Amazon gelten CloudFront, finden Sie unter AWS Services im Umfang nach Compliance-Programm.
- Sicherheit in der Cloud — Ihre Verantwortung hängt von dem AWS Service ab, den Sie nutzen. In Ihre Verantwortung fallen außerdem weitere Faktoren, wie z. B. die Vertraulichkeit der Daten, die Anforderungen Ihrer Organisation sowie geltende Gesetze und Vorschriften.

Diese Dokumentation hilft Ihnen zu verstehen, wie Sie das Modell der gemeinsamen Verantwortung bei der Nutzung anwenden CloudFront. Die folgenden Themen zeigen Ihnen, wie Sie die Konfiguration so einrichten CloudFront , dass Ihre Sicherheits- und Compliance-Ziele erreicht werden. Außerdem erfahren Sie, wie Sie andere AWS Dienste nutzen können, mit denen Sie Ihre CloudFront Ressourcen überwachen und sichern können.

- Datenschutz bei Amazon CloudFront— Schützen Sie Ihre Daten bei der Übertragung und Speicherung.
- Identitäts- und Zugriffsmanagement für Amazon CloudFront— Verwalten Sie den Zugriff auf CloudFront Ressourcen und Berechtigungen.
- Protokollierung und Überwachung in Amazon CloudFront— Verfolgen Sie Anfragen und überwachen Sie den Zustand Ihrer Distributionen.
- Konformitätsüberprüfung für Amazon CloudFront— Erfahren Sie, wie CloudFront Compliance-Programme eingehalten werden.
- Ausfallsicherheit in Amazon CloudFront— Erfahren Sie mehr über die Widerstandsfähigkeit der CloudFront Infrastruktur.

- Infrastruktursicherheit bei Amazon CloudFront— Erfahren Sie mehr über die Infrastruktursicherheit für CloudFront.

Topics

- Datenschutz bei Amazon CloudFront
- Identitäts- und Zugriffsmanagement für Amazon CloudFront
- Protokollierung und Überwachung in Amazon CloudFront
- Konformitätsüberprüfung für Amazon CloudFront
- Ausfallsicherheit in Amazon CloudFront
- Infrastruktursicherheit bei Amazon CloudFront

Datenschutz bei Amazon CloudFront

Das Modell der AWS https://aws.amazon.com/compliance/shared-responsibility-model/ gilt für den Datenschutz bei Amazon CloudFront. Wie in diesem Modell beschrieben, AWS ist es für den Schutz der globalen Infrastruktur verantwortlich, auf der das gesamte System läuft AWS Cloud. Sie sind dafür verantwortlich, die Kontrolle über Ihre in dieser Infrastruktur gehosteten Inhalte zu behalten. Sie sind auch für die Sicherheitskonfiguration und die Verwaltungsaufgaben für die von Ihnen verwendeten AWS-Services verantwortlich. Weitere Informationen zum Datenschutz finden Sie unter Häufig gestellte Fragen zum Datenschutz in der Region . Weitere Informationen zum Datenschutz in Europa finden Sie im Zentrum für die Datenschutz-Grundverordnung (DSGVO).

Aus Datenschutzgründen empfehlen wir, AWS-Konto Anmeldeinformationen zu schützen und einzelne Benutzer mit AWS IAM Identity Center oder AWS Identity and Access Management (IAM) einzurichten. So erhält jeder Benutzer nur die Berechtigungen, die zum Durchführen seiner Aufgaben erforderlich sind. Außerdem empfehlen wir, die Daten mit folgenden Methoden schützen:

- Verwenden Sie für jedes Konto die Multi-Faktor-Authentifizierung (MFA).
- Wird verwendet SSL/TLS , um mit AWS Ressourcen zu kommunizieren. Wir benötigen TLS 1.2 und empfehlen TLS 1.3.
- Richten Sie die API und die Protokollierung von Benutzeraktivitäten mit ein AWS CloudTrail. Informationen zur Verwendung von CloudTrail Pfaden zur Erfassung von AWS Aktivitäten findest du unter Arbeiten mit CloudTrail Pfaden im AWS CloudTrail Benutzerhandbuch.
- Verwenden Sie AWS Verschlüsselungslösungen zusammen mit allen darin enthaltenen Standardsicherheitskontrollen AWS-Services.

- Verwenden Sie erweiterte verwaltete Sicherheitsservices wie Amazon Macie, die dabei helfen, in Amazon S3 gespeicherte persönliche Daten zu erkennen und zu schützen.
- Wenn Sie für den Zugriff AWS über eine Befehlszeilenschnittstelle oder eine API FIPS 140-3-validierte kryptografische Module benötigen, verwenden Sie einen FIPS-Endpunkt. Weitere Informationen über verfügbare FIPS-Endpunkte finden Sie unter Federal Information Processing Standard (FIPS) 140-3.

Wir empfehlen dringend, in Freitextfeldern, z. B. im Feld Name, keine vertraulichen oder sensiblen Informationen wie die E-Mail-Adressen Ihrer Kunden einzugeben. Dies gilt auch, wenn Sie mit der Konsole, der API CloudFront oder den SDKs arbeiten oder diese auf andere Weise AWS-Services verwenden. AWS CLI AWS Alle Daten, die Sie in Tags oder Freitextfelder eingeben, die für Namen verwendet werden, können für Abrechnungs- oder Diagnoseprotokolle verwendet werden. Wenn Sie eine URL für einen externen Server bereitstellen, empfehlen wir dringend, keine Anmeldeinformationen zur Validierung Ihrer Anforderung an den betreffenden Server in die URL einzuschließen.

Amazon CloudFront bietet mehrere Optionen, mit denen Sie die bereitgestellten Inhalte sichern können:

- Konfigurieren von HTTPS-Verbindungen.
- Konfigurieren Sie die Verschlüsselung auf Feldebene, um während der Übertragung zusätzliche Sicherheit für bestimmte Daten zu bieten.
- Beschränken des Zugriffs auf Inhalte, sodass nur bestimmte Personen oder Personen eines bestimmten Bereichs diese sehen können.

Die folgenden Themen erläutern die Optionen im Detail.

Themen

- Verschlüsselung während der Übertragung
- Verschlüsselung im Ruhezustand
- Einschränken des Zugriffs auf Inhalte

Verschlüsselung während der Übertragung

Um Ihre Daten während der Übertragung zu verschlüsseln, konfigurieren Sie Amazon CloudFront so, dass Zuschauer HTTPS verwenden müssen, um Ihre Dateien anzufordern, sodass Verbindungen

bei der CloudFront Kommunikation mit Zuschauern verschlüsselt werden. Sie können auch so konfigurieren CloudFront , dass HTTPS verwendet wird, um Dateien von Ihrem Absender abzurufen, sodass Verbindungen bei der CloudFront Kommunikation mit Ihrem Absender verschlüsselt werden.

Weitere Informationen finden Sie unter Verwenden Sie HTTPS mit CloudFront.

Field-level Verschlüsselung bietet neben HTTPS eine zusätzliche Sicherheitsebene, mit der Sie bestimmte Daten während der gesamten Systemverarbeitung schützen können, sodass nur bestimmte Anwendungen sie sehen können. Durch die Konfiguration der Verschlüsselung auf Feldebene können Sie vertrauliche Informationen CloudFront, die von Benutzern übermittelt wurden, sicher auf Ihre Webserver hochladen. Die sensiblen Informationen, die Ihnen von Clients zur Verfügung gestellt werden, werden an dem Edge-Standort verschlüsselt, der näher am Benutzer ist. Es bleibt in Ihrem gesamten Anwendungsstapel verschlüsselt, sodass nur Anwendungen, die die Daten benötigen — und über die Anmeldeinformationen zu ihrer Entschlüsselung verfügen — dazu in der Lage sind.

Weitere Informationen finden Sie unter Vertrauliche Daten durch Verschlüsselung auf Feldebene schützen.

Die CloudFront API-Endpunkte `cloudfront.amazonaws.com` und, akzeptieren nur HTTPS-Verkehr`cloudfront-fips.amazonaws.com`. Das bedeutet, dass Ihre Daten bei der Übertragung immer verschlüsselt werden, wenn Sie Informationen über die CloudFront API senden und empfangen. Zu diesen Daten gehören Verteilungskonfigurationen, Cache-Richtlinien und Richtlinien für Quellanfragen, Schlüsselgruppen und öffentliche Schlüssel sowie Funktionscode in CloudFront Functions. Darüber hinaus werden alle an die CloudFront API-Endpunkte gesendeten Anfragen mit AWS Anmeldeinformationen signiert und angemeldet AWS CloudTrail.

Funktionscode und Konfiguration in CloudFront Functions werden bei der Übertragung immer verschlüsselt, wenn sie auf die PoPs (Edge Location Points of Presence) und zwischen anderen Speicherorten, die von verwendet werden, kopiert werden. CloudFront

Verschlüsselung im Ruhezustand

Der Funktionscode und die Konfiguration in CloudFront Functions werden immer in einem verschlüsselten Format auf den PoPs am Edge-Standort und an anderen Speicherorten gespeichert, die von verwendet werden. CloudFront

Einschränken des Zugriffs auf Inhalte

Viele Unternehmen, die Inhalte über das Internet bereitstellen, möchten den Zugriff auf Dokumente, Geschäftsdaten, Medien-Streams oder Inhalte, die nur für ausgewählte Benutzer gedacht sind, beschränken. Um diesen Inhalt mithilfe von Amazon sicher bereitzustellen CloudFront, können Sie einen oder mehrere der folgenden Schritte ausführen:

Verwenden signierter URLs oder Cookies

Sie können den Zugriff auf Inhalte einschränken, die für ausgewählte Nutzer bestimmt sind — beispielsweise Nutzer, die eine Gebühr bezahlt haben —, indem Sie diese privaten Inhalte CloudFront mithilfe signierter URLs oder signierter Cookies bereitstellen. Weitere Informationen finden Sie unter Bereitstellen privater Inhalte mit signierten URLs und signierten Cookies.

Beschränken des Zugriffs auf Inhalte in Amazon S3-Buckets

Wenn Sie den Zugriff auf Ihre Inhalte einschränken, indem Sie beispielsweise CloudFront signierte URLs oder signierte Cookies verwenden, möchten Sie auch nicht, dass Personen Dateien ansehen, indem sie die direkte URL für die Datei verwenden. Stattdessen möchten Sie, dass der Zugriff auf die Dateien nur über die CloudFront -URL möglich ist, sodass Ihre Schutzmechanismen funktionieren.

Wenn Sie einen Amazon S3-Bucket als Ursprung für eine CloudFront Distribution verwenden, können Sie eine Origin Access Control (OAC) einrichten, die es ermöglicht, den Zugriff auf den S3-Bucket einzuschränken. Weitere Informationen finden Sie unter the section called “Beschränken des Zugriffs auf einen Amazon-S3-Ursprung”.

Beschränken des Zugriffs auf Inhalte von einem Application Load Balancer

Wenn Sie einen Application Load Balancer in Elastic Load Balancing als Ursprung verwenden CloudFront , können Sie die Konfiguration so konfigurierenCloudFront , dass Benutzer nicht direkt auf den Application Load Balancer zugreifen. Auf diese Weise können Benutzer nur über den Application Load Balancer zugreifen. So stellen Sie sicher CloudFront, dass Sie die Vorteile der Verwendung nutzen können. CloudFront Weitere Informationen finden Sie unter Beschränken des Zugriffs auf Application Load Balancer.

Verwenden Sie AWS WAF Web-ACLs

Sie können einen Firewall-Service für Webanwendungen verwenden AWS WAF, um eine Web Access Control List (Web ACL) zu erstellen, um den Zugriff auf Ihre Inhalte einzuschränken. Basierend auf von Ihnen angegebenen Bedingungen, wie z. B. den IP-

Adressen, von denen Anfragen stammen, oder den Werten von Abfragezeichenfolgen, CloudFront reagiert er auf Anfragen entweder mit dem angeforderten Inhalt oder mit einem HTTP 403-Statuscode (Verboten). Weitere Informationen finden Sie unter Verwenden Sie AWS WAF Schutzmaßnahmen.

Verwenden von geografische Einschränkungen

Mithilfe einer geografischen Einschränkung oder Geoblockierung können Sie verhindern, dass Benutzer aus bestimmten geografischen Regionen auf Inhalte zugreifen, die Sie über eine CloudFront-Verteilung bereitstellen. Beim Konfigurieren geografischer Einschränkungen können Sie aus verschiedenen Optionen wählen. Weitere Informationen finden Sie unter Einschränken der geografischen Distribution Ihrer Inhalte.

Identitäts- und Zugriffsmanagement für Amazon CloudFront

AWS Identity and Access Management (IAM) ist ein Programm AWS-Service , das einem Administrator hilft, den Zugriff auf Ressourcen sicher zu AWS kontrollieren. IAM-Administratoren kontrollieren, wer zur Nutzung von Ressourcen authentifiziert (angemeldet) und autorisiert werden kann (über Berechtigungen verfügt). CloudFront IAM ist ein Dienst AWS-Service , den Sie ohne zusätzliche Kosten nutzen können.

Themen

- Zielgruppe
- Authentifizierung mit Identitäten
- Verwalten des Zugriffs mit Richtlinien
- Wie Amazon mit CloudFront IAM arbeitet
- Identity-based Beispiele für Richtlinien für Amazon CloudFront
- AWS verwaltete Richtlinien für Amazon CloudFront
- Verwenden Sie serviceverknüpfte Rollen für CloudFront
- Beheben Sie Probleme mit der CloudFront Identität und dem Zugriff von Amazon

Zielgruppe

Die Art und Weise, wie Sie AWS Identity and Access Management (IAM) verwenden, unterscheidet sich je nach Ihrer Rolle:

- Servicebenutzer – Fordern Sie von Ihrem Administrator Berechtigungen an, wenn Sie nicht auf Features zugreifen können (siehe Beheben Sie Probleme mit der CloudFront Identität und dem Zugriff von Amazon).
- Serviceadministrator – Bestimmen Sie den Benutzerzugriff und stellen Sie Berechtigungsanfragen (siehe Wie Amazon mit CloudFront IAM arbeitet).
- IAM-Administrator – Schreiben Sie Richtlinien zur Zugriffsverwaltung (siehe Identity-based Beispiele für Richtlinien für Amazon CloudFront).

Authentifizierung mit Identitäten

Bei der Authentifizierung melden Sie sich AWS mit Ihren Identitätsdaten an. Sie müssen als IAM-Benutzer oder als IAM-Benutzer authentifiziert sein, indem Sie eine IAM-Rolle übernehmen. Root-Benutzer des AWS-Kontos

Sie können sich mit Anmeldeinformationen aus einer Identitätsquelle wie AWS IAM Identity Center (IAM Identity Center), Single Sign-On-Authentifizierung oder Anmeldeinformationen als Verbundidentität anmelden. Google/Facebook Weitere Informationen zum Anmelden finden Sie unter So melden Sie sich bei Ihrem AWS-Konto an im Benutzerhandbuch für AWS-Anmeldung .

AWS Stellt für den programmatischen Zugriff ein SDK und eine CLI zum kryptografischen Signieren von Anfragen bereit. Weitere Informationen finden Sie unter AWS Signature Version 4 for API requests im IAM-Benutzerhandbuch.

AWS-Konto Root-Benutzer

Wenn Sie einen erstellen AWS-Konto, beginnen Sie mit einer Anmeldeidentität, dem AWS-Konto Root-Benutzer, der vollständigen Zugriff auf alle AWS-Services Ressourcen hat. Wir raten ausdrücklich davon ab, den Root-Benutzer für Alltagsaufgaben zu verwenden. Eine Liste der Aufgaben, für die Sie sich als Root-Benutzer anmelden müssen, finden Sie unter Tasks that require root user credentials im IAM-Benutzerhandbuch.

Verbundidentität

Es hat sich bewährt, menschliche Benutzer dazu zu verpflichten, einen Verbund mit einem Identitätsanbieter zu verwenden, um AWS-Services mithilfe temporärer Anmeldeinformationen darauf zuzugreifen.

Bei einer Verbundidentität handelt es sich um einen Benutzer aus Ihrem Unternehmensverzeichnis, einem Web-Identitätsanbieter oder Directory Service einem Benutzer, der AWS-Services mithilfe

von Anmeldeinformationen aus einer Identitätsquelle zugreift. Verbundene Identitäten übernehmen Rollen, die temporäre Anmeldeinformationen bereitstellen.

Für die zentrale Zugriffsverwaltung empfehlen wir AWS IAM Identity Center. Weitere Informationen finden Sie unter Was ist IAM Identity Center? im AWS IAM Identity Center -Benutzerhandbuch.

IAM-Benutzer und -Gruppen

Ein IAM-Benutzer ist eine Identität mit bestimmten Berechtigungen für eine einzelne Person oder Anwendung. Wir empfehlen die Verwendung temporärer Anmeldeinformationen anstelle von IAM-Benutzern mit langfristigen Anmeldeinformationen. Weitere Informationen finden Sie im IAM-Benutzerhandbuch unter Erfordern, dass menschliche Benutzer einen Verbund mit einem Identitätsanbieter verwenden, um AWS mithilfe temporärer Anmeldeinformationen auf den Zugriff zuzugreifen.

Eine IAM-Gruppe spezifiziert eine Sammlung von IAM-Benutzern und erleichtert die Verwaltung von Berechtigungen für große Gruppen von Benutzern. Weitere Informationen finden Sie unter Anwendungsfälle für IAM-Benutzer im IAM-Benutzerhandbuch.

IAM-Rollen

Eine IAM-Rolle ist eine Identität mit spezifischen Berechtigungen, die temporäre Anmeldeinformationen bereitstellt. Sie können eine Rolle übernehmen, indem Sie von einer Benutzer- zu einer IAM-Rolle (Konsole) wechseln AWS CLI oder eine AWS API-Operation aufrufen. Weitere Informationen finden Sie unter Methoden, um eine Rolle zu übernehmen im IAM-Benutzerhandbuch.

IAM-Rollen sind nützlich für den Verbundbenutzer-Zugriff, temporäre IAM-Benutzerberechtigungen, kontoübergreifenden Zugriff, serviceübergreifenden Zugriff und Anwendungen, die auf Amazon EC2 laufen. Weitere Informationen finden Sie unter Kontoübergreifender Ressourcenzugriff in IAM im IAM-Benutzerhandbuch.

Verwalten des Zugriffs mit Richtlinien

Sie steuern den Zugriff, AWS indem Sie Richtlinien erstellen und sie an AWS Identitäten oder Ressourcen anhängen. Eine Richtlinie definiert Berechtigungen, wenn sie einer Identität oder Ressource zugeordnet sind. AWS wertet diese Richtlinien aus, wenn ein Principal eine Anfrage stellt. Die meisten Richtlinien werden AWS als JSON-Dokumente gespeichert. Weitere Informationen zu JSON-Richtliniendokumenten finden Sie unter Übersicht über JSON-Richtlinien im IAM-Benutzerhandbuch.

Mit Hilfe von Richtlinien legen Administratoren fest, wer Zugriff auf was hat, indem sie definieren, welches Prinzipal welche Aktionen auf welchen Ressourcenund unter welchen Bedingungendurchführen darf.

Standardmäßig haben Benutzer, Gruppen und Rollen keine Berechtigungen. Ein IAM-Administrator erstellt IAM-Richtlinien und fügt sie zu Rollen hinzu, die die Benutzer dann übernehmen können. IAM-Richtlinien definieren Berechtigungen unabhängig von der Methode, die zur Ausführung der Operation verwendet wird.

Identity-based Richtlinien

Identity-based Richtlinien sind Richtliniendokumente für JSON-Berechtigungen, die Sie an eine Identität (Benutzer, Gruppe oder Rolle) anhängen. Diese Richtlinien steuern, welche Aktionen Identitäten für welche Ressourcen und unter welchen Bedingungen ausführen können. Informationen zum Erstellen identitätsbasierter Richtlinien finden Sie unter Definieren benutzerdefinierter IAM-Berechtigungen mit vom Kunden verwalteten Richtlinien im IAM-Benutzerhandbuch.

Identity-based Richtlinien können integrierte Richtlinien (direkt in eine einzelne Identität eingebettet) oder verwaltete Richtlinien (eigenständige Richtlinien, die mehreren Identitäten zugeordnet sind) sein. Informationen dazu, wie Sie zwischen verwalteten und Inline-Richtlinien wählen, finden Sie unter Choose between managed policies and inline policies im IAM-Benutzerhandbuch.

Resource-based Richtlinien

Resource-based Richtlinien sind JSON-Richtliniendokumente, die Sie an eine Ressource anhängen. Beispiele hierfür sind Vertrauensrichtlinien für IAM-Rollen und Amazon S3-Bucket-Richtlinien. In Services, die ressourcenbasierte Richtlinien unterstützen, können Service-Administratoren sie verwenden, um den Zugriff auf eine bestimmte Ressource zu steuern. Sie müssen in einer ressourcenbasierten Richtlinie einen Prinzipal angeben.

Resource-based Richtlinien sind Inline-Richtlinien, die sich in diesem Dienst befinden. Sie können AWS verwaltete Richtlinien von IAM nicht in einer ressourcenbasierten Richtlinie verwenden.

Weitere Richtlinientypen

AWS unterstützt zusätzliche Richtlinientypen, mit denen die maximale Anzahl von Berechtigungen festgelegt werden kann, die von gängigeren Richtlinientypen gewährt werden:

- Berechtigungsgrenzen – Eine Berechtigungsgrenze legt die maximalen Berechtigungen fest, die eine identitätsbasierte Richtlinie einer IAM-Entität erteilen kann. Weitere Informationen finden Sie unter Berechtigungsgrenzen für IAM-Entitäten im -IAM-Benutzerhandbuch.

- Service-Kontrollrichtlinien (SCPs) – SCPs legen die maximalen Berechtigungen für eine Organisation oder Organisationseinheit in AWS Organizations fest. Weitere Informationen finden Sie unter Service-Kontrollrichtlinien im AWS Organizations -Benutzerhandbuch.
- Ressourcen-Kontrollrichtlinien (RCPs) – RCPs definieren die maximale Anzahl an Berechtigungen, die Ressourcen in Ihren Konten zur Verfügung stehen. Weitere Informationen finden Sie unter Ressourcen-Kontrollrichtlinien im AWS Organizations -Benutzerhandbuch.
- Sitzungsrichtlinien – Sitzungsrichtlinien sind erweiterte Richtlinien, die als Parameter übergeben werden, wenn Sie eine temporäre Sitzung für eine Rolle oder einen Verbundbenutzer erstellen. Weitere Informationen finden Sie unter Sitzungsrichtlinien im IAM-Benutzerhandbuch.

Mehrere Richtlinientypen

Wenn mehrere Arten von Richtlinien für eine Anfrage gelten, sind die daraus resultierenden Berechtigungen komplizierter zu verstehen. Informationen darüber, wie AWS bestimmt wird, ob eine Anfrage zulässig ist, wenn mehrere Richtlinientypen betroffen sind, finden Sie im IAM-Benutzerhandbuch unter Logik zur Richtlinienauswertung.

Wie Amazon mit CloudFront IAM arbeitet

Bevor Sie IAM verwenden, um den Zugriff auf zu verwalten CloudFront, sollten Sie sich mit den IAM-Funktionen vertraut machen. CloudFront

IAM-Funktionen, die Sie mit Amazon verwenden können CloudFront

IAM-Feature	CloudFront Unterstützung
Identity-based Richtlinien	Ja
Resource-based Richtlinien	Nein
Richtlinienaktionen	Ja
Richtlinienressourcen	Ja
Richtlinienbedingungsschlüssel (services pezifisch)	Ja
ACLs	Nein

IAM-Feature	CloudFront Unterstützung
ABAC (Tags in Richtlinien)	Teilweise
Temporäre Anmeldeinformationen	Ja
Forward Access Sessions (FAS)	Nein
Servicerollen	Nein
Service-linked Rollen	Ja

Einen allgemeinen Überblick darüber, wie CloudFront und andere AWS Dienste mit den meisten IAM-Funktionen funktionieren, finden Sie im IAM-Benutzerhandbuch unter AWS Dienste, die mit IAM funktionieren.

Identity-based Richtlinien für CloudFront

Unterstützt Richtlinien auf Identitätsbasis: Ja

Identity-based Richtlinien sind Richtliniendokumente für JSON-Berechtigungen, die Sie an eine Identität anhängen können, z. B. an einen IAM-Benutzer, eine Benutzergruppe oder eine Rolle. Diese Richtlinien steuern, welche Aktionen die Benutzer und Rollen für welche Ressourcen und unter welchen Bedingungen ausführen können. Informationen zum Erstellen identitätsbasierter Richtlinien finden Sie unter Definieren benutzerdefinierter IAM-Berechtigungen mit vom Kunden verwalteten Richtlinien im IAM-Benutzerhandbuch.

Mit identitätsbasierten IAM-Richtlinien können Sie angeben, welche Aktionen und Ressourcen zugelassen oder abgelehnt werden. Darüber hinaus können Sie die Bedingungen festlegen, unter denen Aktionen zugelassen oder abgelehnt werden. Informationen zu sämtlichen Elementen, die Sie in einer JSON-Richtlinie verwenden, finden Sie in der IAM-Referenz für JSON-Richtlinienelemente im IAM-Benutzerhandbuch.

Identity-based Richtlinienbeispiele für CloudFront

Beispiele für CloudFront identitätsbasierte Richtlinien finden Sie unter. Identity-based Beispiele für Richtlinien für Amazon CloudFront

Resource-based Richtlinien innerhalb CloudFront

Unterstützt ressourcenbasierte Richtlinien: Nein

Resource-based Richtlinien sind JSON-Richtliniendokumente, die Sie an eine Ressource anhängen. Beispiele für ressourcenbasierte Richtlinien sind IAM-Rollen-Vertrauensrichtlinien und Amazon-S3-Bucket-Richtlinien. In Services, die ressourcenbasierte Richtlinien unterstützen, können Service-Administratoren sie verwenden, um den Zugriff auf eine bestimmte Ressource zu steuern. Für die Ressource, an welche die Richtlinie angehängt ist, legt die Richtlinie fest, welche Aktionen ein bestimmter Prinzipal unter welchen Bedingungen für diese Ressource ausführen kann. Sie müssen in einer ressourcenbasierten Richtlinie einen Prinzipal angeben. Zu den Principals können Konten, Benutzer, Rollen, Verbundbenutzer oder gehören. AWS-Services

Um kontoübergreifenden Zugriff zu ermöglichen, können Sie ein gesamtes Konto oder IAM-Entitäten in einem anderen Konto als Prinzipal in einer ressourcenbasierten Richtlinie angeben. Weitere Informationen finden Sie unter Kontoübergreifender Ressourcenzugriff in IAM im IAM-Benutzerhandbuch.

Politische Maßnahmen für CloudFront

Unterstützt Richtlinienaktionen: Ja

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer Zugriff auf was hat. Das heißt, welcher Prinzipal Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen kann.

Das Element `Action` einer JSON-Richtlinie beschreibt die Aktionen, mit denen Sie den Zugriff in einer Richtlinie zulassen oder verweigern können. Nehmen Sie Aktionen in eine Richtlinie auf, um Berechtigungen zur Ausführung des zugehörigen Vorgangs zu gewähren.

Eine Liste der CloudFront Aktionen finden Sie unter Von Amazon definierte Aktionen CloudFront in der Referenz zur Serviceautorisierung.

Bei den CloudFront verwendeten Richtlinienaktionen wird vor der Aktion das folgende Präfix verwendet:

```
cloudfront
```

Um mehrere Aktionen in einer einzigen Anweisung anzugeben, trennen Sie sie mit Kommata:

```
"Action": [
      "cloudfront:action1",
      "cloudfront:action2"
         ]
```

Beispiele für CloudFront identitätsbasierte Richtlinien finden Sie unter. Identity-based Beispiele für Richtlinien für Amazon CloudFront

Politische Ressourcen für CloudFront

Unterstützt Richtlinienressourcen: Ja

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer Zugriff auf was hat. Das heißt, welcher Prinzipal Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen kann.

Das JSON-Richtlinienelement `Resource` gibt die Objekte an, auf welche die Aktion angewendet wird. Als Best Practice geben Sie eine Ressource mit dem zugehörigen Amazon-Ressourcennamen (ARN) an. Verwenden Sie für Aktionen, die keine Berechtigungen auf Ressourcenebene unterstützen, einen Platzhalter (*), um anzugeben, dass die Anweisung für alle Ressourcen gilt.

```
"Resource": "*"
```

Eine Liste der CloudFront Ressourcentypen und ihrer ARNs finden Sie unter Von Amazon definierte Ressourcen CloudFront in der Service Authorization Reference. Informationen dazu, mit welchen Aktionen Sie den ARN jeder Ressource angeben können, finden Sie unter Von Amazon CloudFront definierte Aktionen.

Beispiele für CloudFront identitätsbasierte Richtlinien finden Sie unter. Identity-based Beispiele für Richtlinien für Amazon CloudFront

Richtlinien-Bedingungsschlüssel für CloudFront

Unterstützt servicespezifische Richtlinienbedingungsschlüssel: Ja

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer Zugriff auf was hat. Das heißt, welcher Prinzipal Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen kann.

Das Element `Condition` gibt an, wann Anweisungen auf der Grundlage definierter Kriterien ausgeführt werden. Sie können bedingte Ausdrücke erstellen, die Bedingungsoperatoren verwenden, z. B. ist gleich oder kleiner als, damit die Bedingung in der Richtlinie mit Werten in der Anforderung übereinstimmt. Alle AWS globalen Bedingungsschlüssel finden Sie unter AWS Globale Bedingungskontextschlüssel im IAM-Benutzerhandbuch.

Eine Liste der CloudFront Bedingungsschlüssel finden Sie unter Bedingungsschlüssel für Amazon CloudFront in der Serviceautorisierungsreferenz. Informationen dazu, mit welchen Aktionen und Ressourcen Sie einen Bedingungsschlüssel verwenden können, finden Sie unter Von Amazon definierte Aktionen CloudFront.

Beispiele für CloudFront identitätsbasierte Richtlinien finden Sie unter. Identity-based Beispiele für Richtlinien für Amazon CloudFront

ACLs in CloudFront

Unterstützt ACLs: Nein

Zugriffssteuerungslisten (ACLs) steuern, welche Prinzipale (Kontomitglieder, Benutzer oder Rollen) auf eine Ressource zugreifen können. ACLs sind ähnlich wie ressourcenbasierte Richtlinien, verwenden jedoch nicht das JSON-Richtliniendokumentformat.

ABAC mit CloudFront

Unterstützt ABAC (Tags in Richtlinien): Teilweise

Attribute-based Access Control (ABAC) ist eine Autorisierungsstrategie, die Berechtigungen auf der Grundlage von Attributen definiert, die als Tags bezeichnet werden. Sie können Tags an IAM-Entitäten und AWS -Ressourcen anhängen und dann ABAC-Richtlinien so entwerfen, dass Operationen möglich sind, wenn das Tag des Principals mit dem Tag auf der Ressource übereinstimmt.

Um den Zugriff auf der Grundlage von Tags zu steuern, geben Sie im Bedingungselement einer Richtlinie Tag-Informationen an, indem Sie die Schlüssel `aws:ResourceTag/`*`key-name`*, `aws:RequestTag/`*`key-name`*, oder Bedingung `aws:TagKeys` verwenden.

Wenn ein Service alle drei Bedingungsschlüssel für jeden Ressourcentyp unterstützt, lautet der Wert für den Service Ja. Wenn ein Service alle drei Bedingungsschlüssel für nur einige Ressourcentypen unterstützt, lautet der Wert Teilweise.

Weitere Informationen zu ABAC finden Sie unter Definieren von Berechtigungen mit ABAC-Autorisierung im IAM-Benutzerhandbuch. Um ein Tutorial mit Schritten zur Einstellung von ABAC anzuzeigen, siehe Attributbasierte Zugriffskontrolle (ABAC) verwenden im IAM-Benutzerhandbuch.

CloudFront unterstützt ABAC für CloudFront Ressourcen, die Tags unterstützen, wie Distributionen, Mandanten oder Trust Stores.

Verwenden Sie temporäre Anmeldeinformationen mit CloudFront

Unterstützt temporäre Anmeldeinformationen: Ja

Temporäre Anmeldeinformationen ermöglichen den kurzfristigen Zugriff auf AWS Ressourcen und werden automatisch erstellt, wenn Sie den Verbund verwenden oder die Rollen wechseln. AWS empfiehlt, temporäre Anmeldeinformationen dynamisch zu generieren, anstatt langfristige Zugriffsschlüssel zu verwenden. Weitere Informationen finden Sie unter Temporäre Anmeldeinformationen in IAM und AWS-Services , die mit IAM funktionieren im IAM-Benutzerhandbuch.

Zugriffssitzungen weiterleiten für CloudFront

Unterstützt Forward Access Sessions (FAS): Nein

Bei Forward Access-Sitzungen (FAS) werden die Berechtigungen des Prinzipals verwendet, der einen aufruft AWS-Service, in Kombination mit der Anforderung, Anfragen AWS-Service an nachgelagerte Dienste zu stellen. Einzelheiten zu den Richtlinien für FAS-Anforderungen finden Sie unter Zugriffssitzungen weiterleiten.

Servicerollen für CloudFront

Unterstützt Servicerollen: Nein

Eine Servicerolle ist eine IAM-Rolle, die ein Service annimmt, um Aktionen in Ihrem Namen auszuführen. Ein IAM-Administrator kann eine Servicerolle innerhalb von IAM erstellen, ändern und löschen. Weitere Informationen finden Sie unter Erstellen einer Rolle zum Delegieren von Berechtigungen an einen AWS-Service im IAM-Benutzerhandbuch.

> **Warning**
>
> Durch das Ändern der Berechtigungen für eine Servicerolle kann die CloudFront Funktionalität beeinträchtigt werden. Bearbeiten Sie Servicerollen nur CloudFront , wenn Sie dazu aufgefordert werden.

Service-linked Rollen für CloudFront

Unterstützt serviceverknüpfte Rollen: Ja

Eine dienstgebundene Rolle ist eine Art von Servicerolle, die mit einer AWS-Service verknüpft ist. Der Dienst kann die Rolle übernehmen, eine Aktion in Ihrem Namen auszuführen. Service-linked Rollen werden in Ihrem angezeigt AWS-Konto und gehören dem Dienst. Ein IAM-Administrator kann die Berechtigungen für Service-verknüpfte Rollen anzeigen, aber nicht bearbeiten.

CloudFront verwendet dienstverknüpfte Rollen, um Aktionen für Sie auszuführen. Weitere Informationen zum Erstellen oder Verwalten von CloudFront dienstverknüpften Rollen finden Sie unter. Verwenden Sie serviceverknüpfte Rollen für CloudFront Weitere Informationen zum Erstellen oder Verwalten von serviceverknüpften Lambda@Edge-Rollen finden Sie unter Service-linked Rollen für Lambda @Edge.

Details zum Erstellen oder Verwalten von serviceverknüpften Rollen finden Sie unter AWS -Services, die mit IAM funktionieren. Suchen Sie einen Dienst in der Tabelle, die eine `Yes` in der Service-linked Rollenspalte enthält. Wählen Sie den Link Yes (Ja) aus, um die Dokumentation für die serviceverknüpfte Rolle für diesen Service anzuzeigen.

Identity-based Beispiele für Richtlinien für Amazon CloudFront

Benutzer und Rollen haben standardmäßig nicht die Berechtigung, CloudFront-Ressourcen zu erstellen oder zu ändern. Ein IAM-Administrator muss IAM-Richtlinien erstellen, die Benutzern die Berechtigung erteilen, Aktionen für die Ressourcen auszuführen, die sie benötigen.

Informationen dazu, wie Sie unter Verwendung dieser beispielhaften JSON-Richtliniendokumente eine identitätsbasierte IAM-Richtlinie erstellen, finden Sie unter Erstellen von IAM-Richtlinien (Konsole) im IAM-Benutzerhandbuch.

Einzelheiten zu Aktionen und Ressourcentypen, die von definiert werden CloudFront, einschließlich des Formats der ARNs für jeden der Ressourcentypen, finden Sie unter Aktionen, Ressourcen und Bedingungsschlüssel für Amazon CloudFront in der Serviceautorisierungsreferenz.

Themen

- Best Practices für Richtlinien
- Gewähren der Berechtigung zur Anzeige der eigenen Berechtigungen für Benutzer
- Berechtigungen für den programmgesteuerten Zugriff CloudFront
- Für die Verwendung der CloudFront Konsole sind Berechtigungen erforderlich
- Beispiele für vom Kunden verwaltete Richtlinien

Best Practices für Richtlinien

Identity-based Richtlinien legen fest, ob jemand CloudFront Ressourcen in Ihrem Konto erstellen, darauf zugreifen oder löschen kann. Dies kann zusätzliche Kosten für Ihr verursachen AWS-Konto. Wenn du identitätsbasierte Richtlinien erstellst oder bearbeitest, befolge diese Richtlinien und Empfehlungen:

- Beginnen Sie mit AWS verwalteten Richtlinien und wechseln Sie zu Berechtigungen mit den geringsten Rechten — Verwenden Sie die AWS verwalteten Richtlinien, die Berechtigungen für viele gängige Anwendungsfälle gewähren, um Ihren Benutzern und Workloads zunächst Berechtigungen zu gewähren. Sie sind in Ihrem verfügbar. AWS-Konto Wir empfehlen Ihnen, die Berechtigungen weiter zu reduzieren, indem Sie vom AWS Kunden verwaltete Richtlinien definieren, die speziell auf Ihre Anwendungsfälle zugeschnitten sind. Weitere Informationen finden Sie unter Von AWS verwaltete Richtlinien oder Von AWS verwaltete Richtlinien für Auftragsfunktionen im IAM-Benutzerhandbuch.
- Anwendung von Berechtigungen mit den geringsten Rechten – Wenn Sie mit IAM-Richtlinien Berechtigungen festlegen, gewähren Sie nur die Berechtigungen, die für die Durchführung einer Aufgabe erforderlich sind. Sie tun dies, indem Sie die Aktionen definieren, die für bestimmte Ressourcen unter bestimmten Bedingungen durchgeführt werden können, auch bekannt als die geringsten Berechtigungen. Weitere Informationen zur Verwendung von IAM zum Anwenden von Berechtigungen finden Sie unter Richtlinien und Berechtigungen in IAM im IAM-Benutzerhandbuch.
- Verwenden von Bedingungen in IAM-Richtlinien zur weiteren Einschränkung des Zugriffs – Sie können Ihren Richtlinien eine Bedingung hinzufügen, um den Zugriff auf Aktionen und Ressourcen zu beschränken. Sie können beispielsweise eine Richtlinienbedingung schreiben, um festzulegen, dass alle Anforderungen mithilfe von SSL gesendet werden müssen. Sie können auch Bedingungen verwenden, um Zugriff auf Serviceaktionen zu gewähren, wenn diese im Rahmen einer bestimmten Aktion verwendet werden AWS-Service, z. CloudFormation

B. Weitere Informationen finden Sie unter IAM-JSON-Richtlinienelemente: Bedingung im IAM-Benutzerhandbuch.

- Verwenden von IAM Access Analyzer zur Validierung Ihrer IAM-Richtlinien, um sichere und funktionale Berechtigungen zu gewährleisten – IAM Access Analyzer validiert neue und vorhandene Richtlinien, damit die Richtlinien der IAM-Richtliniensprache (JSON) und den bewährten IAM-Methoden entsprechen. IAM Access Analyzer stellt mehr als 100 Richtlinienprüfungen und umsetzbare Empfehlungen zur Verfügung, damit Sie sichere und funktionale Richtlinien erstellen können. Weitere Informationen finden Sie unter Richtlinienvalidierung mit IAM Access Analyzer im IAM-Benutzerhandbuch.
- Mehrstufige Authentifizierung (MFA) erforderlich — Wenn Sie ein Szenario haben, in dem IAM-Benutzer oder ein Root-Benutzer in Ihrem System erforderlich sind AWS-Konto, aktivieren Sie MFA für zusätzliche Sicherheit. Um MFA beim Aufrufen von API-Vorgängen anzufordern, fügen Sie Ihren Richtlinien MFA-Bedingungen hinzu. Weitere Informationen finden Sie unter Sicherer API-Zugriff mit MFA im IAM-Benutzerhandbuch.

Weitere Informationen zu bewährten Methoden in IAM finden Sie unter Best Practices für die Sicherheit in IAM im IAM-Benutzerhandbuch.

Gewähren der Berechtigung zur Anzeige der eigenen Berechtigungen für Benutzer

In diesem Beispiel wird gezeigt, wie Sie eine Richtlinie erstellen, die IAM-Benutzern die Berechtigung zum Anzeigen der eingebundenen Richtlinien und verwalteten Richtlinien gewährt, die ihrer Benutzeridentität angefügt sind. Diese Richtlinie beinhaltet Berechtigungen zum Ausführen dieser Aktion auf der Konsole oder programmgesteuert mithilfe der OR-API. AWS CLI AWS

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
```

```
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

Berechtigungen für den programmgesteuerten Zugriff CloudFront

Das folgende Beispiel zeigt eine Berechtigungsrichtlinie. Der Abschnitt `Sid` (die Anweisungs-ID) ist optional.

JSON

```
{
   "Version":"2012-10-17",
   "Statement": [
      {
         "Sid": "AllowAllCloudFrontPermissions",
         "Effect": "Allow",
         "Action": ["cloudfront:*"],
         "Resource": "*"
      }
   ]
}
```

Die Richtlinie gewährt Berechtigungen zum Ausführen aller CloudFront Vorgänge, was für den programmgesteuerten Zugriff CloudFront ausreicht. Wenn Sie die Konsole für den Zugriff verwenden

CloudFront, finden Sie weitere Informationen unter. Für die Verwendung der CloudFront Konsole sind Berechtigungen erforderlich

Eine Liste der Aktionen und den ARN, den Sie angeben, um die Berechtigung zur Verwendung der einzelnen Aktionen zu gewähren oder zu verweigern, finden Sie unter Aktionen, Ressourcen und Bedingungsschlüssel für Amazon CloudFront in der Serviceautorisierungsreferenz.

Für die Verwendung der CloudFront Konsole sind Berechtigungen erforderlich

Um vollen Zugriff auf die CloudFront Konsole zu gewähren, gewähren Sie die Berechtigungen gemäß der folgenden Berechtigungsrichtlinie:

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "acm:ListCertificates",
            "cloudfront:*",
            "cloudwatch:DescribeAlarms",
            "cloudwatch:PutMetricAlarm",
            "cloudwatch:GetMetricStatistics",
            "elasticloadbalancing:DescribeLoadBalancers",
            "iam:ListServerCertificates",
            "sns:ListSubscriptionsByTopic",
            "sns:ListTopics",
            "waf:GetWebACL",
            "waf:ListWebACLs"
         ],
         "Resource":"*"
      },
      {
         "Effect":"Allow",
         "Action":[
            "s3:ListAllMyBuckets",
            "s3:PutBucketPolicy"
         ],
         "Resource":"arn:aws:s3:::*"
      }
```

```
    ]
}
```

Gründe, warum die Berechtigungen erforderlich sind

`acm:ListCertificates`

Wenn Sie Distributionen mithilfe der CloudFront Konsole erstellen und aktualisieren und so konfigurieren CloudFront möchten, dass HTTPS zwischen dem Viewer und CloudFront oder zwischen CloudFront und dem Ursprung erforderlich ist, können Sie eine Liste der ACM-Zertifikate anzeigen.

Diese Berechtigung ist nicht erforderlich, wenn Sie die CloudFront Konsole nicht verwenden.

`cloudfront:*`

Sie können alle CloudFront Aktionen ausführen.

`cloudwatch:DescribeAlarms` und **`cloudwatch:PutMetricAlarm`**

Ermöglicht das Erstellen und Anzeigen von CloudWatch Alarmen in der CloudFront Konsole. Weitere Informationen finden Sie auch unter `sns:ListSubscriptionsByTopic` und `sns:ListTopics`.

Diese Berechtigungen sind nicht erforderlich, wenn Sie die CloudFront-Konsole nicht verwenden.

`cloudwatch:GetMetricStatistics`

Lassen Sie uns CloudWatch Metriken in der CloudFront Konsole CloudFront rendern.

Diese Berechtigung ist nicht erforderlich, wenn Sie die CloudFront Konsole nicht verwenden.

`elasticloadbalancing:DescribeLoadBalancers`

Wenn Sie Distributionen erstellen und aktualisieren, können Sie eine Liste der Elastic Load Balancing-Load Balancer in der Liste der verfügbaren Ursprünge einsehen.

Diese Berechtigung ist nicht erforderlich, wenn Sie die CloudFront Konsole nicht verwenden.

`iam:ListServerCertificates`

Wenn Sie Distributionen mithilfe der CloudFront Konsole erstellen und aktualisieren und so konfigurieren CloudFront möchten, dass HTTPS zwischen dem Viewer und CloudFront oder

zwischen CloudFront und dem Ursprung erforderlich ist, können Sie eine Liste der Zertifikate im IAM-Zertifikatsspeicher anzeigen.

Diese Berechtigung ist nicht erforderlich, wenn Sie die CloudFront Konsole nicht verwenden.

`s3:ListAllMyBuckets`

Wenn Sie Distributionen erstellen und aktualisieren, können Sie die folgenden Vorgänge ausführen:

- Anzeigen einer Liste von S3-Buckets in der Liste der verfügbaren Ursprünge
- Anzeigen einer Liste von S3-Buckets, in denen Sie die Zugriffsprotokolle speichern können

Diese Berechtigung ist nicht erforderlich, wenn Sie die CloudFront Konsole nicht verwenden.

`S3:PutBucketPolicy`

Wenn Sie Verteilungen erstellen oder aktualisieren, die den Zugriff auf S3-Buckets beschränken, kann der Benutzer die Bucket-Richtlinie so aktualisieren, mit Zugriff auf die CloudFront-Ursprungszugriffsidentität gewährt wird. Weitere Informationen finden Sie unter the section called “Verwenden einer Ursprungszugriffsidentität (Legacy, nicht empfohlen)”.

Diese Berechtigung ist nicht erforderlich, wenn Sie die CloudFront Konsole nicht verwenden.

`sns:ListSubscriptionsByTopic` und **`sns:ListTopics`**

Wenn Sie CloudWatch Alarme in der CloudFront Konsole erstellen, können Sie ein SNS-Thema für Benachrichtigungen auswählen.

Diese Berechtigungen sind nicht erforderlich, wenn Sie die CloudFront-Konsole nicht verwenden.

`waf:GetWebACL` und **`waf:ListWebACLs`**

Sie können eine Liste der AWS WAF Web-ACLs in der CloudFront Konsole anzeigen.

Diese Berechtigungen sind nicht erforderlich, wenn Sie die CloudFront-Konsole nicht verwenden.

Permission-only Aktionen für die Konsole CloudFront

Auf der Seite „CloudFront Security Savings Bundle“ können Sie die folgenden CloudFront Aktionen ausführen. Die folgenden API-Aktionen sollen nicht von Ihrem Code aufgerufen werden und sind nicht in den AWS SDKs AWS CLI und enthalten.

Action	Description
`CreateSavingsPlan`	Gewährt die Berechtigung zum Erstellen eines neuen Savings Plan.
`GetSavingsPlan`	Gewährt die Berechtigung zum Abrufen eines Savings Plan.
`ListRateCards`	Erteilt die Erlaubnis, CloudFront Preiskarten für das Konto aufzulisten.
`ListSavingsPlans`	Gewährt die Berechtigung zum Auflisten von Savings Plans im Konto.
`ListUsages`	Erteilt die Erlaubnis, die CloudFront Nutzung von Listen aufzulisten.
`UpdateSavingsPlan`	Gewährt die Berechtigung zum Aktualisieren eines Savings Plan.

Hinweise

- Weitere Informationen zu CloudFront Sparplänen finden Sie im Abschnitt CloudFront Sicherheitssparpaket der CloudFront häufig gestellten Fragen zu Amazon.
- Wenn Sie einen Sparplan für erstellen CloudFront und ihn später löschen möchten, wenden Sie sich an AWS Support.

Beispiele für vom Kunden verwaltete Richtlinien

Sie können Ihre eigenen benutzerdefinierten IAM-Richtlinien erstellen, um Berechtigungen für CloudFront API-Aktionen zuzulassen. Die benutzerdefinierten Richtlinien können Sie dann den IAM-Benutzern oder -Gruppen zuweisen, welche die angegebenen Berechtigungen benötigen. Diese Richtlinien funktionieren, wenn Sie die CloudFront API, die AWS SDKs oder die verwenden. AWS CLI Die folgenden Beispiele zeigen Berechtigungen für einige häufige Anwendungsfälle. Die Richtlinie, auf die einem Benutzer vollen Zugriff gewährt wird CloudFront, finden Sie unterFür die Verwendung der CloudFront Konsole sind Berechtigungen erforderlich.

Beispiele

- Beispiel 1: Lesezugriff auf alle Verteilungen gewähren
- Beispiel 2: Erstellen, Aktualisieren und Löschen von Distributionen erlauben
- Beispiel 3: Erstellen und Auflisten von Invalidierungen erlauben
- Beispiel 4: Erlauben der Erstellung einer Distribution

Beispiel 1: Lesezugriff auf alle Verteilungen gewähren

Die folgende Berechtigungsrichtlinie gewährt dem Benutzer die Berechtigung, alle Distributionen in der CloudFront Konsole anzuzeigen:

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "acm:ListCertificates",
            "cloudfront:GetDistribution",
            "cloudfront:GetDistributionConfig",
            "cloudfront:ListDistributions",
            "cloudfront:ListCloudFrontOriginAccessIdentities",
            "elasticloadbalancing:DescribeLoadBalancers",
            "iam:ListServerCertificates",
            "sns:ListSubscriptionsByTopic",
            "sns:ListTopics",
            "waf:GetWebACL",
            "waf:ListWebACLs"
         ],
         "Resource":"*"
      },
      {
         "Effect":"Allow",
         "Action":[
            "s3:ListAllMyBuckets"
         ],
         "Resource":"arn:aws:s3:::*"
      }
```

```
   ]
}
```

Beispiel 2: Erstellen, Aktualisieren und Löschen von Distributionen erlauben

Mit der folgenden Berechtigungsrichtlinie können Sie mithilfe der Konsole Distributionen erstellen, aktualisieren und löschen: CloudFront

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "acm:ListCertificates",
            "cloudfront:CreateDistribution",
            "cloudfront:DeleteDistribution",
            "cloudfront:GetDistribution",
            "cloudfront:GetDistributionConfig",
            "cloudfront:ListDistributions",
            "cloudfront:UpdateDistribution",
            "cloudfront:ListCloudFrontOriginAccessIdentities",
            "elasticloadbalancing:DescribeLoadBalancers",
            "iam:ListServerCertificates",
            "sns:ListSubscriptionsByTopic",
            "sns:ListTopics",
            "waf:GetWebACL",
            "waf:ListWebACLs"
         ],
         "Resource":"*"
      },
      {
         "Effect":"Allow",
         "Action":[
            "s3:ListAllMyBuckets",
            "s3:PutBucketPolicy"
         ],
         "Resource":"arn:aws:s3:::*"
      }
   ]
```

```
}
```

Die `cloudfront:ListCloudFrontOriginAccessIdentities`-Berechtigung erlaubt es Benutzern, automatisch einer vorhandenen Ursprungszugriffsidentität die Berechtigung für den Zugriff auf Objekte in einem Amazon S3-Bucket zu erteilen. Wenn Sie außerdem möchten, dass Benutzer Ursprungszugriffsidentitäten erstellen können, müssen Sie auch die `cloudfront:CreateCloudFrontOriginAccessIdentity`-Berechtigung erteilen.

Beispiel 3: Erstellen und Auflisten von Invalidierungen erlauben

Mit der folgenden Berechtigungsrichtlinie können Sie Ungültigerklärungen erstellen und auflisten. Sie beinhaltet den Lesezugriff auf CloudFront Distributionen, da Sie Ungültigerklärungen erstellen und anzeigen, indem Sie zunächst die Einstellungen für eine Verteilung anzeigen:

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "acm:ListCertificates",
            "cloudfront:GetDistribution",
            "cloudfront:GetStreamingDistribution",
            "cloudfront:GetDistributionConfig",
            "cloudfront:ListDistributions",
            "cloudfront:ListCloudFrontOriginAccessIdentities",
            "cloudfront:CreateInvalidation",
            "cloudfront:GetInvalidation",
            "cloudfront:ListInvalidations",
            "elasticloadbalancing:DescribeLoadBalancers",
            "iam:ListServerCertificates",
            "sns:ListSubscriptionsByTopic",
            "sns:ListTopics",
            "waf:GetWebACL",
            "waf:ListWebACLs"
         ],
         "Resource":"*"
      },
      {
```

```
         "Effect":"Allow",
         "Action":[
            "s3:ListAllMyBuckets"
         ],
         "Resource":"arn:aws:s3:::*"
      }
   ]
}
```

Beispiel 4: Erlauben der Erstellung einer Distribution

Die folgende Berechtigungsrichtlinie gewährt dem Benutzer die Berechtigung, Distributionen in der Konsole zu erstellen und aufzulisten. CloudFront Geben Sie für die Aktion `CreateDistribution` das Platzhalterzeichen (*) für `Resource` und nicht für den Distribution-ARN (`arn:aws:cloudfront::123456789012:distribution/*`) an. Weitere Informationen zum `Resource`-Element finden Sie unter IAM-JSON-Richtlinienelemente: Ressource im IAM-Benutzerhandbuch.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": "cloudfront:CreateDistribution",
            "Resource": "*"
        },
        {
            "Sid": "VisualEditor1",
            "Effect": "Allow",
            "Action": "cloudfront:ListDistributions",
            "Resource": "*"
        }
    ]
}
```

AWS verwaltete Richtlinien für Amazon CloudFront

Um Benutzern, Gruppen und Rollen Berechtigungen hinzuzufügen, ist es einfacher, AWS verwaltete Richtlinien zu verwenden, als Richtlinien selbst zu schreiben. Es erfordert Zeit und Fachwissen, um von Kunden verwaltete IAM-Richtlinien zu erstellen, die Ihren Benutzern nur die benötigten Berechtigungen bieten. Um schnell loszulegen, können Sie unsere AWS verwalteten Richtlinien verwenden. Diese Richtlinien decken allgemeine Anwendungsfälle ab und sind in Ihrem AWS-Konto verfügbar. Weitere Informationen zu AWS verwalteten Richtlinien finden Sie im IAM-Benutzerhandbuch unter AWS Verwaltete Richtlinien.

AWS Dienste verwalten und aktualisieren AWS verwaltete Richtlinien. Sie können die Berechtigungen in AWS verwalteten Richtlinien nicht ändern. Services fügen einer von AWS verwalteten Richtlinien gelegentlich zusätzliche Berechtigungen hinzu, um neue Features zu unterstützen. Diese Art von Update betrifft alle Identitäten (Benutzer, Gruppen und Rollen), an welche die Richtlinie angehängt ist. Services aktualisieren eine von AWS verwaltete Richtlinie am ehesten, wenn eine neue Funktion gestartet wird oder neue Berechtigungen verfügbar werden. Dienste entfernen keine Berechtigungen aus einer AWS verwalteten Richtlinie, sodass durch Richtlinienaktualisierungen Ihre bestehenden Berechtigungen nicht beeinträchtigt werden.

AWS Unterstützt außerdem verwaltete Richtlinien für Jobfunktionen, die sich über mehrere Dienste erstrecken. Die ReadOnlyAccess AWS verwaltete Richtlinie bietet beispielsweise schreibgeschützten Zugriff auf alle AWS Dienste und Ressourcen. Wenn ein Dienst eine neue Funktion startet, werden nur Leseberechtigungen für neue Operationen und Ressourcen AWS hinzugefügt. Eine Liste und Beschreibungen der Richtlinien für Auftragsfunktionen finden Sie in Verwaltete AWS -Richtlinien für Auftragsfunktionen im IAM-Leitfaden.

Themen

- AWS verwaltete Richtlinie: CloudFrontReadOnlyAccess
- AWS verwaltete Richtlinie: CloudFrontFullAccess
- AWS verwaltete Richtlinie: AWSCloudFrontLogger
- AWS verwaltete Richtlinie: AWSLambdaReplicator
- AWS verwaltete Richtlinie: AWSCloudFrontVPCOriginServiceRolePolicy
- CloudFront Aktualisierungen für AWS Verwaltete Richtlinien

AWS verwaltete Richtlinie: CloudFrontReadOnlyAccess

Sie können die CloudFrontReadOnlyAccess-Richtlinie an Ihre IAM-Identitäten anfügen. Diese Richtlinie ermöglicht nur Leseberechtigungen für Ressourcen. CloudFront Sie ermöglicht auch Nur-Leseberechtigungen für andere AWS Dienstressourcen, die sich auf die Konsole beziehen CloudFront und in der Konsole sichtbar sind. CloudFront

Details zu Berechtigungen

Diese Richtlinie umfasst die folgenden Berechtigungen.

- `cloudfront:Describe*`— Ermöglicht Prinzipalen das Abrufen von Informationen über Metadaten zu Ressourcen. CloudFront
- `cloudfront:Get*`— Ermöglicht Prinzipalen, detaillierte Informationen und Konfigurationen für CloudFront Ressourcen abzurufen.
- `cloudfront:List*`— Ermöglicht Prinzipalen das Abrufen von Ressourcenlisten. CloudFront
- `cloudfront-keyvaluestore:Describe*` – Ermöglicht es Prinzipalen, Informationen über den Schlüsselwertspeicher abzurufen.
- `cloudfront-keyvaluestore:Get*` – Ermöglicht es Prinzipalen, detaillierte Informationen und Konfigurationen für den Schlüsselwertspeicher abzurufen.
- `cloudfront-keyvaluestore:List*` – Ermöglicht es Prinzipalen, eine Liste der Schlüsselwertspeicher abzurufen.
- `acm:DescribeCertificate` – Ermöglicht es Prinzipalen, Details über ein ACM-Zertifikat abzurufen.
- `acm:ListCertificates` – Ermöglicht es Prinzipalen, eine Liste von ACM-Zertifikaten abzurufen.
- `iam:ListServerCertificates` – Ermöglicht es Prinzipalen, eine Liste der in IAM gespeicherten Serverzertifikate abzurufen.
- `route53:List*` – Ermöglicht es Prinzipalen, Listen von Route 53-Ressourcen abzurufen.
- `waf:ListWebACLs` – Ermöglicht Prinzipalen das Abrufen einer Liste von Web-ACLs in AWS WAF.
- `waf:GetWebACL` – Ermöglicht es Prinzipalen, detaillierte Informationen zu Web-ACLs in AWS WAF abzurufen.
- `wafv2:ListWebACLs` – Ermöglicht Prinzipalen das Abrufen einer Liste von Web-ACLs in AWS WAF.
- `wafv2:GetWebACL` – Ermöglicht es Prinzipalen, detaillierte Informationen zu Web-ACLs in AWS WAF abzurufen.

- `pricingplanmanager:GetSubscription`— Ermöglicht Prinzipalen nur Lesezugriff, um Details zu Preisplanabonnements abzurufen.
- `pricingplanmanager:ListSubscriptions`— Ermöglicht Prinzipalen den Lesezugriff auf Abonnements mit Listenpreisen.
- `ec2:DescribeIpamPools`— Ermöglicht Prinzipalen, detaillierte Informationen zu Ihren IPAM-Pools abzurufen.
- `ec2:GetIpamPoolCidrs`— Ermöglicht Prinzipalen das Abrufen der CIDRs, die für einen IPAM-Pool bereitgestellt werden.

Informationen zu den Berechtigungen für diese Richtlinie finden Sie unter CloudFrontReadOnlyAccess in der Referenz zu von AWS verwalteten Richtlinien.

AWS verwaltete Richtlinie: CloudFrontFullAccess

Sie können die CloudFrontFullAccess-Richtlinie an Ihre IAM-Identitäten anfügen. Diese Richtlinie ermöglicht Administratorberechtigungen für CloudFront Ressourcen. Sie ermöglicht auch Nur-Leseberechtigungen für andere AWS Dienstressourcen, die sich auf die Konsole beziehen CloudFront und in der CloudFront Konsole sichtbar sind.

Details zu Berechtigungen

Diese Richtlinie umfasst die folgenden Berechtigungen.

- `s3:ListAllMyBuckets` – Ermöglicht es Prinzipalen, eine Liste aller Amazon S3 Buckets abzurufen.
- `acm:DescribeCertificate` – Ermöglicht es Prinzipalen, Details über ein ACM-Zertifikat abzurufen.
- `acm:ListCertificates` – Ermöglicht es Prinzipalen, eine Liste von ACM-Zertifikaten abzurufen.
- `acm:RequestCertificate` – Ermöglicht es Prinzipalen, verwaltete Zertifikate von ACM anzufordern.
- `cloudfront:*`— Ermöglicht Prinzipalen, alle Aktionen für alle Ressourcen auszuführen. CloudFront
- `cloudfront-keyvaluestore:*` – Ermöglicht es Prinzipalen, alle Aktionen im Schlüsselwertspeicher durchzuführen.
- `iam:ListServerCertificates` – Ermöglicht es Prinzipalen, eine Liste der in IAM gespeicherten Serverzertifikate abzurufen.

- `waf:ListWebACLs` – Ermöglicht Prinzipalen das Abrufen einer Liste von Web-ACLs in AWS WAF.
- `waf:GetWebACL` – Ermöglicht es Prinzipalen, detaillierte Informationen zu Web-ACLs in AWS WAF abzurufen.
- `waf:CreateWebACLs`— Ermöglicht Prinzipalen das Erstellen einer Web-ACL in. AWS WAF
- `wafv2:ListWebACLs` – Ermöglicht Prinzipalen das Abrufen einer Liste von Web-ACLs in AWS WAF.
- `wafv2:GetWebACL` – Ermöglicht es Prinzipalen, detaillierte Informationen zu Web-ACLs in AWS WAF abzurufen.
- `kinesis:ListStreams` – Ermöglicht Prinzipalen das Abrufen einer Liste von Amazon Kinesis Streams.
- `elasticloadbalancing:DescribeLoadBalancers` – Ermöglicht es Prinzipalen, detaillierte Informationen über Load Balancer in Elastic Load Balancing abzurufen.
- `kinesis:DescribeStream` – Ermöglicht Prinzipalen das Abrufen detaillierter Informationen zu einem Kinesis Stream.
- `iam:ListRoles` – Erlaubt Prinzipalen das Abrufen einer Liste von Rollen in IAM.
- `pricingplanmanager:AssociateResourcesToSubscription`— Ermöglicht Prinzipalen, Ressourcen einem Abonnement zuzuordnen. Dadurch können die Ressourcen durch den Preisplan des Abonnements abgedeckt werden.
- `pricingplanmanager:CancelSubscription`- Ermöglicht Principals, ein bestehendes Abonnement zu kündigen.
- `pricingplanmanager:CancelSubscriptionChange`- Ermöglicht Prinzipalen, eine ausstehende Änderung an einem bestehenden Abonnement, z. B. ein Plan-Upgrade, zu stornieren, bevor die Änderung übernommen wird.
- `pricingplanmanager:CreateSubscription`- Ermöglicht Prinzipalen, ein Abonnement für einen Preisplan zu erstellen.
- `pricingplanmanager:DisassociateResourcesFromSubscription`- Ermöglicht Prinzipalen, die Zuordnung zwischen Ressourcen und einem bestehenden Abonnement aufzuheben.
- `pricingplanmanager:UpdateSubscription`- Ermöglicht Prinzipalen, ein bestehendes Abonnement zu ändern, z. B. den Preisplan zu ändern.
- `pricingplanmanager:GetSubscription`— Ermöglicht Prinzipalen nur Lesezugriff, um Details zu Preisplanabonnements abzurufen.

- `pricingplanmanager:ListSubscriptions`— Ermöglicht Prinzipalen den Lesezugriff auf Abonnements mit Listenpreisen.
- `pricingplanmanager:ApprovePaidSubscription`— Ermöglicht Prinzipalen, ein kostenpflichtiges Abonnement zu genehmigen und mit der Abrechnung zu beginnen.
- `freetier:GetAccountPlanState`— Ermöglicht Principals, den Status des AWS kostenlosen Kontingents für das Konto abzurufen, anhand dessen die Eignung für Pauschaltarife geprüft wird.
- `cloudwatch:GetMetricData`— Ermöglicht Prinzipalen das Abrufen von CloudWatch Metrikdaten, anhand derer festgestellt wird, ob sie für Pauschaltarife in Frage kommen.
- `shield:ListProtections`— Ermöglicht es Principals, AWS Shield-Schutzmaßnahmen aufzulisten, anhand derer festgestellt wird, ob Pauschalpreise in Frage kommen.
- `ec2:DescribeInstances` – Ermöglicht es Prinzipalen, detaillierte Informationen über Instances in Amazon EC2 abzurufen.
- `ec2:DescribeInternetGateways` – Ermöglicht es Prinzipalen, detaillierte Informationen über Internet-Gateways in Amazon EC2 abzurufen.
- `ec2:DescribeIpamPools`— Ermöglicht Principals, detaillierte Informationen über Ihre IPAM-Pools abzurufen.
- `ec2:GetIpamPoolCidrs`— Ermöglicht Prinzipalen das Abrufen der CIDRs, die für einen IPAM-Pool bereitgestellt werden.

Informationen zu den Berechtigungen für diese Richtlinie finden Sie unter CloudFrontFullAccess in der Referenz zu von AWS verwalteten Richtlinien.

> **Note**
>
> Die CloudFrontFullAccessRichtlinie beinhaltet die `pricingplanmanager:ApprovePaidSubscription` Genehmigung, mit der Ihr Konto zur Zahlung von Gebühren für kostenpflichtige Tarife verpflichtet wird, die erst am Ende des aktuellen Abrechnungszeitraums gekündigt werden können. Für automatisierte Workflows — wie Bereitstellungspipelines, Agenten oder Servicerollen — empfehlen wir eine abgegrenzte Richtlinie, bei der Zuschüsse auch ohne Genehmigung erteilt werden, sodass ein Mitarbeiter die Abrechnung autorisiert. `pricingplanmanager:CreateSubscription` `pricingplanmanager:ApprovePaidSubscription` Weitere Informationen finden Sie unter Erste Schritte mit der API. PricingPlanManager

> **Important**
>
> Wenn Sie Zugriffsprotokolle erstellen und speichern möchten CloudFront , müssen Sie zusätzliche Berechtigungen erteilen. Weitere Informationen finden Sie unter Berechtigungen.

AWS verwaltete Richtlinie: AWSCloudFrontLogger

Sie können die AWSCloudFrontLoggerRichtlinie nicht an Ihre IAM-Identitäten anhängen. Diese Richtlinie ist mit einer dienstbezogenen Rolle verknüpft, die es ermöglicht, Aktionen CloudFront in Ihrem Namen durchzuführen. Weitere Informationen finden Sie unter the section called “Service-linked Rollen für Lambda @Edge”.

Diese Richtlinie ermöglicht es CloudFront , Protokolldateien an Amazon zu übertragen CloudWatch. Details zu Berechtigungen dieser Richtlinie finden Sie unter the section called “Service-linked Rollenberechtigungen für Logger CloudFront”.

Informationen zu den Berechtigungen für diese Richtlinie finden Sie unter AWSCloudFrontLogger in der Referenz zu von AWS verwalteten Richtlinien.

AWS verwaltete Richtlinie: AWSLambdaReplicator

Sie können die AWSLambdaReplicatorRichtlinie nicht an Ihre IAM-Identitäten anhängen. Diese Richtlinie ist mit einer dienstbezogenen Rolle verknüpft, die es ermöglicht, Aktionen CloudFront in Ihrem Namen durchzuführen. Weitere Informationen finden Sie unter the section called “Service-linked Rollen für Lambda @Edge”.

Diese Richtlinie ermöglicht das Erstellen CloudFront , Löschen und Deaktivieren von Funktionen, AWS Lambda auf die Lambda @Edge -Funktionen repliziert werden sollen. AWS-Regionen Details zu Berechtigungen dieser Richtlinie finden Sie unter the section called “Service-linked Rollenberechtigungen für Lambda-Replikator”.

Informationen zu den Berechtigungen für diese Richtlinie finden Sie unter AWSLambdaReplicator in der Referenz zu von AWS verwalteten Richtlinien.

AWS verwaltete Richtlinie: AWSCloudFrontVPCOriginServiceRolePolicy

Sie können die AWSCloudFrontVPCOriginServiceRolePolicy-Richtlinie Ihren IAM-Entitäten nicht anfügen. Diese Richtlinie ist mit einer dienstbezogenen Rolle verknüpft, mit der CloudFront Sie

Aktionen in Ihrem Namen ausführen können. Weitere Informationen finden Sie unter Verwenden Sie serviceverknüpfte Rollen für CloudFront.

Diese Richtlinie ermöglicht CloudFront die Verwaltung elastischer EC2-Netzwerkschnittstellen und Sicherheitsgruppen in Ihrem Namen. Details zu Berechtigungen dieser Richtlinie finden Sie unter the section called “Dienstbezogene Rollenberechtigungen für CloudFront VPC Origins”.

Informationen zu den Berechtigungen für diese Richtlinie finden Sie unter AWSCloudFrontVPCOriginServiceRolePolicy in der Referenz zu von AWS verwalteten Richtlinien.

CloudFront Aktualisierungen für AWS Verwaltete Richtlinien

Hier finden Sie Informationen zu Aktualisierungen der AWS verwalteten Richtlinien, die CloudFront seit Beginn der Nachverfolgung dieser Änderungen durch diesen Dienst vorgenommen wurden. Abonnieren Sie den RSS-Feed auf der Seite CloudFront Dokumentenverlauf, um automatische Benachrichtigungen über Änderungen an dieser Seite zu erhalten.

Änderungen	Beschreibung	Date
CloudFrontFullAccess – Aktualisierung auf eine bestehende Richtlinie	CloudFront Es wurden neue Berechtigungen für die Inanspruchnahme von Pauschalpreisen und die Genehmigung von Abonnements hinzugefügt. Die neuen Berechtigungen ermöglichen es Prinzipalen, die Aktionen`pricingplanmanager:ApprovePaidSubscription` , `freetier:GetAccountPlanState` `cloudwatch:GetMetricData` , und zu verwenden. `shield:ListProtections`	29. Juli 2026

Änderungen	Beschreibung	Date
CloudFrontReadOnlyAccess – Aktualisierung auf eine bestehende Richtlinie	CloudFront neue Berechtigungen für Amazon EC2 hinzugefügt. Die neuen Berechtigungen ermöglichen es Prinzipalen, die Aktionen `ec2:DescribeIpamPools` und `ec2:GetIpamPoolCidrs` zu verwenden.	24. November 2025
CloudFrontFullAccess – Aktualisierung auf eine bestehende Richtlinie	CloudFront neue Berechtigungen für Amazon EC2 hinzugefügt. Die neuen Berechtigungen ermöglichen es Prinzipalen, die Aktionen `ec2:DescribeIpamPools` und `ec2:GetIpamPoolCidrs` zu verwenden.	24. November 2025
CloudFrontFullAccess – Aktualisierung auf eine bestehende Richtlinie	CloudFront hat eine neue Berechtigung zum Erstellen einer AWS WAF ACL-Ressource hinzugefügt und dem AWS Pricing Plan Manager Berechtigungen zum Erstellen, Aktualisieren, Löschen und Lesen hinzugefügt.	18. November 2025

Änderungen	Beschreibung	Date
CloudFrontFullAccess – Aktualisierung auf eine bestehende Richtlinie	CloudFront hat eine neue Berechtigung zum Erstellen einer AWS WAF ACL-Resso urce hinzugefügt und dem AWS Pricing Plan Manager Berechtigungen zum Erstellen , Aktualisieren, Löschen und Lesen hinzugefügt.	18. November 2025
CloudFrontReadOnlyAccess – Aktualisierung auf eine bestehende Richtlinie	CloudFront neue Berechtig ungen für den Nur-Lese-Zugriff auf den AWS Pricing Plan Manager hinzugefügt.	18. November 2025
CloudFrontReadOnlyAccess – Aktualisierung auf eine bestehende Richtlinie	CloudFront neue Berechtig ungen für den Nur-Lese-Zugriff auf den AWS Pricing Plan Manager hinzugefügt.	18. November 2025
CloudFrontReadOnlyAccess – Aktualisierung auf eine bestehende Richtlinie	CloudFront neue Erlaubnis für ACM hinzugefügt. Diese neue Berechtigung ermöglicht es Prinzipalen, Details zu einem ACM-Zerti fikat abzurufen.	28. April 2025
CloudFrontFullAccess – Aktualisierung auf eine bestehende Richtlinie	CloudFront neue Berechtig ungen für ACM hinzugefügt. Diese neuen Berechtigungen ermöglichen es Prinzipal en, Details zu einem ACM-Zertifikat abzurufen und ein verwaltetes Zertifikat von ACM anzufordern.	28. April 2025

Änderungen	Beschreibung	Date
CloudFrontFullAccess – Aktualisierung auf eine bestehende Richtlinie	CloudFront neue Berechtigungen für Amazon EC2 und Elastic Load Balancing hinzugefügt. Die neuen Berechtigungen ermöglichen es CloudFront , detaillierte Informationen über Load Balancer in Elastic Load Balancing und Instances und Internet-Gateways in Amazon EC2 abzurufen.	20. November 2024
AWSCloudFrontVPCOriginServiceRolePolicy – Neue Richtlinie	CloudFront hat eine neue Richtlinie hinzugefügt. Diese Richtlinie ermöglicht CloudFront die Verwaltung elastischer EC2-Netzwerkschnittstellen und Sicherheitsgruppen in Ihrem Namen.	20. November 2024
CloudFrontReadOnlyAccess und CloudFrontFullAccess – Update von zwei vorhandenen Richtlinien.	CloudFront neue Berechtigungen für Key-Value-Stores hinzugefügt. Die neuen Berechtigungen ermöglichen es Benutzern, Informationen über Schlüsselwertspeicher abzurufen und entsprechende Maßnahmen zu ergreifen.	19. Dezember 2023

Änderungen	Beschreibung	Date
CloudFrontReadOnlyAccess – Aktualisierung auf eine bestehende Richtlinie	CloudFront hat eine neue Berechtigung zur Beschreib ung von CloudFront Funktione n hinzugefügt. Diese Berechtigung ermöglicht es dem Benutzer, der Gruppe oder der Rolle, Informati onen und Metadaten zu einer Funktion zu lesen, nicht jedoch den Code der Funktion.	8. September 2021
CloudFront hat begonnen, Änderungen zu verfolgen	CloudFront hat begonnen, Änderungen für die AWS verwalteten Richtlinien zu verfolgen.	8. September 2021

Verwenden Sie serviceverknüpfte Rollen für CloudFront

Amazon CloudFront verwendet AWS Identity and Access Management (IAM) serviceverknüpfte Rollen. Eine serviceverknüpfte Rolle ist eine einzigartige Art von IAM-Rolle, mit der direkt verknüpft ist. CloudFront Mit Diensten verknüpfte Rollen sind vordefiniert CloudFront und enthalten alle Berechtigungen, die der Dienst benötigt, um andere AWS Dienste in Ihrem Namen aufzurufen.

Eine dienstbezogene Rolle CloudFront erleichtert die Einrichtung, da Sie die erforderlichen Berechtigungen nicht manuell hinzufügen müssen. CloudFront definiert die Berechtigungen ihrer dienstbezogenen Rollen und CloudFront kann, sofern nicht anders definiert, nur ihre Rollen übernehmen. Die definierten Berechtigungen umfassen die Vertrauens- und Berechtigungsrichtlinie. Diese Berechtigungsrichtlinie kann keinen anderen IAM-Entitäten zugewiesen werden.

Sie können eine serviceverknüpfte Rolle erst löschen, nachdem ihre verwandten Ressourcen gelöscht wurden. Dadurch werden Ihre CloudFront Ressourcen geschützt, da Sie nicht versehentlich die Zugriffsberechtigung für die Ressourcen entziehen können.

Informationen zu anderen Diensten, die dienstverknüpfte Rollen unterstützen, finden Sie unter AWS Dienste, die mit IAM funktionieren. Suchen Sie in der Spalte Dienstverknüpfte Rollen nach

den Diensten, für die Ja steht. Wählen Sie über einen Link Ja aus, um die Dokumentation zu einer serviceverknüpften Rolle für diesen Service anzuzeigen.

Dienstbezogene Rollenberechtigungen für CloudFront VPC Origins

CloudFront VPC Origins verwendet die serviceverknüpfte Rolle mit dem Namen AWSServiceRoleForCloudFrontVPCOrigin— Ermöglicht CloudFront die Verwaltung elastischer EC2-Netzwerkschnittstellen und Sicherheitsgruppen in Ihrem Namen.

Die serviceverknüpfte Rolle AWSServiceRoleForCloudFrontVPCOrigin vertraut darauf, dass die folgenden Services die Rolle annehmen:

- `vpcorigin.cloudfront.amazonaws.com`

Die Rollenberechtigungsrichtlinie mit dem Namen AWSCloud Front VPCOrigin ServiceRolePolicy ermöglicht es CloudFront VPC Origins, die folgenden Aktionen für die angegebenen Ressourcen durchzuführen:

- Aktion: `ec2:CreateNetworkInterface` für `arn:aws:ec2:*:*:network-interface/*`
- Aktion: `ec2:CreateNetworkInterface` für `arn:aws:ec2:*:*:subnet/*` und `arn:aws:ec2:*:*:security-group/*`
- Aktion: `ec2:CreateSecurityGroup` für `arn:aws:ec2:*:*:security-group/*`
- Aktion: `ec2:CreateSecurityGroup` für `arn:aws:ec2:*:*:vpc/*`
- Aktion: `ec2:ModifyNetworkInterfaceAttribute`, `ec2:DeleteNetworkInterface`, `ec2:DeleteSecurityGroup`, `ec2:AssignIpv6Addresses` und `ec2:UnassignIpv6Addresses` für `supported AWS resources that have the aws:ResourceTag/aws.cloudfront.vpcorigin tag enabled`
- Aktion: `ec2:DescribeNetworkInterfaces`, `ec2:DescribeSecurityGroups`, `ec2:DescribeInstances`, `ec2:DescribeInternetGateways`, `ec2:DescribeSubnets`, `ec2:DescribeRegions` und `ec2:DescribeAddresses` für `all AWS resources that the actions support`
- Aktion: `ec2:CreateTags` und `arn:aws:ec2:*:*:security-group/*` für `arn:aws:ec2:*:*:network-interface/*`
- Aktion: `elasticloadbalancing:DescribeLoadBalancers`, `elasticloadbalancing:DescribeListeners` und

`elasticloadbalancing:DescribeTargetGroups` für `all AWS resources that the actions support`

Sie müssen Berechtigungen konfigurieren, damit eine Benutzer, Gruppen oder Rollen eine serviceverknüpfte Rolle erstellen, bearbeiten oder löschen können. Weitere Informationen finden Sie unter serviceverknüpfte Rollenberechtigung im IAM-Benutzerhandbuch.

Eine serviceverknüpfte Rolle für CloudFront VPC Origins erstellen

Sie müssen eine serviceverknüpfte Rolle nicht manuell erstellen. Wenn Sie einen VPC-Ursprung in der AWS-Managementkonsole, der oder der AWS API erstellen AWS CLI, erstellt CloudFront VPC Origins die serviceverknüpfte Rolle für Sie.

Wenn Sie diese serviceverknüpfte Rolle löschen und sie dann erneut erstellen müssen, können Sie dasselbe Verfahren anwenden, um die Rolle in Ihrem Konto neu anzulegen. Wenn Sie einen VPC-Ursprung erstellen, erstellt CloudFront VPC Origins die serviceverknüpfte Rolle erneut für Sie.

Eine serviceverknüpfte Rolle für CloudFront VPC Origins bearbeiten

CloudFront In VPC Origins können Sie die AWSServiceRoleForCloudFrontVPCOrigin serviceverknüpfte Rolle nicht bearbeiten. Da möglicherweise verschiedene Entitäten auf die Rolle verweisen, kann der Rollenname nach dem Erstellen einer serviceverknüpften Rolle nicht mehr geändert werden. Sie können jedoch die Beschreibung der Rolle mit IAM bearbeiten. Weitere Informationen finden Sie unter Bearbeiten einer serviceverknüpften Rolle im IAM-Benutzerhandbuch.

Löschen Sie eine serviceverknüpfte Rolle für CloudFront VPC Origins

Wenn Sie ein Feature oder einen Dienst, die bzw. der eine serviceverknüpften Rolle erfordert, nicht mehr benötigen, sollten Sie diese Rolle löschen. Auf diese Weise haben Sie keine ungenutzte juristische Stelle, die nicht aktiv überwacht oder verwaltet wird. Sie müssen jedoch die Ressourcen für Ihre serviceverknüpften Rolle zunächst bereinigen, bevor Sie sie manuell löschen können.

> **Note**
>
> Wenn der CloudFront Dienst die Rolle verwendet, wenn Sie versuchen, die Ressourcen zu löschen, schlägt das Löschen möglicherweise fehl. Wenn dies passiert, warten Sie einige Minuten und versuchen Sie es erneut.

Um CloudFront VPC Origins-Ressourcen zu löschen, die von AWSServiceRoleForCloudFrontVPCOrigin

- Löschen Sie die VPC-Ursprungsressourcen in Ihrem Konto.
  - Es kann einige Zeit dauern CloudFront , bis das Löschen der Ressourcen aus Ihrem Konto abgeschlossen ist. Wenn Sie die serviceverknüpfte Rolle nicht sofort löschen können, warten Sie und versuchen Sie es erneut.

So löschen Sie die serviceverknüpfte Rolle mit IAM

Verwenden Sie die IAM-Konsole, die oder die AWS API AWS CLI, um die AWSServiceRoleForCloudFrontVPCOrigin serviceverknüpfte Rolle zu löschen. Weitere Informationen finden Sie unter Löschen einer serviceverknüpften Rolle im IAM-Benutzerhandbuch.

Unterstützte Regionen für serviceverknüpfte Rollen mit CloudFront VPC Origins

CloudFront VPC Origins unterstützt nicht die Verwendung von serviceverknüpften Rollen in allen Regionen, in denen der Service verfügbar ist. Sie können die Rolle AWSServiceRoleForCloudFrontVPCOrigin in den folgenden Regionen verwenden.

Name der Region	Regions-ID	Support in CloudFront
USA Ost (Nord-Virginia)	us-east-1	Ja
USA Ost (Ohio)	us-east-2	Ja
USA West (Nordkalifornien)	us-west-1 (außer AZ usw1-az2)	Ja
USA West (Oregon)	us-west-2	Ja
Afrika (Kapstadt)	af-south-1	Ja
Asien-Pazifik (Hongkong)	ap-east-1	Ja
Asien-Pazifik (Jakarta)	ap-southeast-3	Ja
Asien-Pazifik (Melbourne)	ap-southeast-4	Ja

Name der Region	Regions-ID	Support in CloudFront
Asien-Pazifik (Mumbai)	ap-south-1	Ja
Asien-Pazifik (Hyderabad)	ap-south-2	Ja
Asien-Pazifik (Osaka)	ap-northeast-3	Ja
Asien-Pazifik (Seoul)	ap-northeast-2	Ja
Asien-Pazifik (Singapore)	ap-southeast-1	Ja
Asien-Pazifik (Sydney)	ap-southeast-2	Ja
Asien-Pazifik (Tokyo)	ap-northeast-1 (außer AZ apne1-az3)	Ja
Kanada (Zentral)	ca-central-1 (außer AZ cac1-az3)	Ja
Kanada West (Calgary)	ca-west-1	Ja
Europa (Frankfurt)	eu-central-1	Ja
Europa (Ireland)	eu-west-1	Ja
Europa (London)	eu-west-2	Ja
Europa (Milan)	eu-south-1	Ja
Europa (Paris)	eu-west-3	Ja
Europa (Spain)	eu-south-2	Ja
Europa (Stockholm)	eu-north-1	Ja
Europa (Zürich)	eu-central-2	Ja
Israel (Tel Aviv)	il-central-1	Ja
Naher Osten (Bahrain)	me-south-1	Ja

Name der Region	Regions-ID	Support in CloudFront
Naher Osten (VAE)	me-central-1	Ja
Südamerika (São Paulo)	sa-east-1	Ja

Beheben Sie Probleme mit der CloudFront Identität und dem Zugriff von Amazon

Verwenden Sie die folgenden Informationen, um häufig auftretende Probleme zu diagnostizieren und zu beheben, die bei der Arbeit mit einem CloudFront IAM auftreten können.

Themen

- Ich bin nicht berechtigt, eine Aktion auszuführen in CloudFront
- Ich bin nicht berechtigt, IAM auszuführen: PassRole
- Ich möchte Leute außerhalb von mir zulassen AWS-Konto um auf meine CloudFront Ressourcen zuzugreifen

Ich bin nicht berechtigt, eine Aktion auszuführen in CloudFront

Wenn Sie eine Fehlermeldung erhalten, dass Sie nicht zur Durchführung einer Aktion berechtigt sind, müssen Ihre Richtlinien aktualisiert werden, damit Sie die Aktion durchführen können.

Der folgende Beispielfehler tritt auf, wenn der IAM-Benutzer `mateojackson` versucht, über die Konsole Details zu einer fiktiven *`my-example-widget`*-Ressource anzuzeigen, jedoch nicht über `cloudfront:`*`GetWidget`*-Berechtigungen verfügt.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 cloudfront:GetWidget on resource: my-example-widget
```

In diesem Fall muss die Richtlinie für den Benutzer `mateojackson` aktualisiert werden, damit er mit der `cloudfront:`*`GetWidget`*-Aktion auf die *`my-example-widget`*-Ressource zugreifen kann.

Wenn Sie Hilfe benötigen, wenden Sie sich an Ihren AWS Administrator. Ihr Administrator hat Ihnen Ihre Anmeldeinformationen zur Verfügung gestellt.

Ich bin nicht berechtigt, IAM auszuführen: PassRole

Wenn Sie die Fehlermeldung erhalten, dass Sie nicht zum Durchführen der `iam:PassRole`-Aktion autorisiert sind, müssen Ihre Richtlinien aktualisiert werden, um eine Rolle an CloudFront übergeben zu können.

In einigen AWS-Services Fällen können Sie eine vorhandene Rolle an diesen Dienst übergeben, anstatt eine neue oder mit einem Dienst verknüpfte Rolle zu erstellen. Hierzu benötigen Sie Berechtigungen für die Übergabe der Rolle an den Dienst.

Der folgende Beispielfehler tritt auf, wenn ein IAM-Benutzer mit dem Namen `marymajor` versucht, die Konsole zu verwenden, um eine Aktion in CloudFront auszuführen. Die Aktion erfordert jedoch, dass der Service über Berechtigungen verfügt, die durch eine Servicerolle gewährt werden. Mary besitzt keine Berechtigungen für die Übergabe der Rolle an den Dienst.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

In diesem Fall müssen die Richtlinien von Mary aktualisiert werden, um die Aktion `iam:PassRole` ausführen zu können.

Wenn Sie Hilfe benötigen, wenden Sie sich an Ihren AWS Administrator. Ihr Administrator hat Ihnen Ihre Anmeldeinformationen zur Verfügung gestellt.

Ich möchte Leute außerhalb von mir zulassen AWS-Konto um auf meine CloudFront Ressourcen zuzugreifen

Sie können eine Rolle erstellen, mit der Benutzer in anderen Konten oder Personen außerhalb Ihrer Organisation auf Ihre Ressourcen zugreifen können. Sie können festlegen, wem die Übernahme der Rolle anvertraut wird. Im Fall von Diensten, die ressourcenbasierte Richtlinien oder Zugriffskontrolllisten (Access Control Lists, ACLs) verwenden, können Sie diese Richtlinien verwenden, um Personen Zugriff auf Ihre Ressourcen zu gewähren.

Weitere Informationen dazu finden Sie hier:

- Informationen darüber, ob diese Funktionen CloudFront unterstützt werden, finden Sie unterWie Amazon mit CloudFront IAM arbeitet.
- Informationen zum Bereitstellen des Zugriffs auf Ihre Ressourcen in Ihrem eigenen AWS-Konten Bereich finden Sie im IAM-Benutzerhandbuch unter Bereitstellen des Zugriffs für einen IAM-Benutzer in einem anderen Benutzer AWS-Konto , dessen Eigentümer Sie sind.

- Informationen dazu, wie Sie Dritten Zugriff auf Ihre Ressourcen gewähren AWS-Konten, finden Sie im IAM-Benutzerhandbuch unter Gewähren des Zugriffs auf Ressourcen, die Dritten AWS-Konten gehören.
- Informationen dazu, wie Sie über einen Identitätsverbund Zugriff gewähren, finden Sie unter Gewähren von Zugriff für extern authentifizierte Benutzer (Identitätsverbund) im IAM-Benutzerhandbuch.
- Informationen zum Unterschied zwischen der Verwendung von Rollen und ressourcenbasierten Richtlinien für den kontoübergreifenden Zugriff finden Sie unter Kontoübergreifender Ressourcenzugriff in IAM im IAM-Benutzerhandbuch.

Protokollierung und Überwachung in Amazon CloudFront

Die Überwachung ist ein wichtiger Bestandteil der Aufrechterhaltung der Verfügbarkeit CloudFront und Leistung Ihrer AWS Lösungen. Sie sollten Überwachungsdaten aus allen Teilen Ihrer AWS Lösung sammeln, damit Sie einen etwaigen Ausfall an mehreren Stellen leichter debuggen können. AWS bietet verschiedene Tools zur Überwachung Ihrer CloudFront Ressourcen und Aktivitäten sowie zur Reaktion auf potenzielle Vorfälle:

CloudWatch Amazon-Alarme

Mithilfe von CloudWatch Alarmen beobachten Sie eine einzelne Metrik über einen von Ihnen festgelegten Zeitraum. Wenn die Metrik einen bestimmten Schwellenwert überschreitet, wird eine Benachrichtigung an ein Amazon SNS SNS-Thema oder eine AWS Auto Scaling Richtlinie gesendet. CloudWatch Alarme lösen keine Aktionen aus, wenn sich eine Metrik in einem bestimmten Status befindet. Der Status muss sich stattdessen geändert haben und für eine festgelegte Anzahl an Zeiträumen aufrechterhalten worden sein.

Weitere Informationen finden Sie unter Überwachen Sie CloudFront Metriken mit Amazon CloudWatch.

AWS CloudTrail Logs

CloudTrail bietet eine Aufzeichnung der API-Aktionen, die von einem Benutzer, einer Rolle oder einem AWS Dienst in ausgeführt wurden CloudFront. Anhand der von gesammelten Informationen können Sie die API-Anfrage ermitteln CloudTrail, an die die Anfrage gestellt wurde CloudFront, die IP-Adresse, von der aus die Anfrage gestellt wurde, wer die Anfrage gestellt hat, wann sie gestellt wurde, und weitere Details.

Weitere Informationen finden Sie unter Protokollieren Amazon CloudFront Amazon-API-Aufrufen mit AWS CloudTrail.

CloudFront Standardprotokolle und Zugriffsprotokolle in Echtzeit

CloudFront Protokolle enthalten detaillierte Aufzeichnungen über Anfragen, die an eine Distribution gestellt werden. Diese Protokolle sind für viele Anwendungen nützlich. Beispielsweise können Protokollinformationen bei Sicherheits- und Zugriffsprüfungen nützlich sein.

Weitere Informationen erhalten Sie unter Zugriffsprotokolle (Standardprotokolle) und Erstellen und verwenden Sie Konfigurationen für Echtzeit-Zugriffsprotokolle.

Protokolle für Edge-Funktionen

Von Edge-Funktionen generierte Protokolle, sowohl CloudFront Functions als auch Lambda @Edge, werden direkt an Amazon CloudWatch Logs gesendet und nirgends von CloudFront gespeichert. CloudFront Functions verwendet eine dienstbezogene AWS Identity and Access Management (IAM) -Rolle, um von Kunden generierte Protokolle direkt an Logs in Ihrem Konto zu CloudWatch senden.

Weitere Informationen finden Sie unter Protokolle für Edge-Funktionen.

CloudFront Konsolenberichte

Die CloudFront Konsole enthält eine Vielzahl von Berichten, darunter den Cache-Statistikbericht, den Bericht über beliebte Objekte und den Bericht mit den häufigsten Verweisen. Die meisten CloudFront Konsolenberichte basieren auf den Daten in den CloudFront Zugriffsprotokollen, die detaillierte Informationen über jede eingehende Benutzeranfrage enthalten CloudFront . Sie müssen Zugriffsprotokolle nicht aktivieren, um die Berichte anzeigen zu können.

Weitere Informationen finden Sie unter CloudFront Berichte in der Konsole anzeigen.

Konformitätsüberprüfung für Amazon CloudFront

Third-party Auditoren bewerten die Sicherheit und Compliance von Amazon im CloudFront Rahmen mehrerer AWS Compliance-Programme. Zu diesen Programmen gehören SOC, PCI, HIPAA und andere.

Eine Liste der AWS Services, die im Rahmen bestimmter Compliance-Programme angeboten werden, finden Sie unter AWS Services im Umfang nach Compliance-Programmen. Allgemeine Informationen finden Sie unter AWS -Compliance-Programme.

Auditberichte von Drittanbietern können Sie mit herunterladen AWS Artifact. Weitere Informationen finden Sie unter Berichte herunterladen in AWS Artifact.

Ihre Verantwortung für die Einhaltung der Vorschriften bei der Nutzung CloudFront hängt von der Vertraulichkeit Ihrer Daten, den Compliance-Zielen Ihres Unternehmens sowie den geltenden Gesetzen und Vorschriften ab. AWS stellt die folgenden Ressourcen zur Verfügung, um Sie bei der Einhaltung der Vorschriften zu unterstützen:

- Schnellstartanleitungen zu Sicherheit und Compliance — In diesen Bereitstellungshandbüchern werden architektonische Überlegungen erörtert und Schritte für die Bereitstellung sicherheits- und Compliance-orientierter Basisumgebungen beschrieben. AWS
- Architektur für HIPAA-Sicherheit und -Compliance am AWS — In diesem Whitepaper wird beschrieben, wie Unternehmen Anwendungen erstellen können. AWS HIPAA-compliant

  Das AWS HIPAA-Compliance-Programm umfasst CloudFront (mit Ausnahme der Bereitstellung von Inhalten über CloudFront eingebettete PoPs) als HIPAA-fähigen Service. Wenn Sie einen Business Associate Addendum (BAA) mit abgeschlossen haben AWS, können Sie diesen (mit Ausnahme der Inhaltsbereitstellung über CloudFront eingebettete POPs) verwenden, um Inhalte bereitzustellen, die geschützte Gesundheitsinformationen CloudFront (PHI) enthalten. Weitere Informationen finden Sie unter HIPAA-Compliance.
- AWS Ressourcen zur Einhaltung von Vorschriften — Diese Sammlung von Arbeitsmappen und Leitfäden könnte für Ihre Branche und Ihren Standort gelten.
- AWS Config— Dieser AWS Service bewertet, wie gut Ihre Ressourcenkonfigurationen den internen Praktiken, Branchenrichtlinien und Vorschriften entsprechen.
- AWS Security Hub CSPM— Dieser AWS Service nutzt Sicherheitskontrollen, um Ressourcenkonfigurationen und Sicherheitsstandards zu evaluieren, damit Sie die verschiedenen Compliance-Frameworks einhalten können. Weitere Informationen zur Verwendung von Security Hub CSPM zur Bewertung von CloudFront Ressourcen finden Sie unter Amazon CloudFront Controls im AWS Security Hub CSPM Benutzerhandbuch.

CloudFront Bewährte Verfahren zur Einhaltung

In diesem Abschnitt finden Sie bewährte Verfahren und Empfehlungen zur Einhaltung der Vorschriften, wenn Sie Amazon für CloudFront die Bereitstellung Ihrer Inhalte verwenden.

Sie können HIPAA-compliant Workloads ausführen PCI-compliant , die auf dem Modell der AWS gemeinsamen Verantwortung basieren. In diesem Fall empfehlen wir, dass Sie Ihre CloudFront

Nutzungsdaten der letzten 365 Tage für zukünftige Prüfzwecke protokollieren. Sie können Nutzungsdaten wie folgt protokollieren:

- Aktivieren CloudFront Sie Zugriffsprotokolle. Weitere Informationen finden Sie unter Zugriffsprotokolle (Standardprotokolle).
- Erfassen Sie Anfragen, die an die CloudFront API gesendet werden. Weitere Informationen finden Sie unter Protokollieren Amazon CloudFront Amazon-API-Aufrufen mit AWS CloudTrail.

Darüber hinaus finden Sie im Folgenden Einzelheiten zur Einhaltung CloudFront der PCI-DSS- und SOC-Standards.

Payment Card Industry Data Security Standard (PCI DSS)

CloudFront (mit Ausnahme der Inhaltsbereitstellung über CloudFront eingebettete PoPs) unterstützt die Verarbeitung, Speicherung und Übertragung von Kreditkartendaten durch einen Händler oder Dienstanbieter und wurde als konform mit dem Payment Card Industry (PCI) Data Security Standard (DSS) validiert. Weitere Informationen zu PCI DSS, einschließlich der Frage, wie Sie eine Kopie des AWS PCI Compliance Package anfordern können, finden Sie unter PCI DSS Level 1.

Aus Sicherheitsgründen empfehlen wir, Kreditkarteninformationen nicht in CloudFront Edge-Caches zwischenzuspeichern. Beispielsweise können Sie Ihren Ursprung so konfigurieren, dass Antworten, die Kreditkarteninformationen enthalten, wie z. B. die letzten vier Ziffern einer Kreditkartennummer und die Kontaktinformationen des Karteninhabers, eine `Cache-Control:no-cache="` *`field-name`* `"` Kopfzeile enthalten.

System and Organization Controls (SOC)

CloudFront (mit Ausnahme der Inhaltsbereitstellung über CloudFront eingebettete PoPs) entspricht den Maßnahmen zur System- und Organisationskontrolle (SOC), einschließlich SOC 1, SOC 2 und SOC 3. SOC-Berichte sind unabhängige Prüfungsberichte von Drittanbietern, aus denen hervorgeht, wie wichtige Compliance-Kontrollen und -Ziele AWS erreicht werden. Diese Audits stellen sicher, dass die geeigneten Sicherheitsvorkehrungen und Verfahren zum Schutz vor Risiken getroffen wurden, die die Sicherheit, Vertraulichkeit und Verfügbarkeit von Kunden- und Unternehmensdaten beeinträchtigen könnten. Die Ergebnisse dieser Prüfungen durch Dritte sind auf der AWS SOC-Compliance-Website verfügbar. Dort finden Sie die veröffentlichten Berichte mit weiteren Informationen zu den Kontrollen, die den AWS Betrieb und die Einhaltung von Vorschriften unterstützen.

Ausfallsicherheit in Amazon CloudFront

Im Zentrum der globalen AWS-Infrastruktur stehen die AWS-Regionen und Availability Zones (Verfügbarkeitszonen, AZs). AWS Regionen stellen mehrere physisch getrennte und isolierte Availability Zones bereit, die mit Netzwerken mit geringer Latenz, hohem Durchsatz und hochredundanten Vernetzungen verbunden sind. Mithilfe von Availability Zones können Sie Anwendungen und Datenbanken erstellen und ausführen, die automatisch Failover zwischen Availability Zones ausführen, ohne dass es zu Unterbrechungen kommt. Availability Zones sind besser hoch verfügbar, fehlertoleranter und skalierbarer als herkömmliche Infrastrukturen mit einem oder mehreren Rechenzentren.

Weitere Informationen über AWS-Regionen und -Availability Zones finden Sie unter AWS Globale Infrastruktur.

Origin Failover in CloudFront

Neben der Unterstützung der globalen AWS-Infrastruktur bietet Amazon CloudFront die Funktion Origin Failover, um Ihren Anforderungen an die Datenresilienz gerecht zu werden. CloudFront ist ein globaler Service, der Ihre Inhalte über ein weltweites Netzwerk von Rechenzentren bereitstellt, die als Edge-Standorte oder Points of Presence (POP) bezeichnet werden. Wenn Ihre Inhalte nicht bereits an einem Edge-Standort zwischengespeichert sind, ruft CloudFront sie aus einem Ursprungs-Server ab, den Sie als Quelle für die definitive Version des Inhalts identifiziert haben.

Sie können die Ausfallsicherheit und Verfügbarkeit für bestimmte Szenarien erhöhen, indem Sie CloudFront mit einem Origin Failover einrichten. Erstellen Sie zuerst eine Ursprungsgruppe, in der Sie einen primären Ursprung plus einen zweiten Ursprung für CloudFront festlegen. CloudFront wechselt automatisch zum zweiten Ursprung, wenn der primäre Ursprung bestimmte HTTP-Statuscode-Fehlerantworten ausgibt. Weitere Informationen finden Sie unter Optimieren Sie die Hochverfügbarkeit mit CloudFront Origin-Failover.

Infrastruktursicherheit bei Amazon CloudFront

Als verwalteter Service CloudFront ist Amazon durch AWS globale Netzwerksicherheit geschützt. Informationen zu AWS Sicherheitsdiensten und zum AWS Schutz der Infrastruktur finden Sie unter AWS Cloud-Sicherheit. Informationen zum Entwerfen Ihrer AWS Umgebung anhand der Best Practices für die Infrastruktursicherheit finden Sie unter Infrastrukturschutz im Security Pillar AWS Well-Architected Framework.

Für den Zugriff CloudFront über das Netzwerk verwenden Sie AWS veröffentlichte API-Aufrufe. Kunden müssen Folgendes unterstützen:

- Transport Layer Security (TLS). Wir benötigen TLS 1.2 und empfehlen TLS 1.3.
- Verschlüsselungssammlungen mit Perfect Forward Secrecy (PFS) wie DHE (Ephemeral) oder ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Diffie-Hellman Die meisten modernen Systeme wie Java 7 und höher unterstützen diese Modi.

CloudFront Functions verwendet eine hochsichere Isolationsbarriere zwischen AWS Konten und stellt so sicher, dass die Kundenumgebungen vor Seitenkanalangriffen wie Spectre und Meltdown geschützt sind. Funktionen können nicht auf Daten anderer Kunden zugreifen oder diese ändern. Funktionen werden in einem dedizierten Single-Thread-Prozess auf einer dedizierten CPU ohne Hyperthreading ausgeführt An jedem beliebigen CloudFront Edge Location Point of Presence (POP) bedient CloudFront Functions jeweils nur einen Kunden, und alle kundenspezifischen Daten werden zwischen den Funktionsausführungen gelöscht.

Fehlerbehebung

Verwenden Sie diesen Abschnitt, um häufig auftretende Probleme zu beheben, die auftreten können, wenn Sie Amazon CloudFront für den Vertrieb Ihrer Inhalte einrichten.

Jedes Thema enthält detaillierte Anleitungen zur Identifizierung der Hauptursache häufiger Probleme sowie schrittweise Anleitungen zu deren Behebung.

Themen

- Fehlerbehebung bei Problemen mit der Distribution
- Problembehandlung bei Statuscodes für die Fehlerantwort in CloudFront
- Testen CloudFront

Fehlerbehebung bei Problemen mit der Distribution

Mithilfe der hier aufgeführten Informationen können Sie Zertifikatsfehler, Probleme mit Zugriffsverweigerungen oder andere häufig auftretende Probleme diagnostizieren und beheben, die bei der Einrichtung Ihrer Website oder Anwendung mit CloudFront Amazon-Distributionen auftreten können.

Themen

- CloudFront gibt eine zurück Zugriff verweigert error
- CloudFront gibt eine zurück InvalidViewerCertificate Fehler beim Versuch, einen alternativen Domainnamen hinzuzufügen
- CloudFront gibt einen falsch konfigurierten DNS-Eintragsfehler zurück, wenn ich versuche, einen neuen CNAME hinzuzufügen
- Ich kann die Dateien in meiner Distribution nicht anzeigen.
- Fehlermeldung: Certificate: < certificate-id > wird verwendet von CloudFront

CloudFront gibt eine zurück Zugriff verweigert error

Wenn Sie einen Amazon S3-Bucket als Ursprung für Ihre CloudFront Distribution verwenden, wird in den folgenden Beispielen möglicherweise die Fehlermeldung Access Denied (403) angezeigt.

Inhalt

- Sie haben ein fehlendes Objekt aus dem Amazon-S3-Ursprung angegeben
- Ihrem Amazon-S3-Ursprungs-Server fehlen IAM-Berechtigungen
- Sie verwenden ungültige Anmeldeinformationen oder verfügen nicht über ausreichende Berechtigungen

Sie haben ein fehlendes Objekt aus dem Amazon-S3-Ursprung angegeben

Stellen Sie sicher, dass das angeforderte Objekt in Ihrem Bucket vorhanden ist. Bei den Objektnamen muss die Groß- und Kleinschreibung beachtet werden. Wenn Sie einen ungültigen Objektnamen eingeben, kann dies den Fehlercode „Zugriff verweigert“ zurückgeben.

Wenn Sie beispielsweise dem CloudFront Tutorial zur Erstellung einer Basisverteilung folgen, erstellen Sie einen Amazon S3-Bucket als Ursprung und laden eine `index.html` Beispieldatei hoch.

Wenn Sie in Ihrem Webbrowser `https://d111111abcdef8.cloudfront.net/INDEX.HTML` statt `https://d111111abcdef8.cloudfront.net/index.html` eingeben, wird möglicherweise eine ähnliche Meldung angezeigt, da bei der `index.html`-Datei im URL-Pfad zwischen Groß- und Kleinschreibung unterschieden wird.

```
<Error>
<Code>AccessDenied</Code>
<Message>Access Denied</Message>
<RequestId>22Q367AHT7Y1ABCD</RequestId>
<HostId>
ABCDE/Vg+7PSNa/d/IfFQ8Fb92TGQ0KH0ZwG5iEKbc6+e06DdMS1ZW+ryB9GFRIVtS66rSSy6So=
</HostId>
</Error>
```

Ihrem Amazon-S3-Ursprungs-Server fehlen IAM-Berechtigungen

Stellen Sie sicher, dass Sie den richtigen Amazon-S3-Bucket als Ursprungsdomain und -namen ausgewählt haben. Der Ursprung (Amazon S3) muss über die richtigen Berechtigungen verfügen.

Wenn Sie nicht die richtigen Berechtigungen angeben, kann Ihren Viewern eine `AccessDenied`-Meldung angezeigt werden.

Wenn Sie Inhalte von Amazon S3 verteilen und außerdem AWS Key Management Service (AWS KMS) serviceseitige Verschlüsselung (SSE-KMS) verwenden, müssen Sie zusätzliche IAM-Berechtigungen für den KMS-Schlüssel und den Amazon S3-Bucket angeben. Ihre CloudFront

Distribution benötigt diese Berechtigungen, um den KMS-Schlüssel zu verwenden, der für die Verschlüsselung des Amazon S3-Quell-Buckets verwendet wird.

Die Konfigurationen der Amazon S3-Bucket-Richtlinie ermöglichen es der CloudFront Verteilung, die verschlüsselten Objekte für die Inhaltsbereitstellung abzurufen.

So überprüfen Sie Ihre Schlüsselberechtigungen für Amazon-S3-Bucket und KMS

1. Stellen Sie sicher, dass der KMS-Schlüssel, den Sie verwenden, derselbe Schlüssel ist, den Ihr Amazon-S3-Bucket für die Standardverschlüsselung verwendet. Weitere Informationen finden Sie unter Angeben der serverseitigen Verschlüsselung mit AWS KMS (SSE-KMS) im Amazon Simple Storage Service-Benutzerhandbuch.
2. Stellen Sie sicher, dass die Objekte im Bucket mit demselben KMS-Schlüssel verschlüsselt sind. Sie können ein beliebiges Objekt aus dem Amazon-S3-Bucket auswählen und in den serverseitigen Verschlüsselungseinstellungen den KMS-Schlüssel-ARN überprüfen.
3. Bearbeiten Sie die Amazon S3-Bucket-Richtlinie, um die CloudFront Erlaubnis zum Aufrufen des `GetObject` API-Vorgangs aus dem Amazon S3-Bucket zu gewähren. Eine Beispielrichtlinie für einen Amazon-S3-Bucket, die die Ursprungszugriffssteuerung verwendet, finden Sie unter Erteilen Sie die CloudFront Erlaubnis, auf den S3-Bucket zuzugreifen.
4. Bearbeiten Sie die KMS-Schlüsselrichtlinie, um die CloudFront Berechtigung zum Ausführen der Aktionen für `EncryptDecrypt`, und zu gewährenGenerateDataKey*. Geben Sie ein `Condition` Element an, sodass nur die angegebene Distribution die aufgelisteten Aktionen ausführen kann, damit nur die angegebene CloudFront Distribution die aufgelisteten Aktionen ausführen kann. Sie können die Richtlinie an Ihre bestehende AWS KMS Richtlinie anpassen. Ein Beispiel für eine KMS-Schlüsselrichtlinie finden Sie unter SSE-KMS.

Wenn Sie die Ursprungszugriffsidentität (OAI) anstelle von OAC verwenden, unterscheiden sich die Berechtigungen für den Amazon-S3-Bucket geringfügig, da Sie die Berechtigung für eine Identität statt für den AWS-Service erteilen. Weitere Informationen finden Sie unter Erteilen der Berechtigung zum Lesen von Dateien im Amazon-S3-Bucket für eine Ursprungszugriffsidentität.

Wenn Ihre Dateien in Ihrer Distribution immer noch nicht angezeigt werden, finden Sie weitere Informationen unter Ich kann die Dateien in meiner Distribution nicht anzeigen..

Sie verwenden ungültige Anmeldeinformationen oder verfügen nicht über ausreichende Berechtigungen

Die Fehlermeldung „Zugriff verweigert“ kann angezeigt werden, wenn Sie falsche oder abgelaufene AWS SCT Anmeldeinformationen (Zugriffsschlüssel und geheimer Schlüssel) verwenden oder Ihrer IAM-Rolle oder Ihrem IAM-Benutzer die erforderliche Berechtigung zum Ausführen einer Aktion an einer CloudFront Ressource fehlt. Weitere Informationen zur Fehlermeldung „Zugriff verweigert“ finden Sie im IAM-Benutzerhandbuch unter Problembehandlung bei der Fehlermeldung „Zugriff verweigert“.

Informationen zur Funktionsweise von IAM finden Sie unter CloudFront. Identitäts- und Zugriffsmanagement für Amazon CloudFront

CloudFront gibt eine zurück InvalidViewerCertificate Fehler beim Versuch, einen alternativen Domainnamen hinzuzufügen

Wenn beim Versuch, Ihrer Distribution einen alternativen Domainnamen (CNAME) hinzuzufügen, ein `InvalidViewerCertificate` Fehler CloudFront zurückgegeben wird, lesen Sie die folgenden Informationen, um das Problem zu beheben. Dieser Fehler kann darauf hinweisen, dass Sie eines der folgenden Probleme lösen müssen, bevor Sie den alternativen Domänennamen erfolgreich hinzufügen können.

Die folgenden Fehler sind in der Reihenfolge aufgeführt, CloudFront in der nach der Autorisierung zum Hinzufügen eines alternativen Domainnamens gesucht wird. Dies kann Ihnen bei der Behebung von Problemen helfen, da Sie anhand des CloudFront zurückgegebenen Fehlers erkennen können, welche Verifizierungsprüfungen erfolgreich abgeschlossen wurden.

Ihrer Distribution ist kein Zertifikat angefügt.

> Um einen alternativen Domänennamen (CNAME) hinzuzufügen, müssen Sie Ihrer Distribution ein vertrauenswürdiges, gültiges Zertifikat hinzufügen. Überprüfen Sie die Anforderungen, erwerben Sie ein gültiges Zertifikat, das diese Anforderungen erfüllt, und wiederholen Sie den Vorgang. Weitere Informationen finden Sie unter Voraussetzungen für die Verwendung von alternativen Domänennamen.

Es gibt zu viele Zertifikate in der Zertifikatskette für das von Ihnen angefügte Zertifikat.

> Die Zahl der Zertifikate in einer Zertifikatskette ist auf fünf beschränkt. Reduzieren Sie die Anzahl der Zertifikate in der Kette und wiederholen Sie den Vorgang.

Die Zertifikatskette enthält mindestens ein Zertifikat, das für das aktuelle Datum nicht gültig ist.

Die Zertifikatskette für ein von Ihnen hinzugefügtes Zertifikat enthält mindestens ein ungültiges Zertifikat, das entweder noch nicht gültig oder abgelaufen ist. Überprüfen Sie die Felder Not Valid Before (Nicht gültig vor) und Not Valid After (Nicht gültig nach) in den Zertifikaten in Ihrer Zertifikatskette, um sicherzustellen, dass alle Zertifikate basierend auf den von Ihnen aufgelisteten Daten gültig sind.

Das von Ihnen angefügte Zertifikat wurde nicht von einer vertrauenswürdigen Zertifizierungsstelle (Certificate Authority, CA) signiert.

Das Zertifikat, das Sie CloudFront zur Überprüfung eines alternativen Domainnamens beifügen, kann kein selbstsigniertes Zertifikat sein. Es muss von einer vertrauenswürdigen CA signiert worden sein. Weitere Informationen finden Sie unter Voraussetzungen für die Verwendung von alternativen Domänennamen.

Das von Ihnen angefügte Zertifikat ist nicht korrekt formatiert.

Das Format des Domänennamens und der IP-Adresse im Zertifikat und das Format des Zertifikats selbst müssen den Standards für Zertifikate folgen.

Es ist ein CloudFront interner Fehler aufgetreten.

CloudFront wurde durch ein internes Problem blockiert und konnte keine Validierungsprüfungen für Zertifikate durchführen. CloudFront Gibt in diesem Szenario einen HTTP 500-Statuscode zurück und weist darauf hin, dass beim Anhängen des Zertifikats ein internes CloudFront Problem vorliegt. Warten Sie einige Minuten und wiederholen Sie dann den Vorgang, um dem Zertifikat den alternativen Domänennamen hinzuzufügen.

Das von Ihnen angefügte Zertifikat deckt den alternativen Domänennamen nicht ab, den Sie hinzufügen möchten.

CloudFront Erfordert, dass Sie für jeden alternativen Domainnamen, den Sie hinzufügen, ein gültiges SSL/TLS Zertifikat einer vertrauenswürdigen Zertifizierungsstelle (CA) beifügen, das den Domainnamen abdeckt, um Ihre Nutzungsberechtigung zu bestätigen. Bitte aktualisieren Sie Ihr Zertifikat, sodass es einen Domänennamen enthält, der den CNAME-Wert abdeckt, den Sie hinzufügen möchten. Weitere Informationen und Beispiele zur Verwendung von Domänennamen mit Platzhalterzeichen finden Sie unter Voraussetzungen für die Verwendung von alternativen Domänennamen.

CloudFront gibt einen falsch konfigurierten DNS-Eintragsfehler zurück, wenn ich versuche, einen neuen CNAME hinzuzufügen

Wenn Sie einen vorhandenen Platzhalter-DNS-Eintrag haben, der auf eine CloudFront Distribution verweist, und Sie versuchen, einen neuen CNAME mit einem spezifischeren Namen hinzuzufügen, tritt möglicherweise der folgende Fehler auf:

```
One or more aliases specified for the distribution includes an incorrectly configured
 DNS record that points to another CloudFront distribution. You must update the DNS
 record to correct the problem.
```

Dieser Fehler tritt auf, weil das DNS CloudFront anhand des CNAME abgefragt wird und der Platzhalter-DNS-Eintrag in eine andere Distribution aufgelöst wird.

Um dieses Problem zu beheben, erstellen Sie zunächst eine weitere Distribution und dann einen DNS-Eintrag, der auf die neue Distribution verweist. Abschließend fügen Sie den spezifischeren CNAME hinzu. Weitere Information zum Hinzufügen von CNAMEs finden Sie unter Hinzufügen eines alternativen Domainnamens.

Ich kann die Dateien in meiner Distribution nicht anzeigen.

Wenn Sie die Dateien in Ihrer CloudFront Distribution nicht anzeigen können, finden Sie in den folgenden Themen einige gängige Lösungen.

Haben Sie sich für beide CloudFront und Amazon S3 angemeldet?

Um Amazon CloudFront mit einem Amazon S3-Ursprung zu verwenden, müssen Sie sich für beide CloudFront und Amazon S3 separat registrieren. Weitere Informationen zur Registrierung für CloudFront und Amazon S3 finden Sie unterRichten Sie Ihre ein AWS-Konto.

Sind Ihre Amazon-S3-Bucket- und Objektberechtigungen korrekt festgelegt?

Wenn Sie es CloudFront mit einem Amazon S3-Ursprung verwenden, werden die Originalversionen Ihrer Inhalte in einem S3-Bucket gespeichert. Um den Inhalt Ihren Zuschauern zur Verfügung zu stellen, empfehlen wir Ihnen, CloudFront Origin Access Control (OAC) zu verwenden, um den Zugriff auf Amazon S3-Buckets zu sichern. Das bedeutet, dass Ihr S3-Bucket nur über CloudFront erreichbar ist. OAC steuert den Zuschauerzugriff und die sichere Übertragung mithilfe CloudFront von. Weitere Informationen zu OAC finden Sie unter the section called “Beschränken des Zugriffs auf einen Amazon-S3-Ursprung”.

Weitere Informationen zur Verwaltung Ihres Bucket-Zugriffs finden Sie unter Blockieren des öffentlichen Zugriffs auf Ihren Amazon-S3-Speicher im Amazon-S3-Benutzerhandbuch.

Objekteigenschaften und Bucket-Eigenschaften existieren unabhängig voneinander. Sie müssen jedem Objekt in Amazon S3 explizit Berechtigungen zuweisen. Objekte erben ihre Eigenschaften nicht von Buckets; Objekteigenschaften müssen unabhängig vom jeweiligen Bucket festgelegt werden.

Ist Ihr alternativer Domänenname (CNAME) richtig konfiguriert?

Wenn Sie bereits über einen CNAME-Datensatz für den Namen Ihrer Domain verfügen, aktualisieren Sie den vorhandenen Datensatz oder ersetzen Sie ihn mit einem neuen, der auf den Domain-Namen Ihrer Distribution verweist.

Stellen Sie darüber hinaus sicher, dass der CNAME-Datensatz auf den Domänennamen Ihrer Distribution, und nicht auf das Amazon S3-Bucket zeigt. Sie können überprüfen, ob der CNAME-Datensatz in Ihrem DNS-System auf den Domainnamen Ihrer Distribution zeigt. Verwenden Sie dafür ein DNS-Tool wie dig.

Das folgende Beispiel zeigt eine Dig-Anfrage für eine Domain mit dem Namen `images.example.com` sowie den relevanten Teil der Antwort. Unter `ANSWER SECTION` finden Sie eine Zeile, die den `CNAME`-Wert enthält. Der CNAME-Eintrag für Ihren Domainnamen ist korrekt eingerichtet, wenn der Wert auf der rechten Seite von CNAME der Domainname Ihrer CloudFront Distribution ist. Wenn das Bucket des Amazon S3-Ursprungsservers oder ein anderer Domainname angegeben wird, ist der CNAME-Datensatz nicht korrekt eingerichtet.

```
[prompt]> dig images.example.com

; <<> DiG 9.3.3rc2 <<> images.example.com
;; global options: printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 15917
;; flags: qr rd ra; QUERY: 1, ANSWER: 9, AUTHORITY: 2, ADDITIONAL: 0
;; QUESTION SECTION:
;images.example.com.   IN  A
;; ANSWER SECTION:
images.example.com. 10800 IN CNAME d111111abcdef8.cloudfront.net.
...
...
```

Weitere Informationen zu CNAMEs finden Sie unter Verwenden von benutzerdefinierten URLs durch Hinzufügen von alternativen Domainnamen (CNAMEs).

Verweisen Sie auf die richtige URL für Ihre Distribution? CloudFront

Stellen Sie sicher, dass die URL, auf die Sie verweisen, den Domainnamen (oder CNAME) Ihrer CloudFront Distribution verwendet, nicht Ihren Amazon S3-Bucket oder Ihren benutzerdefinierten Ursprung.

Benötigen Sie Hilfe bei der Fehlerbehebung in Zusammenhang mit einem benutzerdefinierten Ursprungsserver?

Wenn Sie bei der Fehlerbehebung AWS für einen benutzerdefinierten Ursprung helfen müssen, müssen AWS Sie wahrscheinlich die `X-Amz-Cf-Id` Header-Einträge Ihrer Anfragen überprüfen. Wenn Sie diese Einträge nicht bereits protokollieren, sollten Sie das für die Zukunft in Erwägung ziehen. Weitere Informationen finden Sie unter the section called “Verwenden von Amazon EC2 (oder eines anderen benutzerdefinierten Ursprungs)”. Weitere Hilfe erhalten Sie im AWS -Supportcenter.

Fehlermeldung: Certificate: < certificate-id > wird verwendet von CloudFront

Problem: Sie versuchen, ein SSL/TLS Zertifikat aus dem IAM-Zertifikatsspeicher zu löschen, und Sie erhalten die Meldung „Zertifikat: <certificate-id>wird verwendet von“. CloudFront

Lösung: Jede CloudFront Distribution muss entweder mit dem CloudFront Standardzertifikat oder mit einem benutzerdefinierten SSL/TLS Zertifikat verknüpft sein. Bevor Sie ein SSL/TLS Zertifikat löschen können, müssen Sie das Zertifikat entweder rotieren (das aktuelle benutzerdefinierte SSL/TLS Zertifikat durch ein anderes benutzerdefiniertes SSL/TLS Zertifikat ersetzen) oder von der Verwendung eines benutzerdefinierten SSL/TLS Zertifikats zur Verwendung des CloudFront Standardzertifikats zurückkehren. Um dies zu beheben, führen Sie die Schritte in einem der folgenden Verfahren aus:

- SSL/TLS Zertifikate rotieren
- Kehren Sie von einem benutzerdefinierten SSL/TLS Zertifikat zum Standardzertifikat zurück CloudFront

Problembehandlung bei Statuscodes für die Fehlerantwort in CloudFront

Wenn CloudFront Sie ein Objekt von Ihrem Ursprung anfordern und der Absender einen HTTP 4xx- oder 5xx-Statuscode zurückgibt, liegt ein Problem mit der Kommunikation zwischen CloudFront und Ihrem Absender vor.

Dieses Thema enthält auch Schritte zur Fehlerbehebung für diese Statuscodes bei der Verwendung von Lambda @Edge oder Functions. CloudFront

Die folgenden Themen enthalten detaillierte Erläuterungen zu den möglichen Ursachen für diese Fehlerantworten und bieten schrittweise Anleitungen zur Diagnose und Lösung der zugrunde liegenden Probleme.

Themen

- HTTP 400-Statuscode (Bad Request)
- HTTP-Statuscode 401 (Nicht autorisiert)
- HTTP-Statuscode 403 (Ungültige Methode)
- HTTP-Statuscode 403 (Genehmigung verweigert)
- HTTP-Statuscode 404 (Nicht gefunden)
- HTTP-Statuscode 412 (Vorbedingung fehlgeschlagen)
- HTTP-Statuscode 500 (Interner Serverfehler)
- HTTP 502-Statuscode (Bad Gateway)
- HTTP 503-Statuscode (Service nicht verfügbar)
- HTTP 504-Statuscode (Gateway Timeout)

HTTP 400-Statuscode (Bad Request)

CloudFront gibt eine ungültige 400-Anforderung zurück, wenn der Client einige ungültige Daten in der Anfrage sendet, z. B. fehlende oder falsche Inhalte in der Nutzlast oder in den Parametern. Es kann sich auch um einen generischen Client-Fehler handeln.

Amazon-S3-Ursprung gibt einen 400-Fehler zurück

Wenn Sie für Ihre CloudFront Distribution einen Amazon S3-Ursprung verwenden, sendet Ihre Distribution möglicherweise Fehlerantworten mit dem HTTP-Statuscode 400 Bad Request und einer Meldung ähnlich der folgenden:

Der Autorisierungs-Header ist falsch formatiert; die Region '*<AWS Region>*' ist falsch; es wird " *<AWS Region>* erwartet

Beispiel:

Der Autorisierungs-Header ist fehlerhaft; die Region "us-east-1" ist falsch; erwartet wird "us-west-2"

Dieses Problem kann im folgenden Szenario auftreten:

1. Der Ursprung Ihrer CloudFront Distribution ist ein Amazon S3-Bucket.
2. Sie haben den S3-Bucket von einer AWS Region in eine andere verschoben. Das heißt, Sie haben den S3-Bucket gelöscht und später einen neuen Bucket mit demselben Bucket-Namen erstellt, jedoch in einer anderen AWS Region als der, in der sich der ursprüngliche S3-Bucket befand.

Um diesen Fehler zu beheben, aktualisieren Sie Ihre CloudFront Distribution so, dass sie den S3-Bucket in der aktuellen AWS Region des Buckets findet.

Um Ihre CloudFront Distribution zu aktualisieren

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home.
2. Wählen Sie die Distribution aus, die diesen Fehler verursacht.
3. Wählen Sie Ursprünge und Ursprungsgruppen aus.
4. Suchen Sie den Ursprung für den S3-Bucket, den Sie verschoben haben. Aktivieren Sie das Kontrollkästchen neben diesem Ursprung und wählen Sie dann Bearbeiten aus.
5. Wählen Sie Yes, Edit aus. Sie müssen keine Einstellungen ändern, bevor Sie Yes, Edit (Ja, Bearbeiten) auswählen.

Wenn Sie diese Schritte abgeschlossen haben, wird Ihre CloudFront Distribution erneut bereitgestellt. Während die Distribution bereitgestellt wird, wird in der Spalte Letzte Änderung der Status Bereitgestellt angezeigt. Einige Zeit nach Abschluss der Bereitstellung sollten Sie die Fehlerantwort `AuthorizationHeaderMalformed` nicht mehr erhalten.

Der als Ursprung verwendete Application Load Balancer gibt einen 400-Fehler zurück

Wenn Sie für Ihre CloudFront Distribution einen Application Load Balancer-Ursprung verwenden, kann ein 400-Fehler unter anderem folgende Ursachen haben:

- Der Client hat eine falsch formatierte Anforderung gesendet, die die HTTP-Spezifikation nicht erfüllt.
- Der Anforderungsheader überschreitet 16 KB pro Anforderungszeile, 16 KB pro einzelnem Header oder 64 KB für den gesamtem Anforderungsheader.
- Der Client hat die Verbindung beendet, bevor er den vollständigen Anfragetext gesendet hat.

HTTP-Statuscode 401 (Nicht autorisiert)

Der Statuscode 401 „Nicht autorisiert“ gibt an, dass die Client-Anforderung nicht abgeschlossen wurde, da sie keine gültigen Authentifizierungsdaten für die angeforderte Ressource enthält. Dieser Statuscode wird mit dem HTTP-Antwortheader `WWW-Authenticate` gesendet, der Informationen darüber enthält, wie der Client die Ressource erneut anfordern kann, nachdem er den Benutzer zur Eingabe von Authentifizierungsdaten aufgefordert hat. Weitere Informationen finden Sie unter 401 Nicht autorisiert.

Wenn Ihr Absender erwartet CloudFront, dass ein `Authorization` Header die Anfragen authentifiziert, CloudFront muss er den `Authorization` Header an den Ursprung weiterleiten, um den Fehler 401 Unauthorized zu vermeiden. Wenn Sie CloudFront eine Viewer-Anfrage an Ihren Ursprung weiterleiten, CloudFront werden standardmäßig einige Viewer-Header entfernt, einschließlich des Headers. `Authorization` Um sicherzustellen, dass Ihr Ursprung immer den `Authorization`-Header in Ursprungsanforderungen erhält, haben Sie folgende Möglichkeiten:

- Fügen Sie den `Authorization`-Header mithilfe einer Cache-Richtlinie zum Cache-Schlüssel hinzu. Alle Header im Cache-Schlüssel werden automatisch in Ursprungsanforderungen eingeschlossen. Weitere Informationen finden Sie unter Steuern des Cache-Schlüssels mit einer Richtlinie.
- Füge den `Authorization` Header einzeln in einer Quell-Anforderungsrichtlinie hinzu. Weitere Informationen finden Sie unter Steuern von Ursprungsanforderungen anhand einer Richtlinie.
- Verwenden Sie eine Quellanforderungsrichtlinie, die alle Viewer-Header an den Ursprung weiterleitet. CloudFront bietet eine verwaltete Quellanforderungsrichtlinie für diesen Anwendungsfall mit dem Namen. Managed-AllViewer Weitere Informationen finden Sie unter Verwenden verwalteter Ursprungsanforderungsrichtlinien.

> **Important**
>
> Wenn Sie den `Authorization` Header an Ihren Ursprung weiterleiten, ohne ihn in den Cache-Schlüssel aufzunehmen, stellen Sie sicher, dass Ihr Ursprung bei der Zugriffskontrolle für zwischengespeicherte Inhalte nicht auf den `Authorization` Header angewiesen ist. Wenn der `Authorization` Header nicht Teil des Cache-Schlüssels ist, CloudFront kann er sowohl autorisierten als auch nicht autorisierten Zuschauern dieselbe zwischengespeicherte Antwort senden. Nehmen Sie den `Authorization` Header entweder mithilfe einer Cache-Richtlinie in den Cache-Schlüssel auf oder deaktivieren Sie das Caching vollständig für Ursprünge, für die eine Autorisierungsverarbeitung auf der Originalseite erforderlich ist.

Weitere Informationen findest du unter Wie kann ich konfigurieren, dass der Authorization-Header CloudFront an den Ursprung weitergeleitet wird?

HTTP-Statuscode 403 (Ungültige Methode)

CloudFront gibt einen 403-Fehler (Ungültige Methode) zurück, wenn Sie versuchen, eine HTTP-Methode zu verwenden, die Sie in der CloudFront Distribution nicht angegeben haben. Sie können für Ihre Distribution eine der folgenden Optionen auswählen:

- CloudFront leitet nur `GET` und `HEAD` Anfragen weiter.
- CloudFront leitet nur weiter `GET` und `HEAD` `OPTIONS` Anfragen.
- CloudFront leitet`GET`,,`HEAD`, `OPTIONS` `PUT` `PATCHPOST`, und `DELETE` Anfragen weiter. (Wenn Sie diese Option auswählen, müssen Sie möglicherweise den Zugriff auf Ihren Amazon-S3-Bucket bzw. auf Ihren benutzerdefinierten Ursprung beschränken, um zu verhindern, dass Benutzer Operationen ausführen, für die sie nicht befugt sind. Möglicherweise möchten Sie nicht, dass Benutzer über Berechtigungen zum Löschen von Objekten von Ihrem Ursprung verfügen.

HTTP-Statuscode 403 (Genehmigung verweigert)

Der HTTP-Fehler 403 bedeutet, dass der Client nicht berechtigt ist, auf die angeforderte Ressource zuzugreifen. Der Client versteht die Anforderung, kann den Viewer-Zugriff jedoch nicht autorisieren. Die folgenden Ursachen treten häufig auf, wenn dieser Statuscode CloudFront zurückgegeben wird:

Themen

- Alternativer CNAME ist falsch konfiguriert

- AWS WAF wird bei der CloudFront Verteilung oder am Ursprung konfiguriert
- Benutzerdefinierter Ursprung gibt einen 403-Fehler zurück
- Amazon-S3-Ursprung gibt einen 403-Fehler zurück
- Geografische Einschränkungen geben einen 403-Fehler zurück
- Die Konfiguration einer signierten URL oder eines signierten Cookies gibt einen 403-Fehler zurück
- Gestapelte Distributionen verursachen einen 403-Fehler

Alternativer CNAME ist falsch konfiguriert

Stellen Sie sicher, dass Sie den richtigen CNAME für die Distribution angegeben haben. Um einen alternativen CNAME anstelle der CloudFront Standard-URL zu verwenden:

1. Erstellen Sie einen CNAME-Eintrag in Ihrem DNS, um den CNAME auf die Verteilungs-URL zu CloudFront verweisen.
2. Fügen Sie den CNAME zu Ihrer CloudFront Distributionskonfiguration hinzu.

Wenn Sie den DNS-Eintrag erstellen, aber den CNAME nicht in Ihrer CloudFront Distributionskonfiguration hinzufügen, gibt die Anfrage einen 403-Fehler zurück. Weitere Informationen zum Konfigurieren eines benutzerdefinierten CNAME finden Sie unter Verwenden von benutzerdefinierten URLs durch Hinzufügen von alternativen Domainnamen (CNAMEs).

AWS WAF wird bei der CloudFront Verteilung oder am Ursprung konfiguriert

Wenn zwischen dem Client und AWS WAF liegt CloudFront, CloudFront kann nicht zwischen einem 403-Fehlercode, der von Ihrem Ursprung zurückgegeben wird, und einem 403-Fehlercode, der zurückgegeben wird, AWS WAF wenn eine Anfrage blockiert wird, unterscheiden.

Um die Quelle des 403-Statuscodes zu finden, überprüfen Sie Ihre ACL-Regel (AWS WAF Web Access Control List) auf eine blockierte Anfrage. Weitere Informationen finden Sie unter den folgenden Themen:

- AWS WAF Web-Zugriffskontrolllisten (Web-ACLs)
- Testen und Optimieren Ihrer AWS WAF -Schutzmaßnahmen

Benutzerdefinierter Ursprung gibt einen 403-Fehler zurück

Wenn Sie einen benutzerdefinierten Ursprung verwenden, wird möglicherweise ein 403-Fehler angezeigt, wenn am Ursprung eine benutzerdefinierte Firewall-Konfiguration vorhanden ist. Um diesen Fehler zu beheben, richten Sie die Anforderung direkt an den Ursprung. Wenn Sie den Fehler ohne replizieren können CloudFront, verursacht der Ursprung den 403-Fehler.

Wenn der benutzerdefinierte Ursprung den Fehler verursacht, überprüfen Sie die Ursprungsprotokolle, um die Ursache des Fehlers herauszufinden. Weitere Informationen zur Behebung von Fehlern finden Sie in den folgenden Themen:

- Wie behebe ich den HTTP-Fehler 403 von API Gateway?
- Wie behebe ich den HTTP-Fehler 403 „Verboten“ im Application Load Balancer?

Amazon-S3-Ursprung gibt einen 403-Fehler zurück

Der 403-Fehler wird möglicherweise aus folgenden Gründen angezeigt:

- CloudFront hat keinen Zugriff auf den Amazon S3-Bucket. Dies kann passieren, wenn die Ursprungszugriffsidentität (OAI) oder Ursprungszugriffskontrolle (OAC) für Ihre Distribution nicht aktiviert ist und der Bucket privat ist.
- Der angegebene Pfad in der angeforderten URL ist nicht korrekt.
- Das angeforderte Objekt ist nicht vorhanden.
- Der Host-Header wurde mit dem REST-API-Endpunkt weitergeleitet. Weitere Informationen finden Sie unter Angabe des Buckets mit dem HTTP-Host-Header im Benutzerhandbuch für Amazon Simple Storage Service.
- Sie haben benutzerdefinierte Fehlerseiten konfiguriert. Weitere Informationen finden Sie unter Wie CloudFront verarbeitet Fehler, wenn Sie benutzerdefinierte Fehlerseiten konfiguriert haben.

Geografische Einschränkungen geben einen 403-Fehler zurück

Wenn Sie geografische Einschränkungen (auch bekannt als Geoblocking) aktiviert haben, um zu verhindern, dass Benutzer an bestimmten geografischen Standorten auf Inhalte zugreifen, die Sie über eine CloudFront Distribution verteilen, erhalten blockierte Benutzer einen 403-Fehler.

Weitere Informationen finden Sie unter Einschränken der geografischen Distribution Ihrer Inhalte.

Die Konfiguration einer signierten URL oder eines signierten Cookies gibt einen 403-Fehler zurück

Wenn Sie für die Verhaltenskonfiguration Ihrer Distribution die Option Viewer-Zugriff einschränken aktiviert haben, führen Anforderungen, die keine signierten Cookies oder signierten URLs verwenden, zu einem 403-Fehler. Weitere Informationen finden Sie unter den folgenden Themen:

- Bereitstellen privater Inhalte mit signierten URLs und signierten Cookies
- Wie behebe ich Probleme im Zusammenhang mit einer signierten URL oder angemeldeten Cookies? CloudFront

Gestapelte Distributionen verursachen einen 403-Fehler

Wenn Sie zwei oder mehr Distributionen innerhalb einer Kette von Anfragen an den ursprünglichen Endpunkt haben, wird ein 403-Fehler CloudFront zurückgegeben. Wir empfehlen nicht, eine Distribution vor einer anderen zu platzieren.

HTTP-Statuscode 404 (Nicht gefunden)

CloudFront gibt einen 404-Fehler (Nicht gefunden) zurück, wenn der Client versucht, auf eine Ressource zuzugreifen, die nicht existiert. Wenn Sie diesen Fehler in Ihrer CloudFront Distribution erhalten, hat das häufig folgende Ursachen:

- Die Ressource ist nicht vorhanden.
- Die URL ist nicht korrekt.
- Der benutzerdefinierte Ursprung gibt einen 404-Fehler zurück.
- Benutzerdefinierte Fehlerseiten geben einen 404-Fehler zurück. (Jeder Fehlercode kann in 404 übersetzt werden.) Weitere Informationen finden Sie unter Wie CloudFront verarbeitet Fehler, wenn Sie benutzerdefinierte Fehlerseiten konfiguriert haben.
- Die benutzerdefinierte Fehlerseite wurde versehentlich gelöscht, was zu einem 404-Fehler führt, da die Anforderung nach der gelöschten benutzerdefinierten Fehlerseite sucht. Weitere Informationen finden Sie unter Wie CloudFront verarbeitet Fehler, wenn Sie keine benutzerdefinierten Fehlerseiten konfiguriert haben.
- Falscher Ursprungspfad. Wenn der Ursprungspfad ausgefüllt ist, wird sein Wert an den Pfad jeder Anforderung vom Browser angehängt, bevor die Anforderung an den Ursprung weitergeleitet wird. Weitere Informationen finden Sie unter Ursprungspfad.

HTTP-Statuscode 412 (Vorbedingung fehlgeschlagen)

CloudFront gibt den Fehlercode 412 (Vorbedingung fehlgeschlagen) zurück, wenn der Zugriff auf die Zielressource verweigert wurde. In einigen Fällen ist ein Server so konfiguriert, dass er Anforderungen erst akzeptiert, wenn bestimmte Bedingungen erfüllt sind. Wenn eine der angegebenen Bedingungen nicht erfüllt ist, erlaubt der Server dem Client nicht, auf die angegebene Ressource zuzugreifen. Stattdessen antwortet der Server mit dem Fehlercode 412.

Zu den häufigsten Ursachen für einen 412-Fehler in CloudFront gehören:

- Bedingte Anforderungen für andere Methoden als `GET` oder `HEAD`, wenn die durch die Header `If-None-Match` oder `If-Unmodified-Since` definierte Bedingung nicht erfüllt ist. In diesem Fall kann die Anforderung, normalerweise ein Upload oder die Änderung einer Ressource, nicht gestellt werden.
- Eine Bedingung in einem oder mehreren der Anforderungsfelder des CloudFront UpdateDistribution API-Vorgangs wird als falsch ausgewertet.

HTTP-Statuscode 500 (Interner Serverfehler)

Der HTTP-Statuscode 500 (Interner Serverfehler) gibt an, dass der Server auf einen unerwarteten Zustand gestoßen ist, der ihn daran gehindert hat, die Anforderung zu bearbeiten. Im Folgenden sind einige häufige Ursachen für 500-Fehler bei Amazon aufgeführt CloudFront.

Themen

- Der Origin-Server gibt den Fehler 500 zurück an CloudFront

Der Origin-Server gibt den Fehler 500 zurück an CloudFront

Ihr Ursprungsserver gibt möglicherweise einen 500-Fehler an zurück CloudFront. Weitere Informationen zur Fehlerbehebung erhalten Sie in folgenden Themen:

- Wenn Amazon S3 einen 500-Fehler zurückgibt, finden Sie weitere Informationen unter Wie behebe ich den HTTP-Fehler 500 oder 503 von Amazon S3?
- Wenn API Gateway einen 500-Fehler zurückgibt, finden Sie weitere Informationen unter Wie behebe ich 5xx-Fehler für die REST-API in API Gateway?
- Wenn Elastic Load Balancing einen 500-Fehler zurückgibt, finden Sie weitere Informationen unter HTTP 500: Interner Serverfehler im Benutzerhandbuch für Application Load Balancer.

Wenn die obige Liste den 500-Fehler nicht behebt, liegt das Problem möglicherweise daran, dass ein CloudFront Point of Presence einen internen Serverfehler zurückgibt. Sie können sich an Support wenden, um Unterstützung zu erhalten.

HTTP 502-Statuscode (Bad Gateway)

CloudFront gibt einen HTTP 502-Statuscode (Bad Gateway) zurück, wenn das angeforderte Objekt CloudFront nicht bereitgestellt werden konnte, weil es keine Verbindung zum Ursprungsserver herstellen konnte.

Wenn Sie Lambda@Edge verwenden, ist das Problem möglicherweise ein Lambda-Validierungsfehler. Wenn Sie einen HTTP 502-Fehler mit dem `NonS3OriginDnsError` Fehlercode erhalten, liegt wahrscheinlich ein DNS-Konfigurationsproblem vor, das die Verbindung zum Ursprung CloudFront verhindert.

Themen

- SSL/TLS Fehler bei der Aushandlung zwischen CloudFront und einem benutzerdefinierten Ursprungsserver
- Origin reagiert nicht mit supported ciphers/protocols
- SSL/TLS Das Ursprungszertifikat ist abgelaufen, ungültig, selbstsigniert oder die Zertifikatskette ist in der falschen Reihenfolge
- Ursprungsserver ist über die eingestellten Ports in den Ursprungseinstellungen nicht erreichbar
- Lambda-Validierungsfehler
- CloudFront Fehler bei der Validierung der Funktion
- DNS-Fehler (NonS3OriginDnsError)
- 502-Fehler des als Ursprung verwendeten Application Load Balancer
- API Gateway Ursprung 502-Fehler

SSL/TLS Fehler bei der Aushandlung zwischen CloudFront und einem benutzerdefinierten Ursprungsserver

Wenn Sie einen benutzerdefinierten Ursprung verwenden, für den HTTPS zwischen CloudFront und Ihrem Absender erforderlich ist, können nicht übereinstimmende Domainnamen zu Fehlern führen. Das SSL/TLS Zertifikat für Ihren Ursprung muss einen Domainnamen enthalten, der entweder mit der Ursprungsdomain, die Sie für die CloudFront Verteilung angegeben haben, oder mit dem `Host` Header der Quellanfrage übereinstimmt.

Wenn die Domainnamen nicht übereinstimmen, schlägt der SSL/TLS Handshake fehl und CloudFront gibt den HTTP-Statuscode 502 (Bad Gateway) zurück und setzt den `X-Cache` Header auf`Error from cloudfront`.

Sie können bestimmen, ob die Domainnamen im Zertifikat der Ursprungsdomain in der Distribution oder dem `Host`-Header entsprechen, indem Sie eine Online-SSL-Prüfung vornehmen oder OpenSSL verwenden. Wenn die Domain-Namen nicht übereinstimmen, haben Sie zwei Optionen:

- Besorgen Sie sich ein neues SSL/TLS Zertifikat, das die entsprechenden Domainnamen enthält.

  Wenn Sie AWS Certificate Manager (ACM) verwenden, finden Sie unter Beantragen eines öffentlichen Zertifikats im AWS Certificate Manager Benutzerhandbuch Informationen zum Anfordern eines neuen Zertifikats.
- Ändern Sie die Distributionskonfiguration, sodass nicht CloudFront mehr versucht wird, SSL zu verwenden, um eine Verbindung zu Ihrem Ursprungsserver herzustellen.

Online-SSL-Prüfung

Sie finden ein Test-Tool für SSL-Verbindungen, indem Sie im Internet nach den Begriffen "online ssl checker" suchen. In der Regel geben Sie den Namen Ihrer Domain an, und das Tool gibt eine Vielzahl von Informationen zu Ihrem SSL/TLS Zertifikat zurück. Vergewissern Sie sich, dass das Zertifikat Ihren Domänennamen aus den Feldern Allgemeiner Name oder Alternative Antragstellernamen enthält.

OpenSSL

Um bei der Behebung von HTTP 502-Fehlern von zu helfen CloudFront, können Sie OpenSSL verwenden, um zu versuchen, eine SSL/TLS Verbindung zu Ihrem Ursprungsserver herzustellen. Wenn OpenSSL keine Verbindung herstellen kann, kann dies auf ein Problem mit der Konfiguration Ihres Ursprungsservers hinweisen. SSL/TLS Wenn OpenSSL eine Verbindung herstellen kann, gibt es Informationen über das Zertifikat des Ursprungsservers zurück, einschließlich des allgemeinen Namens (Feld `Subject CN`) des Zertifikats und des alternativen Antragstellernamens (Feld `Subject Alternative Name`).

Verwenden Sie den folgenden OpenSSL-Befehl, um die Verbindung zu Ihrem Ursprungsserver zu testen (*`origin domain`*ersetzen Sie ihn durch den Domainnamen Ihres Ursprungsservers, z. B. example.com):

```
openssl s_client -connect origin domain name:443
```

Wenn Folgendes zutrifft:

- Ihr Ursprungsserver unterstützt mehrere Domainnamen mit mehreren Zertifikaten SSL/TLS
- Ihre Distribution ist so konfiguriert, dass der `Host`-Header an den Ursprung weitergeleitet wird.

Fügen Sie dann die `-servername` Option wie im folgenden Beispiel zum OpenSSL-Befehl hinzu (*`CNAME`*ersetzen Sie sie durch den CNAME, der in Ihrer Distribution konfiguriert ist):

```
openssl s_client -connect origin domain name:443 -servername CNAME
```

Origin reagiert nicht mit supported ciphers/protocols

CloudFront stellt mithilfe von Chiffren und Protokollen eine Verbindung zu den Ursprungsservern her. Eine Liste der unterstützten Chiffren und Protokolle finden Sie unter. CloudFront the section called “Unterstützte Protokolle und Chiffren zwischen CloudFront und dem Ursprung” Wenn Ihr Ursprungsserver beim SSL/TLS Austausch nicht mit einer dieser Chiffren oder Protokolle antwortet, CloudFront schlägt die Verbindung fehl. Sie können überprüfen, ob Ihr Ursprungsserver diese Verschlüsselungsverfahren und Protokolle unterstützt, indem Sie ein Online-Tool wie SSL-Labs verwenden. Geben Sie den Domain-Namen Ihres Ursprungs-Servers in das Feld Hostname ein und wählen Sie anschließend OK aus. Überprüfen Sie die Felder Common names (Allgemeine Namen) und Alternative names (Alternative Namen) in dem Test; entspricht ihr Inhalt dem Domain-Namen Ihres Ursprungs-Servers? Nachdem der Test beendet ist, suchen Sie die Abschnitte Protocols und Cipher Suites in den Testergebnissen; welche Verschlüsselungsverfahren oder Protokolle werden von Ihrem Ursprungs-Server unterstützt? Vergleichen Sie die Produkte mit der Liste von the section called “Unterstützte Protokolle und Chiffren zwischen CloudFront und dem Ursprung”.

SSL/TLS Das Ursprungszertifikat ist abgelaufen, ungültig, selbstsigniert oder die Zertifikatskette ist in der falschen Reihenfolge

Wenn der Ursprungsserver Folgendes zurückgibt, trennt er CloudFront die TCP-Verbindung, gibt den HTTP-Statuscode 502 (Bad Gateway) zurück und setzt den `X-Cache` Header auf`Error from cloudfront`:

- Abgelaufenes Zertifikat
- Ungültiges Zertifikat
- Self-signed Zertifikat
- Zertifikatkette in der falschen Reihenfolge

> **Note**
>
> Wenn die gesamte Zertifikatskette, einschließlich des Zwischenzertifikats, nicht vorhanden ist, wird CloudFront die TCP-Verbindung unterbrochen.

Hinweise zur Installation eines SSL/TLS Zertifikats auf Ihrem benutzerdefinierten Ursprungsserver finden Sie unterthe section called “Erzwingen von HTTPS für einen benutzerdefinierten Ursprung”.

Ursprungsserver ist über die eingestellten Ports in den Ursprungseinstellungen nicht erreichbar

Wenn Sie in Ihrer CloudFront Distribution einen Ursprung erstellen, können Sie die Ports festlegen, über die eine CloudFront Verbindung zum Ursprung für den HTTP- und HTTPS-Verkehr hergestellt wird. In der Standardeinstellung sind dies TCP 80/443. Sie haben die Möglichkeit, diese Ports zu ändern. Wenn Ihr Ursprung den Verkehr auf diesen Ports aus irgendeinem Grund ablehnt oder wenn Ihr Backend-Server auf den Ports nicht reagiert, kann keine Verbindung hergestellt CloudFront werden.

Sie können diese Probleme zu beheben, indem Sie alle Firewalls überprüfen, die in Ihrer Infrastruktur ausgeführt werden, und sich vergewissern, dass sie die unterstützten IP-Bereiche nicht blockieren. Weitere Informationen finden Sie unter AWS -IP-Adressbereiche im Benutzerhandbuch für Amazon VPC. Überprüfen Sie außerdem, ob Ihr Webserver auf dem Ursprungsserver ausgeführt wird.

Lambda-Validierungsfehler

Wenn Sie Lambda@Edge verwenden, kann der HTTP-Statuscode 502 anzeigen, dass Ihre Lambda-Funktionsantwort falsch gebildet wurde oder ungültigen Inhalt enthielt. Weitere Informationen zur Behebung von Lambda@Edge-Fehlern finden Sie unter Testen und Debuggen von Lambda@Edge-Funktionen.

CloudFront Fehler bei der Validierung der Funktion

Wenn Sie CloudFront Funktionen verwenden, kann ein HTTP 502-Statuscode darauf hinweisen, dass die CloudFront Funktion versucht, einen schreibgeschützten Header hinzuzufügen, zu löschen oder zu ändern. Dieser Fehler tritt beim Testen nicht auf, sondern erst, nachdem Sie die Funktion bereitgestellt und die Anforderung ausgeführt haben. Um diesen Fehler zu beheben, überprüfen und aktualisieren Sie Ihre CloudFront Funktion. Weitere Informationen finden Sie unter Aktualisieren von Funktionen.

DNS-Fehler (**`NonS3OriginDnsError`**)

Ein HTTP 502-Fehler mit dem `NonS3OriginDnsError` Fehlercode weist darauf hin, dass ein DNS-Konfigurationsproblem vorliegt, das die Verbindung zum Ursprung CloudFront verhindert. Wenn Sie diesen Fehler von erhalten CloudFront, stellen Sie sicher, dass die DNS-Konfiguration des Ursprungs korrekt ist und funktioniert.

Wenn es eine Anfrage für ein Objekt CloudFront erhält, das abgelaufen ist oder sich nicht in seinem Cache befindet, sendet es eine Anfrage an den Ursprung, um das Objekt abzurufen. Um eine erfolgreiche Anfrage an den Ursprung zu stellen, CloudFront führt eine DNS-Auflösung für die Ursprungsdomäne durch. Wenn beim DNS-Dienst für Ihre Domain Probleme auftreten, CloudFront können Sie den Domainnamen nicht auflösen, um die IP-Adresse abzurufen, was zu einem HTTP 502-Fehler führt (`NonS3OriginDnsError`). Sie können dieses Problem beheben, indem Sie sich an Ihren DNS-Anbieter wenden. Wenn Sie Amazon Route 53 verwenden, finden Sie weitere Informationen unter Warum kann ich nicht auf meine Website zugreifen, die Route-53-DNS-Services verwendet?

Um dieses Problem weiter zu beheben, stellen Sie sicher, dass die autoritativen Nameserver der Root-Domain oder der Zone Apex Ihres Ursprungs (z. B.`example.com`) ordnungsgemäß funktionieren. Sie können die folgenden Befehle verwenden, um die Nameserver für Ihren Ursprungsserver-Apex zu finden, zum Beispiel mit einem Tool wie dig oder nslookup:

```
dig OriginAPEXDomainName NS +short
```

```
nslookup -query=NS OriginAPEXDomainName
```

Wenn Sie die Namen Ihrer Name-Server erhalten haben, verwenden Sie die folgenden Befehle, um den Domain-Namen Ihres Ursprungsservers von ihnen abzufragen; so können Sie sicherstellen, dass alle eine Antwort zurückgeben:

```
dig OriginDomainName @NameServer
```

```
nslookup OriginDomainName NameServer
```

> **Important**
>
> Stellen Sie sicher, dass Sie diese DNS-Problembehandlung auf einem Computer durchführen, der mit dem öffentlichen Internet verbunden ist. CloudFront löst die

Ursprungsdomäne mithilfe von öffentlichem DNS im Internet auf. Daher ist es wichtig, die Fehlerbehebung in einem ähnlichen Kontext durchzuführen.

Wenn der Ursprung eine Subdomäne ist, deren DNS-Berechtigung an einen anderen Nameserver als die Stammdomäne delegiert ist, stellen Sie sicher, dass die Datensätze für den Nameserver (NS) und Autoritätsursprung (SOA) für die Subdomäne korrekt konfiguriert sind. Sie können mit Befehlen, die den vorherigen Beispielen ähneln, nach diesen Datensätzen suchen.

Weitere Informationen zu DNS finden Sie unter DNS-Konzepte (Domain Name System) in der Dokumentation zu Amazon Route 53.

502-Fehler des als Ursprung verwendeten Application Load Balancer

Wenn Sie Application Load Balancer als Ursprung verwenden und einen 502-Fehler erhalten, finden Sie weitere Informationen unter Wie behebe ich den HTTP-Fehler 502 des Application Load Balancer?

API Gateway Ursprung 502-Fehler

Wenn Sie API Gateway verwenden und einen 502-Fehler erhalten, finden Sie weitere Informationen unter Wie behebe ich den HTTP-Fehler 502 von REST-APIs in API Gateway mit Lambda-Proxyintegration?

HTTP 503-Statuscode (Service nicht verfügbar)

Ein HTTP-Statuscode 503 (Service nicht verfügbar) zeigt in der Regel an, dass es auf dem Ursprungsserver ein Leistungsproblem gibt. In seltenen Fällen deutet dies darauf hin, dass eine Anforderung aufgrund von Ressourcenbeschränkungen an einem Edge-Standort CloudFront vorübergehend nicht bearbeitet werden kann.

Wenn Sie Lambda @Edge oder CloudFront Functions verwenden, könnte das Problem ein Ausführungsfehler oder ein Lambda @Edge -Limit-Überschreitungsfehler sein.

Wenn Sie Mutual Origin TLS (Origin mTLS) verwenden, kann das Problem auf wiederholte fehlgeschlagene Versuche zurückzuführen sein, eine Verbindung mit einem bestimmten Zertifikat herzustellen.

Themen

- Ursprungsserver verfügt nicht über ausreichend Kapazitäten für die vorliegende Anfragerate
- CloudFront hat den Fehler aufgrund von Ressourcenbeschränkungen am Edge-Standort verursacht
- CloudFront hat den Fehler für einen Ursprung verursacht, der mit gegenseitigem TLS konfiguriert ist
- Fehler bei der Ausführung von Lambda @Edge oder der CloudFront Funktion
- Lambda@Edge-Grenzwert überschritten

Ursprungsserver verfügt nicht über ausreichend Kapazitäten für die vorliegende Anfragerate

Wenn ein Ursprungsserver nicht verfügbar ist oder eingehende Anfragen nicht bearbeiten kann, gibt er einen HTTP 503-Statuscode zurück (Service nicht verfügbar). CloudFront leitet den Fehler dann an den Benutzer zurück. Sie lösen dieses Problem, indem Sie die folgenden Schritte ausführen:

- Wenn Sie Amazon S3 als Ihren Ursprungsserver verwenden:
  - Sie können 3.500 PUT/COPY/POST/DELETE oder 5.500 GET/HEAD Anfragen pro Sekunde pro partitioniertem Amazon S3-Präfix senden. Wenn Amazon S3 die Antwort 503 „Slow Down“ zurückgibt, deutet dies in der Regel auf eine zu hohe Anforderungsrate für ein bestimmtes Amazon-S3-Präfix hin.

    Da die Anforderungsraten pro Präfix in einem S3-Bucket gelten, sollten Objekte auf mehrere Präfixe verteilt werden. Mit steigender Anforderungsrate für die Präfixe skaliert Amazon S3 hoch, um Anforderungen für jedes der Präfixe separat zu bearbeiten. Infolgedessen ist die Gesamtanforderungsrate, die der Bucket verarbeitet, ein Vielfaches der Anzahl der Präfixe.
  - Weitere Informationen zu Amazon-S3-Berechtigungen finden Sie unter Optimieren der Amazon-S3-Leistung im Benutzerhandbuch zu Amazon Simple Storage Service.
- Wenn Sie Elastic Load Balancing als Ursprungsserver verwenden:
  - Stellen Sie sicher, dass Ihre Backend-Instances auf Zustandsprüfungen reagieren können.
  - Stellen Sie sicher, dass Ihr Load Balancer und Ihre Backend-Instances die Last bewältigen können.

  Weitere Informationen finden Sie unter:
  - Wie behebe ich 503-Fehler, die bei der Verwendung von Classic Load Balancer zurückgegeben werden?

  - Wie behebe ich 503-Fehler (Service nicht verfügbar) von meinem Application Load Balancer?
- Wenn Sie einen benutzerdefinierten Ursprung verwenden:
  - Prüfen Sie die Anwendungsprotokolle, um sicherzustellen, dass Ihr Ursprung über ausreichende Ressourcen wie Arbeitsspeicher, CPU und Festplattengröße verfügt.
  - Wenn Sie Amazon EC2 als Backend verwenden, stellen Sie sicher, dass der Instance-Typ über die Ressourcen verfügt, die für die Verarbeitung der eingehenden Anforderungen erforderlich sind. Weitere Informationen finden Sie unter Instance-Typen im Amazon EC2-Benutzerhandbuch.
- Wenn Sie API Gateway verwenden:
  - Dieser Fehler steht im Zusammenhang mit der Backend-Integration, wenn die API in Gateway API keine Antwort empfangen kann. Der Backend-Server ist möglicherweise:
    - über die Kapazität hinaus überlastet und kann keine neuen Client-Anforderungen verarbeiten;
    - gerade im Wartungszustand.
  - Um diesen Fehler zu beheben, schauen Sie sich die Protokolle Ihrer API-Gateway-Anwendung an, um festzustellen, ob ein Problem mit der Backend-Kapazität, der Integration oder etwas anderem vorliegt.

CloudFront hat den Fehler aufgrund von Ressourcenbeschränkungen am Edge-Standort verursacht

Dieser Fehler tritt in dem seltenen Fall auf, dass CloudFront Anfragen nicht an den nächstbesten verfügbaren Edge-Standort weitergeleitet werden können und daher eine Anfrage nicht bearbeitet werden kann. Dieser Fehler tritt häufig auf, wenn Sie in Ihrer CloudFront-Verteilung Belastungstests durchführen. Um dies zu vermeiden, befolgen Sie die the section called “Testen CloudFront”-Richtlinien zum Vermeiden des Fehlers 503 (Kapazität überschritten).

Wenn dieser Fehler in Ihrer Produktionsumgebung auftritt, wenden Sie sich an Support.

CloudFront hat den Fehler für einen Ursprung verursacht, der mit gegenseitigem TLS konfiguriert ist

Wenn Ihre Distribution Mutual TLS (mTLS) verwendet, um eine Verbindung zu einem benutzerdefinierten Ursprung herzustellen, wird CloudFront möglicherweise ein HTTP 503-Statuscode zurückgegeben, wenn keine Verbindungen zum Ursprung hergestellt werden können. Dies kann passieren, wenn der Ursprung die TLS-Handshakes wiederholt nicht abschließt

oder aufgrund von DNS- oder Verbindungsproblemen nicht erreichbar ist. Dieser Zustand ist vorübergehend — nimmt den normalen Verkehr CloudFront automatisch wieder auf, sobald der Ursprung wieder erreichbar ist.

So beheben Sie dieses Problem

- Vergewissern Sie sich, dass das richtige Client-Zertifikat mit Ihrer Distribution verknüpft ist und dass es nicht abgelaufen ist. Wenn Sie das Zertifikat kürzlich aktualisiert haben, warten Sie, bis die Änderung wirksam wird.
- Vergewissern Sie sich, dass der Ursprung erreichbar ist und dass die DNS-Einträge korrekt aufgelöst werden.

Wenn das Problem trotz der Konfiguration des richtigen Zertifikats in Ihrer Produktionsumgebung weiterhin besteht, wenden Sie sich an Support.

Fehler bei der Ausführung von Lambda @Edge oder der CloudFront Funktion

Wenn Sie Lambda @Edge oder CloudFront Functions verwenden, kann ein HTTP 503-Statuscode darauf hinweisen, dass Ihre Funktion einen Ausführungsfehler zurückgegeben hat.

Weitere Informationen zur Identifizierung und Behebung von Lambda@Edge-Fehlern finden Sie unter Testen und Debuggen von Lambda@Edge-Funktionen.

Weitere Hinweise zum Testen von CloudFront Funktionen finden Sie unterTestfunktionen.

Lambda@Edge-Grenzwert überschritten

Wenn Sie Lambda@Edge verwenden, kann der HTTP-Statuscode 503 anzeigen, dass Lambda einen Fehler zurückgegeben hat. Der Fehler kann durch eine der folgenden Ursachen bedingt sein.

- Die Anzahl der Funktionsausführungen hat eines der Kontingente überschritten, die Lambda zur Drosselung der Ausführung in einem festlegt AWS-Region (gleichzeitige Ausführungen oder Aufrufhäufigkeit).
- Die Funktion hat das Timeout-Kontingent für die Lambda-Funktion überschritten.

Weitere Informationen zu den Lambda@Edge-Kontingenten finden Sie unter Kontingente für Lambda@Edge. Weitere Informationen zur Identifizierung und Behebung von Lambda@Edge-Fehlern finden Sie unter the section called “Testen und Debuggen”. Die Lambda-Servicekontingente finden Sie auch im Entwicklungshandbuch für AWS Lambda .

HTTP 504-Statuscode (Gateway Timeout)

Ein HTTP 504-Statuscode (Gateway-Timeout) weist darauf hin, dass bei der CloudFront Weiterleitung einer Anfrage an den Ursprung (weil sich das angeforderte Objekt nicht im Edge-Cache befand) einer der folgenden Ereignisse eingetreten ist:

- Der Ursprung hat einen HTTP 504-Statuscode an CloudFront zurückgegeben.
- Der Ursprung reagierte nicht, bevor die Anforderung ablief.

CloudFront gibt einen HTTP 504-Statuscode zurück, wenn der Datenverkehr zum Ursprung durch eine Firewall oder Sicherheitsgruppe blockiert wird oder wenn der Ursprung im Internet nicht zugänglich ist. Überprüfen Sie diese Punkte zuerst. Wenn der Zugriff nicht das Problem ist, sehen Sie sich Anwendungsverzögerungen und Server-Timeouts an, um die Probleme zu ermitteln und zu beheben.

Themen

- Konfigurieren Sie die Firewall auf Ihrem Ursprungsserver so, dass CloudFront Datenverkehr zugelassen wird
- Konfigurieren Sie die Sicherheitsgruppen auf Ihrem Ursprungsserver so, dass Datenverkehr zugelassen CloudFront wird
- Erlauben des Zugriffs auf Ihren benutzerdefinierten Ursprungsserver über das Internet
- Suchen und Beheben verzögerter Antworten von Anwendungen auf Ihrem Ursprungsserver

Konfigurieren Sie die Firewall auf Ihrem Ursprungsserver so, dass CloudFront Datenverkehr zugelassen wird

Wenn die Firewall auf Ihrem Ursprungsserver den CloudFront Datenverkehr blockiert, CloudFront gibt sie einen HTTP 504-Statuscode zurück. Stellen Sie daher sicher, dass dies nicht das Problem ist, bevor Sie nach anderen Problemen suchen.

Die Methode, die Sie nutzen, um zu bestimmen, ob es sich um ein Problem mit Ihrer Firewall handelt, hängt davon ab, welches System Ihr Ursprungs-Server verwendet:

- Wenn Sie eine IPTable-Firewall auf einem Linux-Server verwenden, suchen Sie Tools und Informationen, die Sie bei der Arbeit mit IPTables unterstützen.

- Wenn Sie die Windows-Firewall auf einem Windows-Server verwenden, finden Sie weitere Informationen unter Hinzufügen oder Bearbeiten einer Firewallregel in der Microsoft-Dokumentation.

Achten Sie bei der Bewertung der Firewallkonfiguration auf Ihrem Ursprungsserver auf alle Firewalls oder Sicherheitsregeln, die den Datenverkehr von CloudFront Edge-Standorten aus blockieren, basierend auf dem veröffentlichten IP-Adressbereich. Weitere Informationen finden Sie unter Standorte und IP-Adressbereiche von CloudFront Edge-Servern.

Wenn der CloudFront IP-Adressbereich eine Verbindung zu Ihrem Ursprungsserver herstellen darf, stellen Sie sicher, dass Sie die Sicherheitsregeln Ihres Servers aktualisieren, um die Änderungen zu berücksichtigen. Sie können ein Amazon-SNS-Thema abonnieren und Benachrichtigungen erhalten, wenn die IP-Adressbereichsdatei aktualisiert wird. Nachdem Sie die Benachrichtigung erhalten haben, können Sie Code verwenden, um die Datei abzurufen, sie zu analysieren und ggf. Anpassungen Ihrer lokalen Umgebung vorzunehmen. Weitere Informationen finden Sie im AWS News-Blog unter Abonnieren von Änderungen an AWS öffentlichen IP-Adressen mithilfe von Amazon SNS.

Konfigurieren Sie die Sicherheitsgruppen auf Ihrem Ursprungsserver so, dass Datenverkehr zugelassen CloudFront wird

Wenn Ihr Ursprungsserver Elastic Load Balancing verwendet, überprüfen Sie die ELB-Sicherheitsgruppen und stellen Sie sicher, dass die Sicherheitsgruppen eingehenden Datenverkehr von CloudFront zulassen.

Sie können es auch verwenden AWS Lambda , um Ihre Sicherheitsgruppen automatisch zu aktualisieren, um eingehenden Datenverkehr von zuzulassen. CloudFront

Erlauben des Zugriffs auf Ihren benutzerdefinierten Ursprungsserver über das Internet

Wenn Sie nicht auf Ihren benutzerdefinierten Ursprungsserver zugreifen CloudFront können, weil er im Internet nicht öffentlich verfügbar ist, wird ein HTTP 504-Fehler CloudFront zurückgegeben.

CloudFront Edge-Standorte stellen über das Internet eine Verbindung zu den Ursprungsservern her. Wenn sich Ihr benutzerdefinierter Ursprung in einem privaten Netzwerk befindet, CloudFront können Sie ihn nicht erreichen. Aus diesem Grund können Sie keine privaten Server, einschließlich interner Classic Load Balancer, als Ursprungsserver mit CloudFront verwenden.

Um zu überprüfen, ob der Internetverkehr eine Verbindung zu Ihrem Ursprungsserver herstellen kann, führen Sie die folgenden Befehle aus (wo *OriginDomainName* ist der Domainname für Ihren Server):

Für HTTPS-Datenverkehr:

- nc -zv 443 *OriginDomainName*
- Telnet 443 *OriginDomainName*

Für HTTP-Datenverkehr:

- nc -zv 80 *OriginDomainName*
- Telnet 80 *OriginDomainName*

Suchen und Beheben verzögerter Antworten von Anwendungen auf Ihrem Ursprungsserver

Server-Timeouts sind häufig das Ergebnis einer Anwendung, die recht lange braucht, um zu reagieren, oder eines Timeout-Werts, der zu niedrig eingestellt ist.

Eine schnelle Abhilfe zum Vermeiden des HTTP-Fehlers 504 besteht darin, einen höheren CloudFront-Timeout-Wert für Ihre Verteilung festzulegen. Wir empfehlen jedoch, dass Sie zunächst sicherstellen, dass Sie alle Leistungs- und Latenzprobleme mit der Anwendung und dem Ursprungs-Server beheben. Anschließend können Sie einen angemessenen Timeout-Wert festlegen, der zur Vermeidung des HTTP-Fehlers 504 beiträgt und eine gute Reaktionsfähigkeit für Benutzer bietet.

Im Folgenden finden Sie eine Übersicht über die Schritte, die Sie ausführen können, um Leistungsprobleme zu ermitteln und zu beheben:

1. Messen Sie die normale Latenz und die Latenz bei hoher Last (Reaktionsfähigkeit) Ihrer Webanwendung.
2. Fügen Sie weitere Ressourcen, z. B. CPU oder Speicher (bei Bedarf) hinzu. Führen Sie andere Schritte aus, um die Probleme zu beheben, z. B. Optimierung von Datenbankabfragen für Szenarien mit hoher Last.
3. Passen Sie bei Bedarf den Timeout-Wert für Ihre CloudFront Distribution an.

Im Folgenden finden Sie Details zu jedem einzelnen Schritt.

Messen der normalen Latenz und der Latenz bei hoher Last

Um festzustellen, ob auf einem oder mehreren Backend-Webanwendungsservern eine hohe Latenz vorliegt, führen Sie den folgenden Linux curl-Befehl auf jedem Server aus:

```
curl -w "DNS Lookup Time: %{time_namelookup} \nConnect time: %{time_connect}
 \nTLS Setup: %{time_appconnect} \nRedirect Time: %{time_redirect} \nTime to first
 byte: %{time_starttransfer} \nTotal time: %{time_total} \n" -o /dev/null https://
www.example.com/yourobject
```

> **Note**
>
> Wenn Sie Windows auf Ihren Servern ausführen, können Sie curl für Windows suchen und herunterladen, um einen entsprechenden Befehl auszuführen.

Beachten Sie beim Messen und Auswerten der Latenz einer Anwendung, die auf Ihrem Server ausgeführt wird, folgende Punkte:

- Latenzwerte sind für jede Anwendung relativ. Allerdings ist eine Zeit bis zum ersten Byte in Millisekunden statt Sekunden oder mehr sinnvoll.
- Wenn Sie die Anwendungslatenz unter normalen Lastbedingungen messen und das Ergebnis in Ordnung ist, sollten Sie daran denken, dass bei Viewern dennoch Timeouts bei hoher Last auftreten können. Wenn eine hohe Nachfrage besteht, können Server Antworten verzögert senden oder gar nicht reagieren. Überprüfen Sie zur Vermeidung von Latenzproblemen bei hoher Last Ihre Serverressourcen wie CPU, Arbeitsspeicher sowie Lese- und Schreibvorgänge auf Speichermedien, um sicherzustellen, dass Ihre Server über die Kapazitäten zur Skalierung für hohe Auslastung verfügen.

  Sie können den folgenden Linux-Befehl zum Überprüfen des Speichers ausführen, der von Apache-Prozessen verwendet wird:

  ```
  watch -n 1 "echo -n 'Apache Processes: ' && ps -C apache2 --no-headers |
  wc -l && free -m"
  ```

- Eine hohe CPU-Auslastung auf dem Server kann die Leistung einer Anwendung erheblich reduzieren. Wenn Sie eine Amazon EC2-Instance für Ihren Backend-Server verwenden, überprüfen Sie die CloudWatch Messwerte für den Server, um die CPU-Auslastung zu überprüfen. Weitere Informationen finden Sie im CloudWatch Amazon-Benutzerhandbuch. Wenn Sie Ihre

eigenen Server verwenden, finden Sie in der Hilfedokumentation des Servers Anleitungen zum Prüfen der CPU-Auslastung.

- Überprüfen Sie, ob andere potenzielle Probleme unter hoher Last vorliegen, wie beispielsweise Datenbankabfragen, die bei einem hohen Volume von Anfragen langsam ausgeführt werden.

Hinzufügen von Ressourcen und Optimieren von Servern und Datenbanken

Nachdem Sie die Reaktionsfähigkeit Ihrer Anwendungen und Server bewertet haben, stellen Sie sicher, dass Sie über ausreichend Ressourcen für normalen Datenverkehr und Situationen mit hoher Auslastung verfügen:

- Wenn Sie einen eigenen Server nutzen, stellen Sie basierend auf Ihrer Bewertung sicher, dass dieser über ausreichend CPU, Arbeitsspeicher und Festplattenspeicher für die Verarbeitung von Viewer-Anfragen verfügt.
- Wenn Sie eine Amazon EC2-Instance als Backend-Server verwenden, stellen Sie sicher, dass der Instance-Typ über die Ressourcen verfügt, die für die Verarbeitung der eingehenden Anforderungen erforderlich sind. Weitere Informationen finden Sie unter Instance-Typen im Amazon EC2-Benutzerhandbuch.

Außerdem sollten Sie ggf. die folgenden Optimierungsschritte ausführen, um Timeouts zu vermeiden:

- Wenn der Wert für die Zeit bis zum ersten Byte, der vom curl-Befehl zurückgegeben wird, hoch erscheint, ergreifen Sie Maßnahmen zur Verbesserung der Leistung Ihrer Anwendung. Durch Verbessern der Anwendungsreaktionsfähigkeit lassen sich wiederum Timeout-Fehler reduzieren.
- Optimieren Sie Datenbankabfragen, um sicherzustellen, dass sie hohe Abfragezahlen verarbeiten können, ohne dass die Leistung beeinträchtigt wird.
- Richten Sie (persistente) Keepalive-Verbindungen auf Ihrem Backend-Server ein. Mit dieser Option lassen sich Latenzen vermeiden, die auftreten, wenn Verbindungen für nachfolgende Anfragen oder Benutzer erneut hergestellt werden müssen.
- Wenn Sie Elastic Load Balancing als Ursprung verwenden, hat der 504-Fehler möglicherweise folgende Ursachen:
  - Der Load Balancer kann vor Ablauf des Verbindungstimeouts (10 Sekunden) keine Verbindung zum Ziel herstellen.
  - Der Load Balancer stellt eine Verbindung zum Ziel her, aber das Ziel reagiert nicht vor Ablauf des Verbindungstimeouts.

  - Die Netzwerk-Zugriffssteuerungsliste (ACL) für das Subnetz lässt an den flüchtigen Ports (1024-65535) keinen Datenverkehr von den Zielen zu den Load-Balancer-Knoten zu.
  - Das Ziel gibt einen Inhaltslängen-Header zurück, der größer ist als der Entitätskörper. Beim Warten auf die fehlenden Byte wird das Zeitlimit des Load Balancer überschritten.
  - Das Ziel ist eine Lambda-Funktion und Lambda reagiert vor Ablauf des Verbindungstimeouts nicht.

  Weitere Informationen zur Reduzierung der Latenz finden Sie unter Wie behebe ich Probleme mit hoher Latenz auf meinem ELB Classic Load Balancer?
- Wenn Sie MediaTailor als Herkunft verwenden, sind die folgenden Ursachen für einen 504-Fehler möglich:
  - Wenn relative URLs falsch behandelt werden, MediaTailor können von den Playern falsch formatierte URLs empfangen werden.
  - Wenn dies der Ursprung des Manifests MediaPackage ist MediaTailor, können MediaPackage 404-Manifestfehler MediaTailor dazu führen, dass ein 504-Fehler zurückgegeben wird.
  - Es dauert mehr als 2 Sekunden, bis die MediaTailor Anfrage an den Originalserver abgeschlossen ist.
- Wenn Sie Amazon API Gateway als Ursprung verwenden, hat der 504-Fehler möglicherweise folgende Ursachen:
  - Eine Integrationsanforderung dauert länger als der maximale Integrationstimeout-Parameter Ihrer REST-API von API Gateway. Weitere Informationen finden Sie unter Wie kann ich den HTTP-Fehler 504 zum API-Gateway-Timeout beheben?

Passen Sie bei Bedarf den CloudFront Timeout-Wert an

Möglicherweise haben Sie die langsame Anwendungsleistung, die Kapazität des Ursprungsservers und andere Probleme evaluiert und behoben, aber bei Viewern treten immer noch HTTP 504-Fehler auf. In diesem Fall sollten Sie erwägen, die in Ihrer Verteilung für das Timeout der ursprünglichen Antwort angegebene Zeit zu ändern. Weitere Informationen finden Sie unter the section called “Antwort-Timeout”.

Testen CloudFront

Herkömmliche Lasttestmethoden funktionieren nicht gut, CloudFront da DNS CloudFront verwendet wird, um die Lasten an geografisch verteilten Edge-Standorten und innerhalb der einzelnen Edge-

Standorte auszugleichen. Wenn ein Client Inhalte von anfordert CloudFront, erhält der Client eine DNS-Antwort, die eine Reihe von IP-Adressen enthält. Wenn Sie beim Testen Anfragen an nur eine der IP-Adressen senden, die DNS zurückgibt, testen Sie nur eine kleine Teilmenge der Ressourcen an einem CloudFront Edge-Standort, was die tatsächlichen Verkehrsmuster nicht genau wiedergibt. Je nach angeforderter Datenmenge können Tests auf diese Weise die Leistung dieser kleinen Untergruppe von Servern überlasten und beeinträchtigen. CloudFront

CloudFront ist so konzipiert, dass es für Zuschauer mit unterschiedlichen Client-IP-Adressen und unterschiedlichen DNS-Resolvern in mehreren geografischen Regionen skaliert werden kann. Um Auslastungstests durchzuführen, bei denen die CloudFront Leistung genau bewertet wird, empfehlen wir, dass Sie alle folgenden Schritte ausführen:

- Senden Sie Clientanfragen von mehreren geografischen Regionen aus ab.
- Konfigurieren Sie Ihre Tests so, dass die Clients voneinander unabhängige DNS-Anforderungen stellen. Jeder Client erhält dann einen anderen Satz von IP-Adressen von DNS.
- Verteilen Sie die Anforderungen der einzelnen Clients über die gesamte Menge der IP-Adressen, die von DNS zurückgegeben werden. Dadurch wird sichergestellt, dass die Last auf mehrere Server an einem CloudFront Edge-Standort verteilt wird.

Hinweise

- Belastungstests sind bei Cache-Verhaltensweisen mit Lambda@Edge-Auslösern von Viewer-Anforderungen oder -Antworten nicht zulässig.
- Belastungstests sind bei Ursprüngen, für die Origin Shield aktiviert ist, nicht zulässig.

Kontingente

Sie können eine Erhöhung des CloudFront Kontingents beantragen, indem Sie die folgenden Optionen verwenden:

- Sie können die Service-Quotas-Konsole oder die AWS Command Line Interface verwenden. Weitere Informationen finden Sie unter den folgenden Themen:
  - Anfordern einer Kontingenterhöhung im Benutzerhandbuch für Service Quotas
  - request-service-quota-increase in der Befehlsreferenz AWS CLI
- Wenn ein CloudFront Kontingent nicht unter Service Quotas verfügbar ist, verwenden Sie den, AWS Support Center Console um einen Fall zur Erhöhung des Servicekontingents zu erstellen.

CloudFront unterliegt den folgenden Kontingenten.

Themen

- Allgemeine Kontingente
- Allgemeine Kontingente für Verteilungen
- Allgemeine Kontingente für Richtlinien
- Kontingente für mTLs und Trust Stores
- Kontingente für CloudFront Funktionen
- Kontingente für Verbindungsfunktionen
- Kontingente für Schlüsselwertspeicher
- Kontingente für Lambda@Edge
- Kontingente für SSL-Zertifikate
- Kontingente für Aufhebungen
- Kontingente für Schlüsselgruppen
- Kontingente für WebSocket Verbindungen
- Kontingente für Verschlüsselung auf Feldebene
- Kontingente für Cookies (Legacy-Cache-Einstellungen)
- Kontingente für Abfragezeichenfolgen (Legacy-Cache-Einstellungen)
- Kontingente für Header
- Kontingente zu Multi-Tenant-Distributionen

- Ähnliche Informationen

Allgemeine Kontingente

Entity	Standardkontingent
Datenübertragungsrate pro Verteilung (Dieses Kontingent gilt nicht für Distributionen, die CloudFront Pauschalpreise abonniert haben. Weitere Informationen finden Sie unter.) ???	150 Gbit/s Höheres Kontingent anfordern
Anforderungen pro Sekunde und Verteilung (Dieses Kontingent gilt nicht für Distributionen, die CloudFront Pauschalpreise abonniert haben. Weitere Informationen finden Sie unter.) ???	250 000 Höheres Kontingent anfordern
Tags, die einer Verteilung hinzugefügt werden können	50 Höheres Kontingent anfordern
Dateien, die Sie mit einer Verteilung bereitstellen können	Kein Kontingent
maximale Länge einer Anforderung oder Ursprungsantwort, einschließlich Headern und Abfragezeichenfolgen, jedoch ohne Textinhalt	32.768 Byte
Maximale Länge einer URL	8,192 Bytes
Maximale Anzahl von Konfigurationen für die Zustellung von Zugriffsprotokollen in Echtzeit pro AWS-Konto	150
Maximale Anzahl von Verknüpfungen pro Web-ACL	100 Höheres Kontingent anfordern

Allgemeine Kontingente für Verteilungen

Entity	Standardkontingent
Alternative Domänennamen (CNAMEs) pro Verteilung Weitere Informationen finden Sie unter Verwenden von benutzerdefinierten URLs durch Hinzufügen von alternativen Domainnamen (CNAMEs).	100 Höheres Kontingent anfordern
Cache-Verhalten pro Verteilung	75 Höheres Kontingent anfordern
Verbindungsversuche pro Ursprung Weitere Informationen finden Sie unter Verbindungsversuche.	1-3
Verbindungs-Timeout pro Ursprung Weitere Informationen finden Sie unter Verbindungstimeout.	1-10 Sekunden
Reaktions-Timeout per Ursprung Dies wird auch als Ursprungslese-Timeout oder Ursprungsanforderungs-Timeout bezeichnet. Weitere Informationen finden Sie unter Antwort-Timeout.	1—120 Sekunden Höheres Kontingent anfordern
Keep-alive Timeout pro Ursprung Weitere Informationen finden Sie unter Keep-alive Timeout (nur benutzerdefinierte und VPC-Ursprünge).	1-300 Sekunden Höheres Kontingent anfordern
Verteilungen pro AWS-Konto Weitere Informationen finden Sie unter Eine Distribution erstellen.	500 Höheres Kontingent anfordern
Distributionen pro Ursprungszugriffssteuerung	100

Entity	Standardkontingent
	Höheres Kontingent anfordern
Distributionen innerhalb der Kette von Anforderungen an den Ursprungs endpunkt Wir empfehlen nicht, eine Distribution vor einer anderen zu platzieren. Das Überschreiten dieses Kontingents führt zu einem 403-Fehler.	2
Dateikomprimierung: Bereich der Dateigrößen, CloudFront der komprimiert wird Weitere Informationen finden Sie unter Bereitstellen von komprimierten Dateien.	1 000 bis 10 000 000 Bytes
Maximale Dateigröße pro HTTP-GET-Antwort, die zwischengespeichert werden kann. Nur die Antworten für eine HTTP-GET-Anfrage werden zwischeng espeichert. Antworten für POST oder PUT werden nicht zwischeng espeichert.	50 GB
Maximale Größe eines HTTP-Anforderungstextes.	64 GB
Origin-Zugriffskontrollen pro AWS-Konto	100 Höheres Kontingent anfordern
Zugriffsidentitäten von Herkunft pro AWS-Konto	100 Höheres Kontingent anfordern
Ursprünge pro Verteilung	100 Höheres Kontingent anfordern

Entity	Standardkontingent
Ursprungsgruppen pro Verteilung	10 Höheres Kontingent anfordern
Bereitstellen von Verteilungen pro AWS-Konto Weitere Informationen finden Sie unter the section called “Verwenden der kontinuierlichen Bereitstellung zum sicheren Testen von Änderungen”.	20 Höheres Kontingent anfordern
Distributionen, die demselben VPC-Ursprung zugeordnet sind	50
VPC-Ursprünge pro AWS-Konto	25 Höheres Kontingent anfordern
maximale Anzahl von Distributionen, die einer einzelnen statischen Anycast-IP-Liste zugeordnet werden können	100 Höheres Kontingent anfordern

Allgemeine Kontingente für Richtlinien

Entity	Standardkontingent
Benutzerdefinierte Cache-Richtlinien pro AWS-Konto (Gilt nicht für CloudFrontverwaltete Cache-Richtlinien)	20 Höheres Kontingent anfordern
Verteilungen, die derselben Cache-Richtlinie zugeordnet sind	100
Abfragezeichenfolgen pro Cache-Richtlinie	10

Entity	Standardkontingent
	Höheres Kontingent anfordern
Header pro Cache-Richtlinie	10 Höheres Kontingent anfordern
Cookies pro Cache-Richtlinie	10 Höheres Kontingent anfordern
Kombinierte Gesamtlänge aller Abfragezeichenfolgen, Header und Cookie-Namen in einer Cache-Richtlinie	1024
Benutzerdefinierte Richtlinien für ursprüngliche Anfragen pro AWS-Konto (Gilt nicht für Richtlinien für CloudFront verwaltete Anfragen mit Herkunft)	20 Höheres Kontingent anfordern
Verteilungen, die derselben Ursprungsanforderungsrichtlinie zugeordnet sind	100
Abfragezeichenfolgen pro Ursprungsanforderungsrichtlinie	10 Höheres Kontingent anfordern
Header pro Ursprungsanforderungsrichtlinie	10 Höheres Kontingent anfordern
Cookies pro Ursprungsanforderungsrichtlinie	10 Höheres Kontingent anfordern

Entity	Standardkontingent
Kombinierte Gesamtlänge aller Abfragezeichenfolgen, Header und Cookie-Namen in einer Ursprungsanfrage-Richtlinie	1024
Richtlinien für benutzerdefinierte Antwort-Header pro AWS-Konto (Gilt nicht für Richtlinien für CloudFront verwaltete Antwort-Header)	20 Höheres Kontingent anfordern
Verteilungen, die derselben Antwort-Header.Richtlinie zugeordnet sind	100 Höheres Kontingent anfordern
Benutzerdefinierte Header pro Antwort-Header-Richtlinie	10 Höheres Kontingent anfordern
Richtlinien für die kontinuierliche Bereitstellung pro AWS-Konto	20 Höheres Kontingent anfordern

Kontingente für mTLs und Trust Stores

Entity	Standardkontingent
Vertraue Geschäften pro AWS-Konto	20 Höheres Kontingent anfordern
Verteilungen pro Trust Store	300
Größe des CA-Paketes	1 MB
Größe des Zertifikats im CA-Paket	16384

Entity	Standardkontingent
	Höheres Kontingent anfordern
Anzahl der Zertifikate im CA-Paket	25
Tiefe der Zertifikatskette	4

Kontingente für CloudFront Funktionen

Entity	Standardkontingent
Funktionen pro AWS-Konto	100
Maximale Funktionsgröße Dieses Kontingent ist nicht anpassbar. Um zusätzliche Daten für Ihre CloudFront Funktionen zu speichern, erstellen Sie einen Schlüssel wertspeicher und fügen Sie Ihre Schlüssel-Wert-Paare hinzu. Weitere Informationen finden Sie unter Amazon CloudFront KeyValueStore.	10 KB
Maximaler Funktionsspeicher	2 MB
Verteilungen, die derselben Funktion zugeordnet sind	100

Zusätzlich zu diesen Kontingenten gibt es einige weitere Einschränkungen bei der Verwendung von CloudFront Functions. Weitere Informationen finden Sie unter Einschränkungen bei Funktionen CloudFront.

Kontingente für Verbindungsfunktionen

Entity	Standardkontingent
Verbindungsfunktionen pro AWS-Konto	100
Maximale Größe der Verbindungsfunktion	10 KB

Entity	Standardkontingent
Dieses Kontingent ist nicht anpassbar. Um zusätzliche Daten für Ihre Verbindungsfunktionen zu speichern, erstellen Sie einen Schlüssel wertspeicher und fügen Sie Ihre Schlüssel-Wert-Paare hinzu. Weitere Informationen finden Sie unter Amazon CloudFront KeyValueStore.	
Maximaler Speicher für Verbindungsfunktionen	2 MB
Distributionen, die derselben Verbindungsfunktion zugeordnet sind	100

Zusätzlich zu diesen Kontingenten gibt es einige weitere Einschränkungen bei der Verwendung von Verbindungsfunktionen. Weitere Informationen finden Sie unter Ordnen Sie eine CloudFront Verbindungsfunktion zu.

Kontingente für Schlüsselwertspeicher

Entity	Standardkontingent
maximale Größe eines Schlüssels in einem Schlüssel-Wert-Paar	512 Byte
maximale Größe des Werts in einem Schlüssel-Wert-Paar	1 KB
Maximale Anzahl von Schlüssel-Wert-Paaren, die Sie in einer einzigen API-Anfrage aktualisieren können	50 Schlüssel oder 3 MB Nutzlast, je nachdem, was zuerst erreicht wird
Maximale Größe eines einzelnen Schlüsselwertspeichers	5 MB
Maximale Anzahl von Funktionen, denen ein einzelner Schlüssel wertspeicher zugeordnet werden kann	100
Maximale Anzahl von Schlüsselwertspeichern pro Funktion	1
Maximale Anzahl von Schlüsselwertspeichern pro Konto	200

Entity	Standardkontingent
	Höheres Kontingent anfordern

Kontingente für Lambda@Edge

Allgemeine Kontingente

Entity	Standardkontingent
Derartige AWS-Konto Distributionen können Lambda @Edge -Funktionen haben	500 Höheres Kontingent anfordern
Lambda@Edge-Funktionen pro Verteilung	100 Höheres Kontingent anfordern
Gleichzeitige Ausführungen Hinweise • AWS Lambda verwaltet die Parallelitätsquoten für Lambda @Edge. Alle Lambda-Funktionen in der AWS-Region teilen sich dieses Kontingent. • Wir empfehlen Ihnen, die Quote für gleichzeitige Ausführungen in allen Bereichen zu überprüfen, von AWS-Regionen denen Sie erwarten, dass Ihre Zuschaueranfragen herkommen. Darüber hinaus kann jede Instanz Ihrer Lambda @Edge -Funktion bis zu 10 Anfragen pro Sekunde bearbeiten. Das Gesamtlimit für Aufrufe beträgt das Zehnfache Ihres Parallelitätslimits.	1.000 (jeweils AWS-Region) Höheres Kontingent anfordern

Entity	Standardkontingent
Weitere Informationen finden Sie in den folgenden Themen im AWS Lambda Entwicklerhandbuch: • Die Lambda-Funktionsskalierung verstehen • Lambda-API-Anfragen	
Verteilungen, die derselben Funktion zugeordnet sind	500
Maximale komprimierte Größe einer Lambda-Funktion und aller enthaltenen Bibliotheken	50 MB
Lambda@Edge-Anforderungen pro Sekunde (jede unterstützte AWS-Region) Weitere Informationen finden Sie unter Gleichzeitigkeitskontingente im Entwicklerhandbuch für AWS Lambda .	10.000
Funktionsspeichergröße	Wie Lambda-Kontingente
Funktions-Timeout. Die Funktion kann Netzwerkaufrufe an Ressourcen wie Amazon S3-Buckets, DynamoDB-Tabellen oder Amazon EC2-Instances in AWS-Regionen durchführen.	30 Sekunden

Kontingente, die sich nach Ereignistyp unterscheiden

Entity	Viewer-Anfrage- und Viewer-Antwortereignisse	Ursprungsanfrage- und Ursprungsantwortereignisse
Größe einer Antwort, die von einer Lambda-Funktion erzeugt wird, einschließlich Header und Textkörper	40 KB	1 MB

Hinweise

- Eine Liste zusätzlicher Lambda @Edge -Kontingente, die über Service Quotas erhöht werden können, finden Sie unter CloudFrontAmazon-Endpunkte und Kontingente in der. Allgemeine AWS-Referenz
- Zusätzlich zu diesen Kontingenten gibt es bei der Verwendung der Lambda@Edge-Funktionen noch einige andere Einschränkungen. Weitere Informationen finden Sie unter Einschränkungen für Lambda@Edge.

Kontingente für SSL-Zertifikate

Entity	Standardkontingent
SSL-Zertifikate gelten für die AWS-Konto Bearbeitung von HTTPS-Anfragen mit dedizierten IP-Adressen (kein Kontingent bei der Bearbeitung von HTTPS-Anfragen über SNI) Weitere Informationen finden Sie unter Verwenden Sie HTTPS mit CloudFront.	2 Höheres Kontingent anfordern
SSL-Zertifikate, die einer CloudFront Distribution zugeordnet werden können	1

Wenn Ihr SSL-Zertifikat speziell für die HTTPS-Kommunikation zwischen Zuschauern vorgesehen ist und CloudFront Sie AWS Certificate Manager (ACM) oder den IAM-Zertifikatsspeicher für die Bereitstellung oder den Import Ihres Zertifikats verwendet haben, gelten zusätzliche Kontingente. Weitere Informationen finden Sie unter Kontingente für die Verwendung von SSL/TLS Zertifikaten mit CloudFront (HTTPS nur zwischen Zuschauern und CloudFront nur).

Es gibt auch Kontingente für die Anzahl der SSL-Zertifikate, die Sie in AWS Certificate Manager (ACM) importieren oder hochladen AWS Identity and Access Management (IAM) können. Weitere Informationen finden Sie unter Erhöhen Sie die Kontingente für Zertifikate SSL/TLS.

Kontingente für Aufhebungen

Entity	Standardkontingent
Anzahl der Pfade oder Tags Weitere Informationen erhalten Sie unter Aufheben der Gültigkeit von Dateien zum Entfernen von Inhalten und Aufhebung der Gültigkeit durch Cache-Tags.	150 Pfade oder Tags pro Sekunde
Anzahl der Ungültigerklärungen mit Platzhaltern	1 Wildcard-Invalidierung pro Sekunde
Tag-Invalidierung: maximale Anzahl von Dateien, die eine tagbasierte Invalidierung verarbeiten kann	Kein Kontingent
Wildcard-Invalidierung: maximale Anzahl von Dateien, die eine Wildcard-Invalidierung verarbeiten kann	Kein Kontingent

Kontingente für Schlüsselgruppen

Entity	Standardkontingent
Öffentliche Schlüssel in einer einzelnen Schlüsselgruppe	5 Höheres Kontingent anfordern
Schlüsselgruppen, die einem einzelnen Cache-Verhalten zugeordnet sind	4 Höheres Kontingent anfordern
Schlüsselgruppen pro AWS-Konto	10 Höheres Kontingent anfordern

Entity	Standardkontingent
Verteilungen, die einer einzelnen Schlüsselgruppe zugeordnet sind	100 Höheres Kontingent anfordern

Kontingente für WebSocket Verbindungen

Entity	Standardkontingent
Ursprungs-Reaktions-Timeout (Leerlauf-Timeout)	10 Minuten Wenn CloudFront innerhalb der letzten 10 Minuten keine Byte erkannt wurden, die vom Ursprung an den Client gesendet wurden, wird davon ausgegangen, dass die Verbindung inaktiv ist, und wird geschlossen.

Kontingente für Verschlüsselung auf Feldebene

Entity	Standardkontingent
Maximale Länge eines zu verschlüsselnden Feldes Weitere Informationen finden Sie unter Vertrauliche Daten durch Verschlüsselung auf Feldebene schützen.	16 KB

Entity	Standardkontingent
Maximale Anzahl von Feldern in einem Anforderungstext bei konfiguri erter Verschlüsselung auf Feldebene	10
Maximale Länge eines Anforderungstexts, wenn die Verschlüsselung auf Feldebene konfiguriert ist	1 MB
Maximale Anzahl von Verschlüsselungskonfigurationen auf Feldebene, die einer Konfiguration zugeordnet werden können AWS-Konto	10
Maximale Anzahl von Verschlüsselungsprofilen auf Feldebene, die einem zugeordnet werden können AWS-Konto	10
Maximale Anzahl der öffentlichen Schlüssel, die einem AWS-Konto hinzugefügt werden können	10
Maximale Anzahl der zu verschlüsselnden Felder, die in einem Profil angegeben werden kann	10
Maximale Anzahl von CloudFront Verteilungen, die einer Verschlüs selungskonfiguration auf Feldebene zugeordnet werden können	20
Maximale Anzahl der Abfrageargument-Profilzuordnungen, die in eine Konfiguration für die Verschlüsselung auf Feldebene aufgenommen werden können	5

Kontingente für Cookies (Legacy-Cache-Einstellungen)

Diese Kontingente gelten für CloudFront ältere Cache-Einstellungen. Wir empfehlen die Verwendung einer Cache-Richtlinie oder einer Ursprungsanforderungsrichtlinie anstelle der Legacy-Einstellungen.

Entity	Standardkontingent
Cookies pro Cache-Verhalten Weitere Informationen finden Sie unter Zwischenspeichern von Inhalten auf der Grundlage von Cookies.	10 Höheres Kontingent anfordern

Entity	Standardkontingent
Gesamtzahl der Byte in Cookie-Namen (gilt nicht, wenn Sie so konfiguri eren CloudFront , dass alle Cookies an den Ursprung weitergeleitet werden)	512 abzüglich der Anzahl der Cookies

Kontingente für Abfragezeichenfolgen (Legacy-Cache-Einstellungen)

Diese Kontingente gelten für CloudFront ältere Cache-Einstellungen. Wir empfehlen die Verwendung einer Cache-Richtlinie oder einer Ursprungsanforderungsrichtlinie anstelle der Legacy-Einstellungen.

Entity	Standardkontingent
Maximale Anzahl von Zeichen in einer Abfragezeichenfolge	128 Zeichen
Maximale Anzahl der Zeichen für alle Abfragezeichenfolgen im selben Parameter	512 Zeichen
Abfragestrings pro Cache-Verhalten Weitere Informationen finden Sie unter Zwischenspeichern von Inhalten auf der Grundlage von Abfragezeichenfolgeparametern.	10 Höheres Kontingent anfordern

Kontingente für Header

Entity	Standardkontingent
Header pro Cache-Verhalten (Legacy-Cache-Einstellungen) Weitere Informationen finden Sie unter the section called “Zwischen speichern von Inhalten auf der Grundlage von Anforderungsheadern”.	10 Höheres Kontingent anfordern
Weiterleiten von Headern pro Cacheverhalten	25

Entity	Standardkontingent
	Höheres Kontingent anfordern
Benutzerdefinierte Header: maximale Anzahl von benutzerdefinierten Headern, die Sie konfigurieren können, CloudFront um sie zu ursprüngl ichen Anfragen hinzuzufügen Weitere Informationen finden Sie unter the section called “Hinzufügen benutzerdefinierter Header zu Ursprungsanforderungen”.	30 Höheres Kontingent anfordern
Benutzerdefinierte Header: maximale Anzahl benutzerdefinierter Header, die Sie einer Antwort-Header-Richtlinie hinzufügen können	10 Höheres Kontingent anfordern
Benutzerdefinierte Header: maximale Länge eines Header-Namens	256 Zeichen
Benutzerdefinierte Header: maximale Länge eines Header-Wertes	1,783 Zeichen
Benutzerdefinierte Header: maximale Länge aller Werte und Namen kombiniert	10,240 Zeichen
maximale Länge des Wertes des Headers `Content-Security-Policy`	1,783 Zeichen Höheres Kontingent anfordern
maximale Länge eines Wertes des Headers `CORS (Access-Control-Allow-Origin)`	1,783 Zeichen

Kontingente zu Multi-Tenant-Distributionen

Entity	Standardkontingent
Maximale Anzahl von Distributionsmandanten pro AWS-Konto	10.000

Entity	Standardkontingent
	Höheres Kontingent anfordern
Maximale Anzahl von Multi-Tenant-Distributionen pro AWS-Konto	20 Höheres Kontingent anfordern
Maximale Anzahl von Verbindungsgruppen pro AWS-Konto	100 Höheres Kontingent anfordern
maximale Anzahl von Aliasnamen pro Distributionsmandant	100 Höheres Kontingent anfordern
maximale Anzahl von Parametern pro Distributionsmandant	5 Höheres Kontingent anfordern
maximale Anzahl von Parametern pro Multi-Tenant-Distribution	5 Höheres Kontingent anfordern
maximale Anzahl von Parametern in einem Feld in einer Multi-Tenant-Distribution	2 Höheres Kontingent anfordern
maximale Anzahl von Verbindungsgruppen pro statischer Anycast-IP-Liste	5 Höheres Kontingent anfordern

Weitere Informationen zu Multi-Tenant-Distributionen finden Sie unter Verständnis der Funktionsweise von Multi-Tenant-Distributionen.

Ähnliche Informationen

Weitere Informationen finden Sie unter CloudFront Amazon-Endpunkte und Kontingente in der Allgemeine AWS-Referenz.

Codebeispiele für die CloudFront Verwendung AWS SDKs

Die folgenden Codebeispiele zeigen, wie die Verwendung CloudFront mit einem AWS Software Development Kit (SDK) funktioniert.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Während Aktionen Ihnen zeigen, wie Sie einzelne Service-Funktionen aufrufen, können Sie Aktionen im Kontext der zugehörigen Szenarien anzeigen.

Szenarien sind Codebeispiele, die Ihnen zeigen, wie Sie bestimmte Aufgaben ausführen, indem Sie mehrere Funktionen innerhalb eines Service aufrufen oder mit anderen AWS-Services kombinieren.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unterVerwendung CloudFront mit einem AWS SDK. Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Codebeispiele

- Grundlegende Beispiele für die CloudFront Verwendung AWS SDKs
  - Aktionen zur Verwendung CloudFront AWS SDKs
    - Verwenden Sie CreateDistributionmit einem AWS SDK oder CLI
    - Verwenden Sie CreateFunctionmit einem AWS SDK
    - Verwendung von CreateInvalidation mit einer CLI
    - Verwenden Sie CreateKeyGroupmit einem AWS SDK
    - Verwenden Sie CreatePublicKeymit einem AWS SDK oder CLI
    - Verwenden Sie DeleteDistributionmit einem AWS SDK oder CLI
    - Verwendung von GetCloudFrontOriginAccessIdentity mit einer CLI
    - Verwendung von GetCloudFrontOriginAccessIdentityConfig mit einer CLI
    - Verwendung von GetDistribution mit einer CLI
    - Verwenden Sie GetDistributionConfigmit einem AWS SDK oder CLI
    - Verwendung von ListCloudFrontOriginAccessIdentities mit einer CLI
    - Verwenden Sie ListDistributionsmit einem AWS SDK oder CLI
    - Verwenden Sie UpdateDistributionmit einem AWS SDK oder CLI
- Szenarien für die CloudFront Verwendung AWS SDKs
  - SaaS-Manager-Ressourcen erstellen AWS SDK

- Löschen Sie CloudFront Signaturressourcen mit AWS SDK
- Beginnen Sie mit einem einfachen Netzwerk zur Verteilung von Inhalten
- Erste Schritte mit Firewalls für Webanwendungen
- Erstellen Sie signierte URLs und Cookies mit einem AWS SDK

- CloudFront Funktionen, Beispiele für CloudFront
  - Fügen Sie einem CloudFront Functions-Viewer-Antwortereignis HTTP-Sicherheitsheader hinzu
  - Fügen Sie einem CloudFront Functions-Viewer-Antwortereignis einen CORS-Header hinzu
  - Fügen Sie einem Antwortereignis des CloudFront Functions-Viewers einen Cache-Control-Header hinzu
  - Fügen Sie einem CloudFront Functions-Viewer-Anforderungsereignis einen echten Client-IP-Header hinzu
  - Fügen Sie einem CloudFront Functions-Viewer-Anforderungsereignis einen Origin-Header hinzu
  - Fügen Sie index.html hinzu, um URLs ohne Dateinamen in einem CloudFront Functions-Viewer-Anforderungsereignis anzufordern
  - Normalisieren Sie die Parameter der Abfragezeichenfolge in einer CloudFront Functions Viewer-Anforderung
  - In einem CloudFront Functions-Viewer-Anforderungsereignis zu einer neuen URL umleiten
  - Schreiben Sie einen Anforderungs-URI auf der Grundlage der KeyValueStore Konfiguration für ein CloudFront Functions-Viewer-Anforderungsereignis neu
  - Leiten Sie Anfragen in einem CloudFront Functions-Viewer-Anforderungsereignis an einen Ursprung weiter, der näher am Betrachter liegt
  - Verwenden Sie Schlüssel-Wert-Paare in einer CloudFront Functions-Viewer-Anfrage
  - Validieren Sie ein einfaches Token in einer CloudFront Functions-Viewer-Anfrage

Grundlegende Beispiele für die CloudFront Verwendung AWS SDKs

Die folgenden Codebeispiele zeigen, wie Sie die Grundlagen von Amazon CloudFront mit AWS SDKs verwenden können.

Beispiele

- Aktionen zur Verwendung CloudFront AWS SDKs

- Verwenden Sie CreateDistributionmit einem AWS SDK oder CLI
- Verwenden Sie CreateFunctionmit einem AWS SDK
- Verwendung von CreateInvalidation mit einer CLI
- Verwenden Sie CreateKeyGroupmit einem AWS SDK
- Verwenden Sie CreatePublicKeymit einem AWS SDK oder CLI
- Verwenden Sie DeleteDistributionmit einem AWS SDK oder CLI
- Verwendung von GetCloudFrontOriginAccessIdentity mit einer CLI
- Verwendung von GetCloudFrontOriginAccessIdentityConfig mit einer CLI
- Verwendung von GetDistribution mit einer CLI
- Verwenden Sie GetDistributionConfigmit einem AWS SDK oder CLI
- Verwendung von ListCloudFrontOriginAccessIdentities mit einer CLI
- Verwenden Sie ListDistributionsmit einem AWS SDK oder CLI
- Verwenden Sie UpdateDistributionmit einem AWS SDK oder CLI

Aktionen zur Verwendung CloudFront AWS SDKs

Die folgenden Codebeispiele zeigen, wie einzelne CloudFront Aktionen mit AWS SDKs ausgeführt werden. Jedes Beispiel enthält einen Link zu GitHub, wo Sie Anweisungen zum Einrichten und Ausführen des Codes finden.

Diese Auszüge rufen die CloudFront API auf und sind Codeauszüge aus größeren Programmen, die im Kontext ausgeführt werden müssen. Sie können Aktionen im Kontext unter Szenarien für die CloudFront Verwendung AWS SDKs anzeigen.

Die folgenden Beispiele enthalten nur die am häufigsten verwendeten Aktionen. Eine vollständige Liste finden Sie in der Amazon CloudFront API-Referenz.

Beispiele

- Verwenden Sie CreateDistributionmit einem AWS SDK oder CLI
- Verwenden Sie CreateFunctionmit einem AWS SDK
- Verwendung von CreateInvalidation mit einer CLI
- Verwenden Sie CreateKeyGroupmit einem AWS SDK
- Verwenden Sie CreatePublicKeymit einem AWS SDK oder CLI
- Verwenden Sie DeleteDistributionmit einem AWS SDK oder CLI

- Verwendung von GetCloudFrontOriginAccessIdentity mit einer CLI
- Verwendung von GetCloudFrontOriginAccessIdentityConfig mit einer CLI
- Verwendung von GetDistribution mit einer CLI
- Verwenden Sie GetDistributionConfigmit einem AWS SDK oder CLI
- Verwendung von ListCloudFrontOriginAccessIdentities mit einer CLI
- Verwenden Sie ListDistributionsmit einem AWS SDK oder CLI
- Verwenden Sie UpdateDistributionmit einem AWS SDK oder CLI

Verwenden Sie **`CreateDistribution`**mit einem AWS SDK oder CLI

Die folgenden Code-Beispiele zeigen, wie `CreateDistribution` verwendet wird.

Aktionsbeispiele sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Sie können diese Aktion in den folgenden Codebeispielen im Kontext sehen:

- Erstellen einer Multi-Tenant-Distribution und eines Distributionsmandanten
- Beginnen Sie mit einem grundlegenden Netzwerk zur Verteilung von Inhalten

CLI

AWS CLI

Beispiel 1: Um eine CloudFront Distribution zu erstellen

Im folgenden `create-distribution`-Beispiel wird eine Distribution für einen S3-Bucket mit dem Namen `amzn-s3-demo-bucket` erstellt und `index.html` als Standardstammobjekt mithilfe von Befehlszeilenargumenten festgelegt.

```
aws cloudfront create-distribution \
    --origin-domain-name amzn-s3-demo-bucket.s3.amazonaws.com \
    --default-root-object index.html
```

Ausgabe:

```
{
    "Location": "https://cloudfront.amazonaws.com/2019-03-26/distribution/
EMLARXS9EXAMPLE",
    "ETag": "E9LHASXEXAMPLE",
```

```
    "Distribution": {
        "Id": "EMLARXS9EXAMPLE",
        "ARN": "arn:aws:cloudfront::123456789012:distribution/EMLARXS9EXAMPLE",
        "Status": "InProgress",
        "LastModifiedTime": "2019-11-22T00:55:15.705Z",
        "InProgressInvalidationBatches": 0,
        "DomainName": "d111111abcdef8.cloudfront.net",
        "ActiveTrustedSigners": {
            "Enabled": false,
            "Quantity": 0
        },
        "DistributionConfig": {
            "CallerReference": "cli-example",
            "Aliases": {
                "Quantity": 0
            },
            "DefaultRootObject": "index.html",
            "Origins": {
                "Quantity": 1,
                "Items": [
                    {
                        "Id": "amzn-s3-demo-bucket.s3.amazonaws.com-cli-example",
                        "DomainName": "amzn-s3-demo-bucket.s3.amazonaws.com",
                        "OriginPath": "",
                        "CustomHeaders": {
                            "Quantity": 0
                        },
                        "S3OriginConfig": {
                            "OriginAccessIdentity": ""
                        }
                    }
                ]
            },
            "OriginGroups": {
                "Quantity": 0
            },
            "DefaultCacheBehavior": {
                "TargetOriginId": "amzn-s3-demo-bucket.s3.amazonaws.com-cli-
example",
                "ForwardedValues": {
                    "QueryString": false,
                    "Cookies": {
                        "Forward": "none"
                    },
```

```
                    "Headers": {
                        "Quantity": 0
                    },
                    "QueryStringCacheKeys": {
                        "Quantity": 0
                    }
                },
                "TrustedSigners": {
                    "Enabled": false,
                    "Quantity": 0
                },
                "ViewerProtocolPolicy": "allow-all",
                "MinTTL": 0,
                "AllowedMethods": {
                    "Quantity": 2,
                    "Items": [
                        "HEAD",
                        "GET"
                    ],
                    "CachedMethods": {
                        "Quantity": 2,
                        "Items": [
                            "HEAD",
                            "GET"
                        ]
                    }
                },
                "SmoothStreaming": false,
                "DefaultTTL": 86400,
                "MaxTTL": 31536000,
                "Compress": false,
                "LambdaFunctionAssociations": {
                    "Quantity": 0
                },
                "FieldLevelEncryptionId": ""
            },
            "CacheBehaviors": {
                "Quantity": 0
            },
            "CustomErrorResponses": {
                "Quantity": 0
            },
            "Comment": "",
            "Logging": {
```

```
                "Enabled": false,
                "IncludeCookies": false,
                "Bucket": "",
                "Prefix": ""
            },
            "PriceClass": "PriceClass_All",
            "Enabled": true,
            "ViewerCertificate": {
                "CloudFrontDefaultCertificate": true,
                "MinimumProtocolVersion": "TLSv1",
                "CertificateSource": "cloudfront"
            },
            "Restrictions": {
                "GeoRestriction": {
                    "RestrictionType": "none",
                    "Quantity": 0
                }
            },
            "WebACLId": "",
            "HttpVersion": "http2",
            "IsIPV6Enabled": true
        }
    }
}
```

Beispiel 2: Um eine CloudFront Distribution mit einer JSON-Datei zu erstellen

Im folgenden `create-distribution`-Beispiel wird eine Distribution für einen S3-Bucket mit dem Namen `amzn-s3-demo-bucket` erstellt und `index.html` als Standardstammobjekt mithilfe einer JSON-Datei festgelegt.

```
aws cloudfront create-distribution \
    --distribution-config file://dist-config.json
```

Inhalt von `dist-config.json`:

```
{
    "CallerReference": "cli-example",
    "Aliases": {
        "Quantity": 0
    },
    "DefaultRootObject": "index.html",
    "Origins": {
```

```
        "Quantity": 1,
        "Items": [
            {
                "Id": "amzn-s3-demo-bucket.s3.amazonaws.com-cli-example",
                "DomainName": "amzn-s3-demo-bucket.s3.amazonaws.com",
                "OriginPath": "",
                "CustomHeaders": {
                    "Quantity": 0
                },
                "S3OriginConfig": {
                    "OriginAccessIdentity": ""
                }
            }
        ]
    },
    "OriginGroups": {
        "Quantity": 0
    },
    "DefaultCacheBehavior": {
        "TargetOriginId": "amzn-s3-demo-bucket.s3.amazonaws.com-cli-example",
        "ForwardedValues": {
            "QueryString": false,
            "Cookies": {
                "Forward": "none"
            },
            "Headers": {
                "Quantity": 0
            },
            "QueryStringCacheKeys": {
                "Quantity": 0
            }
        },
        "TrustedSigners": {
            "Enabled": false,
            "Quantity": 0
        },
        "ViewerProtocolPolicy": "allow-all",
        "MinTTL": 0,
        "AllowedMethods": {
            "Quantity": 2,
            "Items": [
                "HEAD",
                "GET"
            ],
```

```
            "CachedMethods": {
                "Quantity": 2,
                "Items": [
                    "HEAD",
                    "GET"
                ]
            }
        },
        "SmoothStreaming": false,
        "DefaultTTL": 86400,
        "MaxTTL": 31536000,
        "Compress": false,
        "LambdaFunctionAssociations": {
            "Quantity": 0
        },
        "FieldLevelEncryptionId": ""
    },
    "CacheBehaviors": {
        "Quantity": 0
    },
    "CustomErrorResponses": {
        "Quantity": 0
    },
    "Comment": "",
    "Logging": {
        "Enabled": false,
        "IncludeCookies": false,
        "Bucket": "",
        "Prefix": ""
    },
    "PriceClass": "PriceClass_All",
    "Enabled": true,
    "ViewerCertificate": {
        "CloudFrontDefaultCertificate": true,
        "MinimumProtocolVersion": "TLSv1",
        "CertificateSource": "cloudfront"
    },
    "Restrictions": {
        "GeoRestriction": {
            "RestrictionType": "none",
            "Quantity": 0
        }
    },
    "WebACLId": "",
```

```
    "HttpVersion": "http2",
    "IsIPV6Enabled": true
}
```

Eine Beispielausgabe finden Sie in Beispiel 1.

Beispiel 3: Um eine CloudFront Multi-Tenant-Distribution mit einem Zertifikat zu erstellen

Im folgenden `create-distribution` Beispiel wird eine CloudFront Distribution mit Mehrmandantenunterstützung erstellt und ein TLS-Zertifikat angegeben.

```
aws cloudfront create-distribution \
    --distribution-config file://dist-config.json
```

Inhalt von `dist-config.json`:

```
{
    "CallerReference": "cli-example-with-cert",
    "Comment": "CLI example distribution",
    "DefaultRootObject": "index.html",
    "Origins": {
        "Quantity": 1,
        "Items": [
            {
                "Id": "amzn-s3-demo-bucket.s3.us-east-1.amazonaws.com",
                "DomainName": "amzn-s3-demo-bucket.s3.us-east-1.amazonaws.com",
                "OriginPath": "/{{tenantName}}",
                "CustomHeaders": {
                    "Quantity": 0
                },
                "S3OriginConfig": {
                    "OriginAccessIdentity": ""
                }
            }
        ]
    },
    "DefaultCacheBehavior": {
        "TargetOriginId": "amzn-s3-demo-bucket.s3.us-east-1.amazonaws.com",
        "CachePolicyId": "658327ea-f89d-4fab-a63d-7e88639e5ABC",
        "ViewerProtocolPolicy": "allow-all",
        "AllowedMethods": {
            "Quantity": 2,
            "Items": ["HEAD", "GET"],
```

```
            "CachedMethods": {
                "Quantity": 2,
                "Items": ["HEAD", "GET"]
            }
        }
    },
    "Enabled": true,
    "ViewerCertificate": {
        "ACMCertificateArn": "arn:aws:acm:us-
east-1:123456789012:certificate/191306a1-db01-49ca-90ef-fc414ee5dabc",
        "SSLSupportMethod": "sni-only"
    },
    "HttpVersion": "http2",
    "ConnectionMode": "tenant-only",
    "TenantConfig": {
        "ParameterDefinitions": [
            {
                "Name": "tenantName",
                "Definition": {
                    "StringSchema": {
                        "Comment": "tenantName parameter",
                        "DefaultValue": "root",
                        "Required": false
                    }
                }
            }
        ]
    }
}
```

Ausgabe:

```
{
    "Location": "https://cloudfront.amazonaws.com/2020-05-31/distribution/
E1HVIAU7UABC",
    "ETag": "E20LT7R1BABC",
    "Distribution": {
        "Id": "E1HVIAU7U12ABC",
        "ARN": "arn:aws:cloudfront::123456789012:distribution/E1HVIAU7U12ABC",
        "Status": "InProgress",
        "LastModifiedTime": "2025-07-10T20:33:31.117000+00:00",
        "InProgressInvalidationBatches": 0,
        "DomainName": "example.com",
```

```
        "ActiveTrustedSigners": {
            "Enabled": false,
            "Quantity": 0
        },
        "ActiveTrustedKeyGroups": {
            "Enabled": false,
            "Quantity": 0
        },
        "DistributionConfig": {
            "CallerReference": "cli-example-with-cert",
            "DefaultRootObject": "index.html",
            "Origins": {
                "Quantity": 1,
                "Items": [
                    {
                        "Id": "amzn-s3-demo-bucket.s3.us-east-1.amazonaws.com",
                        "DomainName": "amzn-s3-demo-bucket.s3.us-
east-1.amazonaws.com",
                        "OriginPath": "/{{tenantName}}",
                        "CustomHeaders": {
                            "Quantity": 0
                        },
                        "S3OriginConfig": {
                            "OriginAccessIdentity": ""
                        },
                        "ConnectionAttempts": 3,
                        "ConnectionTimeout": 10,
                        "OriginShield": {
                            "Enabled": false
                        },
                        "OriginAccessControlId": ""
                    }
                ]
            },
            "OriginGroups": {
                "Quantity": 0
            },
            "DefaultCacheBehavior": {
                "TargetOriginId": "amzn-s3-demo-bucket.s3.us-
east-1.amazonaws.com",
                "TrustedKeyGroups": {
                    "Enabled": false,
                    "Quantity": 0
                },
```

```
                "ViewerProtocolPolicy": "allow-all",
                "AllowedMethods": {
                    "Quantity": 2,
                    "Items": ["HEAD", "GET"],
                    "CachedMethods": {
                        "Quantity": 2,
                        "Items": ["HEAD", "GET"]
                    }
                },
                "Compress": false,
                "LambdaFunctionAssociations": {
                    "Quantity": 0
                },
                "FunctionAssociations": {
                    "Quantity": 0
                },
                "FieldLevelEncryptionId": "",
                "CachePolicyId": "658327ea-f89d-4fab-a63d-7e88639e5ABC",
                "GrpcConfig": {
                    "Enabled": false
                }
            },
            "CacheBehaviors": {
                "Quantity": 0
            },
            "CustomErrorResponses": {
                "Quantity": 0
            },
            "Comment": "CLI example distribution",
            "Logging": {
                "Enabled": false,
                "IncludeCookies": false,
                "Bucket": "",
                "Prefix": ""
            },
            "Enabled": true,
            "ViewerCertificate": {
                "CloudFrontDefaultCertificate": false,
                "ACMCertificateArn": "arn:aws:acm:us-
east-1:123456789012:certificate/1954f095-11b6-4daf-9952-0c308a00abc",
                "SSLSupportMethod": "sni-only",
                "MinimumProtocolVersion": "TLSv1.2_2021",
                "Certificate": "arn:aws:acm:us-
east-1:123456789012:certificate/1954f095-11b6-4daf-9952-0c308a00abc",
```

```
                "CertificateSource": "acm"
            },
            "Restrictions": {
                "GeoRestriction": {
                    "RestrictionType": "none",
                    "Quantity": 0
                }
            },
            "WebACLId": "",
            "HttpVersion": "http2",
            "TenantConfig": {
                "ParameterDefinitions": [
                    {
                        "Name": "tenantName",
                        "Definition": {
                            "StringSchema": {
                                "Comment": "tenantName parameter",
                                "DefaultValue": "root",
                                "Required": false
                            }
                        }
                    }
                ]
            },
            "ConnectionMode": "tenant-only"
        }
    }
}
```

Weitere Informationen finden Sie unter Arbeiten mit Distributionen im Amazon CloudFront Developer Guide.

Beispiel 4: So erstellen Sie eine CloudFront Multi-Tenant-Distribution ohne Zertifikat

Im folgenden `create-distribution` Beispiel wird eine CloudFront Distribution mit Mehrmandantenunterstützung, aber ohne TLS-Zertifikat erstellt.

```
aws cloudfront create-distribution \
    --distribution-config file://dist-config.json
```

Inhalt von `dist-config.json`:

```
{
```

```
    "CallerReference": "cli-example",
    "Comment": "CLI example distribution",
    "DefaultRootObject": "index.html",
    "Origins": {
        "Quantity": 1,
        "Items": [
            {
                "Id": "amzn-s3-demo-bucket.s3.us-east-1.amazonaws.com",
                "DomainName": "amzn-s3-demo-bucket.s3.us-east-1.amazonaws.com",
                "OriginPath": "/{{tenantName}}",
                "CustomHeaders": {
                    "Quantity": 0
                },
                "S3OriginConfig": {
                    "OriginAccessIdentity": ""
                }
            }
        ]
    },
    "DefaultCacheBehavior": {
        "TargetOriginId": "amzn-s3-demo-bucket.s3.us-east-1.amazonaws.com",
        "CachePolicyId": "658327ea-f89d-4fab-a63d-7e88639e5ABC",
        "ViewerProtocolPolicy": "allow-all",
        "AllowedMethods": {
            "Quantity": 2,
            "Items": [
                "HEAD",
                "GET"
            ],
            "CachedMethods": {
                "Quantity": 2,
                "Items": [
                    "HEAD",
                    "GET"
                ]
            }
        }
    },
    "Enabled": true,
    "HttpVersion": "http2",
    "ConnectionMode": "tenant-only",
    "TenantConfig": {
        "ParameterDefinitions": [
            {
```

```
                "Name": "tenantName",
                "Definition": {
                    "StringSchema": {
                        "Comment": "tenantName parameter",
                        "DefaultValue": "root",
                        "Required": false
                    }
                }
            }
        ]
    }
}
```

Ausgabe:

```
{
    "Location": "https://cloudfront.amazonaws.com/2020-05-31/distribution/
E2GJ5J9QN12ABC",
    "ETag": "E37YLVVQIABC",
    "Distribution": {
        "Id": "E2GJ5J9QNABC",
        "ARN": "arn:aws:cloudfront::123456789012:distribution/E2GJ5J9QN12ABC",
        "Status": "InProgress",
        "LastModifiedTime": "2025-07-10T20:35:20.565000+00:00",
        "InProgressInvalidationBatches": 0,
        "DomainName": "example.com",
        "ActiveTrustedSigners": {
            "Enabled": false,
            "Quantity": 0
        },
        "ActiveTrustedKeyGroups": {
            "Enabled": false,
            "Quantity": 0
        },
        "DistributionConfig": {
            "CallerReference": "cli-example-no-cert",
            "DefaultRootObject": "index.html",
            "Origins": {
                "Quantity": 1,
                "Items": [
                    {
                        "Id": "amzn-s3-demo-bucket.s3.us-east-1.amazonaws.com",
```

```
                        "DomainName": "amzn-s3-demo-bucket.s3.us-
east-1.amazonaws.com",
                        "OriginPath": "/{{tenantName}}",
                        "CustomHeaders": {
                            "Quantity": 0
                        },
                        "S3OriginConfig": {
                            "OriginAccessIdentity": ""
                        },
                        "ConnectionAttempts": 3,
                        "ConnectionTimeout": 10,
                        "OriginShield": {
                            "Enabled": false
                        },
                        "OriginAccessControlId": ""
                    }
                ]
            },
            "OriginGroups": {
                "Quantity": 0
            },
            "DefaultCacheBehavior": {
                "TargetOriginId": "amzn-s3-demo-bucket.s3.us-
east-1.amazonaws.com",
                "TrustedKeyGroups": {
                    "Enabled": false,
                    "Quantity": 0
                },
                "ViewerProtocolPolicy": "allow-all",
                "AllowedMethods": {
                    "Quantity": 2,
                    "Items": [
                        "HEAD",
                        "GET"
                    ],
                    "CachedMethods": {
                        "Quantity": 2,
                        "Items": [
                            "HEAD",
                            "GET"
                        ]
                    }
                },
                "Compress": false,
```

```
                "LambdaFunctionAssociations": {
                    "Quantity": 0
                },
                "FunctionAssociations": {
                    "Quantity": 0
                },
                "FieldLevelEncryptionId": "",
                "CachePolicyId": "658327ea-f89d-4fab-a63d-7e88639e5ABC",
                "GrpcConfig": {
                    "Enabled": false
                }
            },
            "CacheBehaviors": {
                "Quantity": 0
            },
            "CustomErrorResponses": {
                "Quantity": 0
            },
            "Comment": "CLI example distribution",
            "Logging": {
                "Enabled": false,
                "IncludeCookies": false,
                "Bucket": "",
                "Prefix": ""
            },
            "Enabled": true,
            "ViewerCertificate": {
                "CloudFrontDefaultCertificate": true,
                "SSLSupportMethod": "sni-only",
                "MinimumProtocolVersion": "TLSv1",
                "CertificateSource": "cloudfront"
            },
            "Restrictions": {
                "GeoRestriction": {
                    "RestrictionType": "none",
                    "Quantity": 0
                }
            },
            "WebACLId": "",
            "HttpVersion": "http2",
            "TenantConfig": {
                "ParameterDefinitions": [
                    {
                        "Name": "tenantName",
```

```
                        "Definition": {
                            "StringSchema": {
                                "Comment": "tenantName parameter",
                                "DefaultValue": "root",
                                "Required": false
                            }
                        }
                    }
                ]
            },
            "ConnectionMode": "tenant-only"
        }
    }
}
```

Weitere Informationen finden Sie unter Distributionen konfigurieren im Amazon CloudFront Developer Guide.

- Einzelheiten zur API finden Sie CreateDistributionin der AWS CLI Befehlsreferenz.

Java

SDK für Java 2.x

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS-Code-Beispiel- einrichten und ausführen.

Im folgenden Beispiel wird ein Bucket von Amazon Simple Storage Service(Amazon S3) als Inhaltsquelle verwendet.

Nach dem Erstellen der Distribution erstellt der Code eine Option, CloudFrontWaiterum zu warten, bis die Distribution bereitgestellt ist, bevor die Distribution zurückgegeben wird.

```
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import software.amazon.awssdk.core.internal.waiters.ResponseOrException;
import software.amazon.awssdk.services.cloudfront.CloudFrontClient;
```

```
import
 software.amazon.awssdk.services.cloudfront.model.CreateDistributionResponse;
import software.amazon.awssdk.services.cloudfront.model.Distribution;
import software.amazon.awssdk.services.cloudfront.model.GetDistributionResponse;
import software.amazon.awssdk.services.cloudfront.model.ItemSelection;
import software.amazon.awssdk.services.cloudfront.model.Method;
import software.amazon.awssdk.services.cloudfront.model.ViewerProtocolPolicy;
import software.amazon.awssdk.services.cloudfront.waiters.CloudFrontWaiter;
import software.amazon.awssdk.services.s3.S3Client;

import java.time.Instant;

public class CreateDistribution {

        private static final Logger logger =
 LoggerFactory.getLogger(CreateDistribution.class);

        public static Distribution createDistribution(CloudFrontClient
 cloudFrontClient, S3Client s3Client,
                        final String bucketName, final String keyGroupId, final
 String originAccessControlId) {

                final String region = s3Client.headBucket(b ->
 b.bucket(bucketName)).sdkHttpResponse().headers()
                                .get("x-amz-bucket-region").get(0);
                final String originDomain = bucketName + ".s3." + region +
 ".amazonaws.com";
                String originId = originDomain; // Use the originDomain value for
 the originId.

                // The service API requires some deprecated methods, such as
                // DefaultCacheBehavior.Builder#minTTL and #forwardedValue.
                CreateDistributionResponse createDistResponse =
 cloudFrontClient.createDistribution(builder -> builder
                                .distributionConfig(b1 -> b1
                                                .origins(b2 -> b2
                                                                .quantity(1)
                                                                .items(b3 -> b3

 .domainName(originDomain)

 .id(originId)

 .s3OriginConfig(builder4 -> builder4
```

```
                .originAccessIdentity(
                                    ""))
.originAccessControlId(
                originAccessControlId)))
                                                .defaultCacheBehavior(b2 -> b2
.viewerProtocolPolicy(ViewerProtocolPolicy.ALLOW_ALL)
.targetOriginId(originId)
                                                                    .minTTL(200L)
.forwardedValues(b5 -> b5
.cookies(cp -> cp
                .forward(ItemSelection.NONE))
.queryString(true))
.trustedKeyGroups(b3 -> b3
.quantity(1)
.items(keyGroupId)
.enabled(true))
.allowedMethods(b4 -> b4
.quantity(2)
.items(Method.HEAD, Method.GET)
.cachedMethods(b5 -> b5
                .quantity(2)
                .items(Method.HEAD,
                                    Method.GET))))
```

```
                                                .cacheBehaviors(b -> b
                                                                .quantity(1)
                                                                .items(b2 -> b2

.pathPattern("/index.html")

.viewerProtocolPolicy(

              ViewerProtocolPolicy.ALLOW_ALL)

.targetOriginId(originId)

.trustedKeyGroups(b3 -> b3

              .quantity(1)

              .items(keyGroupId)

              .enabled(true))

.minTTL(200L)

.forwardedValues(b4 -> b4

              .cookies(cp -> cp

                              .forward(ItemSelection.NONE))

              .queryString(true))

.allowedMethods(b5 -> b5.quantity(2)

              .items(Method.HEAD,

                             Method.GET)

              .cachedMethods(b6 -> b6

                             .quantity(2)

                             .items(Method.HEAD,

                                                Method.GET)))))
                                                 .enabled(true)
```

```
                                                .comment("Distribution built with
 java")

 .callerReference(Instant.now().toString())));

                final Distribution distribution =
 createDistResponse.distribution();
                logger.info("Distribution created. DomainName: [{}]  Id: [{}]",
 distribution.domainName(),
                                distribution.id());
                logger.info("Waiting for distribution to be deployed ...");
                try (CloudFrontWaiter cfWaiter =
 CloudFrontWaiter.builder().client(cloudFrontClient).build()) {
                        ResponseOrException<GetDistributionResponse>
 responseOrException = cfWaiter
                                        .waitUntilDistributionDeployed(builder ->
 builder.id(distribution.id()))
                                        .matched();
                        responseOrException.response()
                                        .orElseThrow(() -> new
 RuntimeException("Distribution not created"));
                        logger.info("Distribution deployed. DomainName: [{}]  Id:
 [{}]", distribution.domainName(),
                                        distribution.id());
                }
                return distribution;
        }
}
```

- Einzelheiten zur API finden Sie CreateDistributionunter AWS SDK for Java 2.x API-Referenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: Erstellt eine CloudFront Basisdistribution, die mit Protokollierung und Caching konfiguriert ist.

```
$origin = New-Object Amazon.CloudFront.Model.Origin
$origin.DomainName = "amzn-s3-demo-bucket.s3.amazonaws.com"
```

```
$origin.Id = "UniqueOrigin1"
$origin.S3OriginConfig = New-Object Amazon.CloudFront.Model.S3OriginConfig
$origin.S3OriginConfig.OriginAccessIdentity = ""
New-CFDistribution `
      -DistributionConfig_Enabled $true `
      -DistributionConfig_Comment "Test distribution" `
      -Origins_Item $origin `
      -Origins_Quantity 1 `
      -Logging_Enabled $true `
      -Logging_IncludeCookie $true `
      -Logging_Bucket amzn-s3-demo-logging-bucket.s3.amazonaws.com `
      -Logging_Prefix "help/" `
      -DistributionConfig_CallerReference Client1 `
      -DistributionConfig_DefaultRootObject index.html `
      -DefaultCacheBehavior_TargetOriginId $origin.Id `
      -ForwardedValues_QueryString $true `
      -Cookies_Forward all `
      -WhitelistedNames_Quantity 0 `
      -TrustedSigners_Enabled $false `
      -TrustedSigners_Quantity 0 `
      -DefaultCacheBehavior_ViewerProtocolPolicy allow-all `
      -DefaultCacheBehavior_MinTTL 1000 `
      -DistributionConfig_PriceClass "PriceClass_All" `
      -CacheBehaviors_Quantity 0 `
      -Aliases_Quantity 0
```

- Einzelheiten zur API finden Sie unter CreateDistribution AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: Erstellt eine CloudFront Basisdistribution, die mit Protokollierung und Caching konfiguriert ist.

```
$origin = New-Object Amazon.CloudFront.Model.Origin
$origin.DomainName = "amzn-s3-demo-bucket.s3.amazonaws.com"
$origin.Id = "UniqueOrigin1"
$origin.S3OriginConfig = New-Object Amazon.CloudFront.Model.S3OriginConfig
$origin.S3OriginConfig.OriginAccessIdentity = ""
New-CFDistribution `
      -DistributionConfig_Enabled $true `
      -DistributionConfig_Comment "Test distribution" `
      -Origins_Item $origin `
      -Origins_Quantity 1 `
```

```
    -Logging_Enabled $true `
    -Logging_IncludeCookie $true `
    -Logging_Bucket amzn-s3-demo-logging-bucket.s3.amazonaws.com `
    -Logging_Prefix "help/" `
    -DistributionConfig_CallerReference Client1 `
    -DistributionConfig_DefaultRootObject index.html `
    -DefaultCacheBehavior_TargetOriginId $origin.Id `
    -ForwardedValues_QueryString $true `
    -Cookies_Forward all `
    -WhitelistedNames_Quantity 0 `
    -TrustedSigners_Enabled $false `
    -TrustedSigners_Quantity 0 `
    -DefaultCacheBehavior_ViewerProtocolPolicy allow-all `
    -DefaultCacheBehavior_MinTTL 1000 `
    -DistributionConfig_PriceClass "PriceClass_All" `
    -CacheBehaviors_Quantity 0 `
    -Aliases_Quantity 0
```

- Einzelheiten zur API finden Sie unter CreateDistribution AWS -Tools für PowerShellCmdlet-Referenz (V5).

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwendung CloudFront mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **`CreateFunction`**mit einem AWS SDK

Das folgende Codebeispiel zeigt, wie es verwendet wird`CreateFunction`.

Java

SDK für Java 2.x

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS-Code-Beispiel- einrichten und ausführen.

```
import software.amazon.awssdk.core.SdkBytes;
import software.amazon.awssdk.regions.Region;
```

```
import software.amazon.awssdk.services.cloudfront.CloudFrontClient;
import software.amazon.awssdk.services.cloudfront.model.CloudFrontException;
import software.amazon.awssdk.services.cloudfront.model.CreateFunctionRequest;
import software.amazon.awssdk.services.cloudfront.model.CreateFunctionResponse;
import software.amazon.awssdk.services.cloudfront.model.FunctionConfig;
import software.amazon.awssdk.services.cloudfront.model.FunctionRuntime;
import java.io.InputStream;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class CreateFunction {

    public static void main(String[] args) {
        final String usage = """

                Usage:
                    <functionName> <filePath>

                Where:
                    functionName - The name of the function to create.\s
                    filePath - The path to a file that contains the application
 logic for the function.\s
                """;

        if (args.length != 2) {
            System.out.println(usage);
            System.exit(1);
        }

        String functionName = args[0];
        String filePath = args[1];
        CloudFrontClient cloudFrontClient = CloudFrontClient.builder()
                .region(Region.AWS_GLOBAL)
                .build();

        String funArn = createNewFunction(cloudFrontClient, functionName,
 filePath);
```

```
        System.out.println("The function ARN is " + funArn);
        cloudFrontClient.close();
    }

    public static String createNewFunction(CloudFrontClient cloudFrontClient,
 String functionName, String filePath) {
        try {
            InputStream fileIs =
 CreateFunction.class.getClassLoader().getResourceAsStream(filePath);
            SdkBytes functionCode = SdkBytes.fromInputStream(fileIs);

            FunctionConfig config = FunctionConfig.builder()
                    .comment("Created by using the CloudFront Java API")
                    .runtime(FunctionRuntime.CLOUDFRONT_JS_1_0)
                    .build();

            CreateFunctionRequest functionRequest =
 CreateFunctionRequest.builder()
                    .name(functionName)
                    .functionCode(functionCode)
                    .functionConfig(config)
                    .build();

            CreateFunctionResponse response =
 cloudFrontClient.createFunction(functionRequest);
            return response.functionSummary().functionMetadata().functionARN();

        } catch (CloudFrontException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
        return "";
    }
}
```

- Einzelheiten zur API finden Sie CreateFunctionin der AWS SDK for Java 2.x API-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unterVerwendung CloudFront mit einem AWS SDK. Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung von **`CreateInvalidation`** mit einer CLI

Die folgenden Code-Beispiele zeigen, wie `CreateInvalidation` verwendet wird.

CLI

AWS CLI

Um eine Invalidierung für eine CloudFront Distribution zu erstellen

Das folgende `create-invalidation` Beispiel erstellt eine Invalidierung für die angegebenen Dateien in der angegebenen CloudFront Distribution:

```
aws cloudfront create-invalidation \
    --distribution-id EDFDVBD6EXAMPLE \
    --paths "/example-path/example-file.jpg" "/example-path/example-file2.png"
```

Ausgabe:

```
{
    "Location": "https://cloudfront.amazonaws.com/2019-03-26/distribution/
EDFDVBD6EXAMPLE/invalidation/I1JLWSDAP8FU89",
    "Invalidation": {
        "Id": "I1JLWSDAP8FU89",
        "Status": "InProgress",
        "CreateTime": "2019-12-05T18:24:51.407Z",
        "InvalidationBatch": {
            "Paths": {
                "Quantity": 2,
                "Items": [
                    "/example-path/example-file2.png",
                    "/example-path/example-file.jpg"
                ]
            },
            "CallerReference": "cli-1575570291-670203"
        }
    }
}
```

Im vorherigen Beispiel generierte die AWS CLI automatisch ein zufälliges Ergebnis`CallerReference`. Wenn Sie Ihre eigene `CallerReference` angeben oder die Invalidierungsparameter nicht als Befehlszeilenargumente übergeben möchten, können Sie

eine JSON-Datei verwenden. Im folgende Beispiel wird eine Invalidierung für zwei Dateien erstellt, indem die Invalidierungsparameter in einer JSON-Datei mit dem Namen `inv-batch.json` angegeben werden.

```
aws cloudfront create-invalidation \
    --distribution-id EDFDVBD6EXAMPLE \
    --invalidation-batch file://inv-batch.json
```

Inhalt von `inv-batch.json`:

```
{
    "Paths": {
        "Quantity": 2,
        "Items": [
            "/example-path/example-file.jpg",
            "/example-path/example-file2.png"
        ]
    },
    "CallerReference": "cli-example"
}
```

Ausgabe:

```
{
    "Location": "https://cloudfront.amazonaws.com/2019-03-26/distribution/
EDFDVBD6EXAMPLE/invalidation/I2J0I21PCUYOIK",
    "Invalidation": {
        "Id": "I2J0I21PCUYOIK",
        "Status": "InProgress",
        "CreateTime": "2019-12-05T18:40:49.413Z",
        "InvalidationBatch": {
            "Paths": {
                "Quantity": 2,
                "Items": [
                    "/example-path/example-file.jpg",
                    "/example-path/example-file2.png"
                ]
            },
            "CallerReference": "cli-example"
        }
    }
}
```

- Einzelheiten zur API finden Sie CreateInvalidationin der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: In diesem Beispiel wird eine neue Invalidierung für eine Distribution mit der ID EXAMPLENSTXAXE erstellt. Das CallerReference ist eine vom Benutzer gewählte eindeutige ID. In diesem Fall wird ein Zeitstempel verwendet, der den 15. Mai 2019 um 9:00 Uhr darstellt. Die Variable $Paths speichert drei Pfade zu Bild- und Mediendateien, die der Benutzer nicht als Teil des Caches der Distribution haben möchte. Der Wert des Paths_Quantity Parameters - ist die Gesamtzahl der im Paths_Item Parameter - angegebenen Pfade.

```
$Paths = "/images/*.gif", "/images/image1.jpg", "/videos/*.mp4"
New-CFInvalidation -DistributionId "EXAMPLENSTXAXE" -
InvalidationBatch_CallerReference 20190515090000 -Paths_Item $Paths -
Paths_Quantity 3
```

Ausgabe:

```
Invalidation                          Location

------------                          --------

Amazon.CloudFront.Model.Invalidation https://cloudfront.amazonaws.com/2018-11-05/
distribution/EXAMPLENSTXAXE/invalidation/EXAMPLE8NOK9H
```

- Einzelheiten zur API finden Sie unter CreateInvalidation AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: In diesem Beispiel wird eine neue Invalidierung für eine Distribution mit der ID EXAMPLENSTXAXE erstellt. Das CallerReference ist eine vom Benutzer gewählte eindeutige ID. In diesem Fall wird ein Zeitstempel verwendet, der den 15. Mai 2019 um 9:00 Uhr darstellt. Die Variable $Paths speichert drei Pfade zu Bild- und Mediendateien, die der Benutzer nicht als Teil des Caches der Distribution haben möchte. Der Wert des Paths_Quantity Parameters - ist die Gesamtzahl der im Paths_Item Parameter - angegebenen Pfade.

```
$Paths = "/images/*.gif", "/images/image1.jpg", "/videos/*.mp4"
```

```
New-CFInvalidation -DistributionId "EXAMPLENSTXAXE" -
InvalidationBatch_CallerReference 20190515090000 -Paths_Item $Paths -
Paths_Quantity 3
```

Ausgabe:

```
Invalidation                          Location

------------                          --------

Amazon.CloudFront.Model.Invalidation https://cloudfront.amazonaws.com/2018-11-05/
distribution/EXAMPLENSTXAXE/invalidation/EXAMPLE8NOK9H
```

- Einzelheiten zur API finden Sie unter CreateInvalidation AWS -Tools für PowerShellCmdlet-Referenz (V5).

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwendung CloudFront mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **`CreateKeyGroup`**mit einem AWS SDK

Das folgende Codebeispiel zeigt, wie es verwendet wird`CreateKeyGroup`.

Java

SDK für Java 2.x

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS-Code-Beispiel- einrichten und ausführen.

Eine Schlüsselgruppe erfordert mindestens einen öffentlichen Schlüssel, der zur Überprüfung signierter URLs oder Cookies verwendet wird.

```
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import software.amazon.awssdk.services.cloudfront.CloudFrontClient;
```

```
import java.util.UUID;

public class CreateKeyGroup {
    private static final Logger logger =
 LoggerFactory.getLogger(CreateKeyGroup.class);

    public static String createKeyGroup(CloudFrontClient cloudFrontClient, String
 publicKeyId) {
        String keyGroupId = cloudFrontClient.createKeyGroup(b ->
 b.keyGroupConfig(c -> c
                .items(publicKeyId)
                .name("JavaKeyGroup" + UUID.randomUUID()))
                .keyGroup().id();
        logger.info("KeyGroup created with ID: [{}]", keyGroupId);
        return keyGroupId;
    }
}
```

- Einzelheiten zur API finden Sie CreateKeyGroupin der AWS SDK for Java 2.x API-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unterVerwendung CloudFront mit einem AWS SDK. Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **`CreatePublicKey`**mit einem AWS SDK oder CLI

Die folgenden Code-Beispiele zeigen, wie `CreatePublicKey` verwendet wird.

CLI

AWS CLI

Um einen CloudFront öffentlichen Schlüssel zu erstellen

Im folgenden Beispiel wird ein CloudFront öffentlicher Schlüssel erstellt, indem die Parameter in einer JSON-Datei mit dem Namen bereitgestellt `pub-key-config.json` werden. Bevor Sie diesen Befehl verwenden können, benötigen Sie einen PEM-encoded öffentlichen Schlüssel. Weitere Informationen finden Sie unter Create an RSA Key Pair im Amazon CloudFront Developer Guide.

```
aws cloudfront create-public-key \
```

```
    --public-key-config file://pub-key-config.json
```

Die Datei `pub-key-config.json` ist ein JSON-Dokument im aktuellen Ordner, das Folgendes enthält. Beachten Sie, dass der öffentliche Schlüssel im PEM-Format verschlüsselt ist.

```
{
    "CallerReference": "cli-example",
    "Name": "ExampleKey",
    "EncodedKey": "-----BEGIN PUBLIC KEY-----
\nMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAxPMbCA2Ks0lnd7IR+3pw
\nwd3H/7jPGwj8bLUmore7bX+oeGpZ6QmLAe/1UOWcmZX2u70dYcSIzB1ofZtcn4cJ
\nenHBAzO3ohBY/L1tQGJfS2A+omnN6H16VZE1JCK8XSJyfze7MDLcUyHZETdxuvRb
\nA9X343/vMAuQPnhinFJ8Wdy8YBXSPpy7r95ylUQd9LfYTBzVZYG2tSesplcOkjM3\n2Uu
+oMWxQAw1NINnSLPinMVsutJy6ZqlV3McWNWe4T+STGtWhrPNqJEn45sIcCx4\nq
+kGZ2NQ0FyIyT2eiLKOX5Rgb/a36E/aMk4VoDsaenBQgG7WLTnstb9sr7MIhS6A\nrwIDAQAB\n-----
END PUBLIC KEY-----\n",
    "Comment": "example public key"
}
```

Ausgabe:

```
{
    "Location": "https://cloudfront.amazonaws.com/2019-03-26/public-key/
KDFB19YGCR002",
    "ETag": "E2QWRUHEXAMPLE",
    "PublicKey": {
        "Id": "KDFB19YGCR002",
        "CreatedTime": "2019-12-05T18:51:43.781Z",
        "PublicKeyConfig": {
            "CallerReference": "cli-example",
            "Name": "ExampleKey",
            "EncodedKey": "-----BEGIN PUBLIC KEY-----
\nMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAxPMbCA2Ks0lnd7IR+3pw
\nwd3H/7jPGwj8bLUmore7bX+oeGpZ6QmLAe/1UOWcmZX2u70dYcSIzB1ofZtcn4cJ
\nenHBAzO3ohBY/L1tQGJfS2A+omnN6H16VZE1JCK8XSJyfze7MDLcUyHZETdxuvRb
\nA9X343/vMAuQPnhinFJ8Wdy8YBXSPpy7r95ylUQd9LfYTBzVZYG2tSesplcOkjM3\n2Uu
+oMWxQAw1NINnSLPinMVsutJy6ZqlV3McWNWe4T+STGtWhrPNqJEn45sIcCx4\nq
+kGZ2NQ0FyIyT2eiLKOX5Rgb/a36E/aMk4VoDsaenBQgG7WLTnstb9sr7MIhS6A\nrwIDAQAB\n-----
END PUBLIC KEY-----\n",
            "Comment": "example public key"
        }
```

```
    }
}
```

- Einzelheiten zur API finden Sie unter CreatePublicKey AWS CLIBefehlsreferenz.

Java

SDK für Java 2.x

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS-Code-Beispiel- einrichten und ausführen.

Das folgende Codebeispiel liest einen öffentlichen Schlüssel ein und lädt ihn auf Amazon CloudFront hoch.

```
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import software.amazon.awssdk.services.cloudfront.CloudFrontClient;
import software.amazon.awssdk.services.cloudfront.model.CreatePublicKeyResponse;
import software.amazon.awssdk.utils.IoUtils;

import java.io.IOException;
import java.io.InputStream;
import java.util.UUID;

public class CreatePublicKey {
    private static final Logger logger =
 LoggerFactory.getLogger(CreatePublicKey.class);

    public static String createPublicKey(CloudFrontClient cloudFrontClient,
 String publicKeyFileName) {
        try (InputStream is =
 CreatePublicKey.class.getClassLoader().getResourceAsStream(publicKeyFileName)) {
            String publicKeyString = IoUtils.toUtf8String(is);
            CreatePublicKeyResponse createPublicKeyResponse = cloudFrontClient
                    .createPublicKey(b -> b.publicKeyConfig(c -> c
                            .name("JavaCreatedPublicKey" + UUID.randomUUID())
                            .encodedKey(publicKeyString)
                            .callerReference(UUID.randomUUID().toString())));
```

```
            String createdPublicKeyId = createPublicKeyResponse.publicKey().id();
            logger.info("Public key created with id: [{}]", createdPublicKeyId);
            return createdPublicKeyId;

        } catch (IOException e) {
            throw new RuntimeException(e);
        }
    }
}
```

- Einzelheiten zur API finden Sie CreatePublicKeyin der AWS SDK for Java 2.x API-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unterVerwendung CloudFront mit einem AWS SDK. Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **`DeleteDistribution`**mit einem AWS SDK oder CLI

Die folgenden Code-Beispiele zeigen, wie `DeleteDistribution` verwendet wird.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- Beginnen Sie mit einem grundlegenden Netzwerk zur Verteilung von Inhalten

CLI

AWS CLI

Um eine CloudFront Distribution zu löschen

Im folgenden Beispiel wird die CloudFront Distribution mit der ID `EDFDVBD6EXAMPLE` gelöscht. Bevor Sie eine Distribution löschen können, müssen Sie sie deaktivieren. Verwenden Sie zum Deaktivieren einer Distribution den Befehl „update-distribution“. Weitere Informationen finden Sie in den Beispielen zu „update-distribution“.

Wenn eine Distribution deaktiviert ist, können Sie sie löschen. Um eine Distribution zu löschen, müssen Sie die `--if-match`-Option verwenden, um den `ETag` der Distribution anzugeben. Um den `ETag` zu erhalten, verwenden Sie bitte den Befehl „get-Distribution“ oder „get-Distribution-Config“.

```
aws cloudfront delete-distribution \
    --id EDFDVBD6EXAMPLE \
    --if-match E2QWRUHEXAMPLE
```

Bei erfolgreicher Ausführung gibt dieser Befehl keine Ausgabe aus.

- Einzelheiten zur API finden Sie unter DeleteDistribution AWS CLIBefehlsreferenz.

Java

SDK für Java 2.x

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS-Code-Beispiel- einrichten und ausführen.

Das folgende Codebeispiel aktualisiert eine Distribution auf deaktiviert, verwendet einen Waiter, der auf die Bereitstellung der Änderung wartet, und löscht anschließend die Distribution.

```
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import software.amazon.awssdk.core.internal.waiters.ResponseOrException;
import software.amazon.awssdk.services.cloudfront.CloudFrontClient;
import
 software.amazon.awssdk.services.cloudfront.model.DeleteDistributionResponse;
import software.amazon.awssdk.services.cloudfront.model.DistributionConfig;
import software.amazon.awssdk.services.cloudfront.model.GetDistributionResponse;
import software.amazon.awssdk.services.cloudfront.waiters.CloudFrontWaiter;

public class DeleteDistribution {
        private static final Logger logger =
 LoggerFactory.getLogger(DeleteDistribution.class);

        public static void deleteDistribution(final CloudFrontClient
 cloudFrontClient, final String distributionId) {
                // First, disable the distribution by updating it.
                GetDistributionResponse response =
 cloudFrontClient.getDistribution(b -> b
```

```
                                .id(distributionId));
                String etag = response.eTag();
                DistributionConfig distConfig =
response.distribution().distributionConfig();

                cloudFrontClient.updateDistribution(builder -> builder
                                .id(distributionId)
                                .distributionConfig(builder1 -> builder1

.cacheBehaviors(distConfig.cacheBehaviors())

.defaultCacheBehavior(distConfig.defaultCacheBehavior())
                                                        .enabled(false)
                                                        .origins(distConfig.origins())
                                                        .comment(distConfig.comment())

.callerReference(distConfig.callerReference())

.defaultCacheBehavior(distConfig.defaultCacheBehavior())

.priceClass(distConfig.priceClass())
                                                        .aliases(distConfig.aliases())
                                                        .logging(distConfig.logging())

.defaultRootObject(distConfig.defaultRootObject())

.customErrorResponses(distConfig.customErrorResponses())

.httpVersion(distConfig.httpVersion())

.isIPV6Enabled(distConfig.isIPV6Enabled())

.restrictions(distConfig.restrictions())

.viewerCertificate(distConfig.viewerCertificate())
                                                        .webACLId(distConfig.webACLId())

.originGroups(distConfig.originGroups()))
                                .ifMatch(etag));

                logger.info("Distribution [{}] is DISABLED, waiting for
deployment before deleting ...",
                                distributionId);
                GetDistributionResponse distributionResponse;
```

```
                try (CloudFrontWaiter cfWaiter =
 CloudFrontWaiter.builder().client(cloudFrontClient).build()) {
                        ResponseOrException<GetDistributionResponse>
 responseOrException = cfWaiter
                                        .waitUntilDistributionDeployed(builder ->
 builder.id(distributionId)).matched();
                        distributionResponse = responseOrException.response()
                                        .orElseThrow(() -> new
 RuntimeException("Could not disable distribution"));
                }

                DeleteDistributionResponse deleteDistributionResponse =
 cloudFrontClient
                                .deleteDistribution(builder -> builder
                                                .id(distributionId)

 .ifMatch(distributionResponse.eTag()));
                if (deleteDistributionResponse.sdkHttpResponse().isSuccessful())
 {
                        logger.info("Distribution [{}] DELETED", distributionId);
                }
        }
}
```

- Einzelheiten zur API finden Sie DeleteDistributionin der AWS SDK for Java 2.x API-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unterVerwendung CloudFront mit einem AWS SDK. Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung von **`GetCloudFrontOriginAccessIdentity`** mit einer CLI

Die folgenden Code-Beispiele zeigen, wie `GetCloudFrontOriginAccessIdentity` verwendet wird.

CLI

AWS CLI

Um eine Zugriffsidentität für den CloudFront Ursprung zu erhalten

Im folgenden Beispiel wird die ursprüngliche CloudFront Zugriffsidentität (OAI) mit der ID abgerufen`E74FTE3AEXAMPLE`, einschließlich ihrer `ETag` und der zugehörigen kanonischen S3-ID. Die OAI-ID wird in der Ausgabe der Befehle create-cloud-front-origin-access-identity und list-cloud-front-origin-access-identities zurückgegeben.

```
aws cloudfront get-cloud-front-origin-access-identity --id E74FTE3AEXAMPLE
```

Ausgabe:

```
{
    "ETag": "E2QWRUHEXAMPLE",
    "CloudFrontOriginAccessIdentity": {
        "Id": "E74FTE3AEXAMPLE",
        "S3CanonicalUserId":
 "cd13868f797c227fbea2830611a26fe0a21ba1b826ab4bed9b7771c9aEXAMPLE",
        "CloudFrontOriginAccessIdentityConfig": {
            "CallerReference": "cli-example",
            "Comment": "Example OAI"
        }
    }
}
```

- Einzelheiten zur API finden Sie GetCloudFrontOriginAccessIdentityin AWS CLI der Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: In diesem Beispiel wird eine bestimmte CloudFront Amazon-Ursprungszugriffsidentität zurückgegeben, die durch den Parameter -Id angegeben wird. Obwohl der -Id-Parameter nicht erforderlich ist, werden keine Ergebnisse zurückgegeben, wenn Sie ihn nicht angeben.

```
Get-CFCloudFrontOriginAccessIdentity -Id E3XXXXXXXXXXRT
```

Ausgabe:

```
     CloudFrontOriginAccessIdentityConfig    Id
    S3CanonicalUserId
```

```
     ------------------------------------    --
    -----------------
     Amazon.CloudFront.Model.CloudFrontOr... E3XXXXXXXXXXRT
    4b6e...
```

- Einzelheiten zur API finden Sie unter GetCloudFrontOriginAccessIdentity AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: In diesem Beispiel wird eine bestimmte CloudFront Amazon-Ursprungszugriffsidentität zurückgegeben, die durch den Parameter -Id angegeben wird. Obwohl der -Id-Parameter nicht erforderlich ist, werden keine Ergebnisse zurückgegeben, wenn Sie ihn nicht angeben.

```
Get-CFCloudFrontOriginAccessIdentity -Id E3XXXXXXXXXXRT
```

Ausgabe:

```
     CloudFrontOriginAccessIdentityConfig    Id
    S3CanonicalUserId
     ------------------------------------    --
    -----------------
     Amazon.CloudFront.Model.CloudFrontOr... E3XXXXXXXXXXRT
    4b6e...
```

- Einzelheiten zur API finden Sie unter GetCloudFrontOriginAccessIdentity AWS -Tools für PowerShellCmdlet-Referenz (V5).

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwendung CloudFront mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung von **`GetCloudFrontOriginAccessIdentityConfig`** mit einer CLI

Die folgenden Code-Beispiele zeigen, wie `GetCloudFrontOriginAccessIdentityConfig` verwendet wird.

CLI

AWS CLI

Um eine Konfiguration der CloudFront Origin-Zugriffsidentität zu erhalten

Im folgenden Beispiel werden Metadaten zur ursprünglichen CloudFront Zugriffsidentität (OAI) mit der ID abgerufen`E74FTE3AEXAMPLE`, einschließlich ihrer`ETag`. Die OAI-ID wird in der Ausgabe der Befehle create-cloud-front-origin-access-identity und list-cloud-front-origin-access-identities zurückgegeben.

```
aws cloudfront get-cloud-front-origin-access-identity-config --id E74FTE3AEXAMPLE
```

Ausgabe:

```
{
    "ETag": "E2QWRUHEXAMPLE",
    "CloudFrontOriginAccessIdentityConfig": {
        "CallerReference": "cli-example",
        "Comment": "Example OAI"
    }
}
```

- Einzelheiten zur API finden Sie GetCloudFrontOriginAccessIdentityConfigin der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: In diesem Beispiel werden Konfigurationsinformationen zu einer einzelnen CloudFront Amazon-Ursprungszugriffsidentität zurückgegeben, die durch den Parameter -Id angegeben wird. Fehler treten auf, wenn kein -Id-Parameter angegeben ist.

```
Get-CFCloudFrontOriginAccessIdentityConfig -Id E3XXXXXXXXXXRT
```

Ausgabe:

```
      CallerReference                                       Comment
```

```
     ---------------                                                  -------
     mycallerreference: 2/1/2011 1:16:32 PM                           Caller
reference: 2/1/2011 1:16:32 PM
```

- Einzelheiten zur API finden Sie unter GetCloudFrontOriginAccessIdentityConfig AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: In diesem Beispiel werden Konfigurationsinformationen zu einer einzelnen CloudFront Amazon-Ursprungszugriffsidentität zurückgegeben, die durch den Parameter -Id angegeben wird. Fehler treten auf, wenn kein -Id-Parameter angegeben ist.

```
Get-CFCloudFrontOriginAccessIdentityConfig -Id E3XXXXXXXXXXRT
```

Ausgabe:

```
     CallerReference                                                  Comment
     ---------------                                                  -------
     mycallerreference: 2/1/2011 1:16:32 PM                           Caller
reference: 2/1/2011 1:16:32 PM
```

- Einzelheiten zur API finden Sie unter GetCloudFrontOriginAccessIdentityConfig AWS -Tools für PowerShellCmdlet-Referenz (V5).

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwendung CloudFront mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung von **`GetDistribution`** mit einer CLI

Die folgenden Code-Beispiele zeigen, wie `GetDistribution` verwendet wird.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- Beginnen Sie mit einem grundlegenden Netzwerk zur Verteilung von Inhalten

CLI

AWS CLI

Um eine CloudFront Distribution zu erhalten

Im folgenden `get-distribution` Beispiel wird die CloudFront Distribution mit der ID abgerufen`EDFDVBD6EXAMPLE`, einschließlich ihrer`ETag`. Die Distributions-ID wird in den Befehlen „create-distribution“ und „list-distributions“ zurückgegeben.

```
aws cloudfront get-distribution \
    --id EDFDVBD6EXAMPLE
```

Ausgabe:

```
{
    "ETag": "E2QWRUHEXAMPLE",
    "Distribution": {
        "Id": "EDFDVBD6EXAMPLE",
        "ARN": "arn:aws:cloudfront::123456789012:distribution/EDFDVBD6EXAMPLE",
        "Status": "Deployed",
        "LastModifiedTime": "2019-12-04T23:35:41.433Z",
        "InProgressInvalidationBatches": 0,
        "DomainName": "d111111abcdef8.cloudfront.net",
        "ActiveTrustedSigners": {
            "Enabled": false,
            "Quantity": 0
        },
        "DistributionConfig": {
            "CallerReference": "cli-example",
            "Aliases": {
                "Quantity": 0
            },
            "DefaultRootObject": "index.html",
            "Origins": {
                "Quantity": 1,
                "Items": [
                    {
                        "Id": "amzn-s3-demo-bucket.s3.amazonaws.com-cli-example",
                        "DomainName": "amzn-s3-demo-bucket.s3.amazonaws.com",
                        "OriginPath": "",
                        "CustomHeaders": {
                            "Quantity": 0
```

```
                        },
                        "S3OriginConfig": {
                            "OriginAccessIdentity": ""
                        }
                    }
                ]
            },
            "OriginGroups": {
                "Quantity": 0
            },
            "DefaultCacheBehavior": {
                "TargetOriginId": "amzn-s3-demo-bucket.s3.amazonaws.com-cli-
example",
                "ForwardedValues": {
                    "QueryString": false,
                    "Cookies": {
                        "Forward": "none"
                    },
                    "Headers": {
                        "Quantity": 0
                    },
                    "QueryStringCacheKeys": {
                        "Quantity": 0
                    }
                },
                "TrustedSigners": {
                    "Enabled": false,
                    "Quantity": 0
                },
                "ViewerProtocolPolicy": "allow-all",
                "MinTTL": 0,
                "AllowedMethods": {
                    "Quantity": 2,
                    "Items": [
                        "HEAD",
                        "GET"
                    ],
                    "CachedMethods": {
                        "Quantity": 2,
                        "Items": [
                            "HEAD",
                            "GET"
                        ]
                    }
```

```
                },
                "SmoothStreaming": false,
                "DefaultTTL": 86400,
                "MaxTTL": 31536000,
                "Compress": false,
                "LambdaFunctionAssociations": {
                    "Quantity": 0
                },
                "FieldLevelEncryptionId": ""
            },
            "CacheBehaviors": {
                "Quantity": 0
            },
            "CustomErrorResponses": {
                "Quantity": 0
            },
            "Comment": "",
            "Logging": {
                "Enabled": false,
                "IncludeCookies": false,
                "Bucket": "",
                "Prefix": ""
            },
            "PriceClass": "PriceClass_All",
            "Enabled": true,
            "ViewerCertificate": {
                "CloudFrontDefaultCertificate": true,
                "MinimumProtocolVersion": "TLSv1",
                "CertificateSource": "cloudfront"
            },
            "Restrictions": {
                "GeoRestriction": {
                    "RestrictionType": "none",
                    "Quantity": 0
                }
            },
            "WebACLId": "",
            "HttpVersion": "http2",
            "IsIPV6Enabled": true
        }
    }
}
```

- Einzelheiten zur API finden Sie GetDistributionin der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: Ruft die Informationen für eine bestimmte Distribution ab.

```
Get-CFDistribution -Id EXAMPLE0000ID
```

- Einzelheiten zur API finden Sie unter GetDistribution AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: Ruft die Informationen für eine bestimmte Distribution ab.

```
Get-CFDistribution -Id EXAMPLE0000ID
```

- Einzelheiten zur API finden Sie unter GetDistribution AWS -Tools für PowerShellCmdlet-Referenz (V5).

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwendung CloudFront mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **`GetDistributionConfig`**mit einem AWS SDK oder CLI

Die folgenden Code-Beispiele zeigen, wie `GetDistributionConfig` verwendet wird.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- Beginnen Sie mit einem grundlegenden Netzwerk zur Verteilung von Inhalten

CLI

AWS CLI

Um eine CloudFront Distributionskonfiguration zu erhalten

Im folgenden Beispiel werden Metadaten über die CloudFront Distribution mit der ID abgerufen`EDFDVBD6EXAMPLE`, einschließlich ihrer`ETag`. Die Distributions-ID wird in den Befehlen „create-distribution“ und „list-distributions“ zurückgegeben.

```
aws cloudfront get-distribution-config \
    --id EDFDVBD6EXAMPLE
```

Ausgabe:

```
{
    "ETag": "E2QWRUHEXAMPLE",
    "DistributionConfig": {
        "CallerReference": "cli-example",
        "Aliases": {
            "Quantity": 0
        },
        "DefaultRootObject": "index.html",
        "Origins": {
            "Quantity": 1,
            "Items": [
                {
                    "Id": "amzn-s3-demo-bucket.s3.amazonaws.com-cli-example",
                    "DomainName": "amzn-s3-demo-bucket.s3.amazonaws.com",
                    "OriginPath": "",
                    "CustomHeaders": {
                        "Quantity": 0
                    },
                    "S3OriginConfig": {
                        "OriginAccessIdentity": ""
                    }
                }
            ]
        },
        "OriginGroups": {
            "Quantity": 0
        },
        "DefaultCacheBehavior": {
            "TargetOriginId": "amzn-s3-demo-bucket.s3.amazonaws.com-cli-example",
            "ForwardedValues": {
                "QueryString": false,
                "Cookies": {
                    "Forward": "none"
                },
```

```
                "Headers": {
                    "Quantity": 0
                },
                "QueryStringCacheKeys": {
                    "Quantity": 0
                }
            },
            "TrustedSigners": {
                "Enabled": false,
                "Quantity": 0
            },
            "ViewerProtocolPolicy": "allow-all",
            "MinTTL": 0,
            "AllowedMethods": {
                "Quantity": 2,
                "Items": [
                    "HEAD",
                    "GET"
                ],
                "CachedMethods": {
                    "Quantity": 2,
                    "Items": [
                        "HEAD",
                        "GET"
                    ]
                }
            },
            "SmoothStreaming": false,
            "DefaultTTL": 86400,
            "MaxTTL": 31536000,
            "Compress": false,
            "LambdaFunctionAssociations": {
                "Quantity": 0
            },
            "FieldLevelEncryptionId": ""
        },
        "CacheBehaviors": {
            "Quantity": 0
        },
        "CustomErrorResponses": {
            "Quantity": 0
        },
        "Comment": "",
        "Logging": {
```

```
            "Enabled": false,
            "IncludeCookies": false,
            "Bucket": "",
            "Prefix": ""
        },
        "PriceClass": "PriceClass_All",
        "Enabled": true,
        "ViewerCertificate": {
            "CloudFrontDefaultCertificate": true,
            "MinimumProtocolVersion": "TLSv1",
            "CertificateSource": "cloudfront"
        },
        "Restrictions": {
            "GeoRestriction": {
                "RestrictionType": "none",
                "Quantity": 0
            }
        },
        "WebACLId": "",
        "HttpVersion": "http2",
        "IsIPV6Enabled": true
    }
}
```

- Einzelheiten zur API finden Sie GetDistributionConfigin der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: Ruft die Konfiguration für eine bestimmte Distribution ab.

```
Get-CFDistributionConfig -Id EXAMPLE0000ID
```

- Einzelheiten zur API finden Sie unter GetDistributionConfig AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: Ruft die Konfiguration für eine bestimmte Distribution ab.

```
Get-CFDistributionConfig -Id EXAMPLE0000ID
```

- Einzelheiten zur API finden Sie unter GetDistributionConfig AWS -Tools für PowerShellCmdlet-Referenz (V5).

Python

SDK für Python (Boto3)

> **Note**
>
> Es gibt noch mehr dazu. GitHub Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS-Code-Beispiel- einrichten und ausführen.

```
class CloudFrontWrapper:
    """Encapsulates Amazon CloudFront operations."""

    def __init__(self, cloudfront_client):
        """
        :param cloudfront_client: A Boto3 CloudFront client
        """
        self.cloudfront_client = cloudfront_client


    def update_distribution(self):
        distribution_id = input(
            "This script updates the comment for a CloudFront distribution.\n"
            "Enter a CloudFront distribution ID: "
        )

        distribution_config_response =
 self.cloudfront_client.get_distribution_config(
            Id=distribution_id
        )
        distribution_config = distribution_config_response["DistributionConfig"]
        distribution_etag = distribution_config_response["ETag"]

        distribution_config["Comment"] = input(
            f"\nThe current comment for distribution {distribution_id} is "
            f"'{distribution_config['Comment']}'.\n"
            f"Enter a new comment: "
        )
```

```
        self.cloudfront_client.update_distribution(
            DistributionConfig=distribution_config,
            Id=distribution_id,
            IfMatch=distribution_etag,
        )
        print("Done!")
```

- Einzelheiten zur API finden Sie GetDistributionConfigin AWS SDK for Python (Boto3) API Reference.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwendung CloudFront mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung von **`ListCloudFrontOriginAccessIdentities`** mit einer CLI

Die folgenden Code-Beispiele zeigen, wie `ListCloudFrontOriginAccessIdentities` verwendet wird.

CLI

AWS CLI

Um die ursprünglichen CloudFront Zugriffsidentitäten aufzulisten

Im folgenden Beispiel wird eine Liste der CloudFront Origin-Zugriffsidentitäten (OAIs) in Ihrem Konto abgerufen:AWS

```
aws cloudfront list-cloud-front-origin-access-identities
```

Ausgabe:

```
{
    "CloudFrontOriginAccessIdentityList": {
        "Items": [
            {
                "Id": "E74FTE3AEXAMPLE",
```

```
                "S3CanonicalUserId":
 "cd13868f797c227fbea2830611a26fe0a21ba1b826ab4bed9b7771c9aEXAMPLE",
                "Comment": "Example OAI"
            },
            {
                "Id": "EH1HDMBEXAMPLE",
                "S3CanonicalUserId":
 "1489f6f2e6faacaae7ff64c4c3e6956c24f78788abfc1718c3527c263bf7a17EXAMPLE",
                "Comment": "Test OAI"
            },
            {
                "Id": "E2X2C9TEXAMPLE",
                "S3CanonicalUserId":
 "cbfeebb915a64749f9be546a45b3fcfd3a31c779673c13c4dd460911ae402c2EXAMPLE",
                "Comment": "Example OAI #2"
            }
        ]
    }
}
```

- Einzelheiten zur API finden Sie ListCloudFrontOriginAccessIdentitiesin der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: In diesem Beispiel wird eine Liste der CloudFront Amazon-Origin-Zugriffsidentitäten zurückgegeben. Da der MaxItem Parameter - den Wert 2 angibt, enthalten die Ergebnisse zwei Identitäten.

```
Get-CFCloudFrontOriginAccessIdentityList -MaxItem 2
```

Ausgabe:

```
IsTruncated : True
Items       : {E326XXXXXXXXXT, E1YWXXXXXXX9B}
Marker      :
MaxItems    : 2
NextMarker  : E1YXXXXXXXXX9B
Quantity    : 2
```

- Einzelheiten zur API finden Sie unter ListCloudFrontOriginAccessIdentities AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: In diesem Beispiel wird eine Liste der CloudFront Amazon-Origin-Zugriffsidentitäten zurückgegeben. Da der MaxItem Parameter - den Wert 2 angibt, enthalten die Ergebnisse zwei Identitäten.

```
Get-CFCloudFrontOriginAccessIdentityList -MaxItem 2
```

Ausgabe:

```
IsTruncated : True
Items       : {E326XXXXXXXXXT, E1YWXXXXXXX9B}
Marker      :
MaxItems    : 2
NextMarker  : E1YXXXXXXXXX9B
Quantity    : 2
```

- Einzelheiten zur API finden Sie unter ListCloudFrontOriginAccessIdentities AWS -Tools für PowerShellCmdlet-Referenz (V5).

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwendung CloudFront mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **`ListDistributions`**mit einem AWS SDK oder CLI

Die folgenden Code-Beispiele zeigen, wie `ListDistributions` verwendet wird.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- Erste Schritte mit Webanwendungs-Firewalls

CLI

AWS CLI

Um CloudFront Distributionen aufzulisten

Im folgenden Beispiel wird eine Liste der CloudFront Verteilungen in Ihrem AWS Konto abgerufen.

```
aws cloudfront list-distributions
```

Ausgabe:

```
{
    "DistributionList": {
        "Items": [
            {
                "Id": "E23YS8OEXAMPLE",
                "ARN": "arn:aws:cloudfront::123456789012:distribution/
E23YS8OEXAMPLE",
                "Status": "Deployed",
                "LastModifiedTime": "2024-08-05T18:23:40.375000+00:00",
                "DomainName": "abcdefgh12ijk.cloudfront.net",
                "Aliases": {
                    "Quantity": 0
                },
                "Origins": {
                    "Quantity": 1,
                    "Items": [
                        {
                            "Id": "amzn-s3-demo-bucket.s3.us-
east-1.amazonaws.com",
                            "DomainName": "amzn-s3-demo-bucket.s3.us-
east-1.amazonaws.com",
                            "OriginPath": "",
                            "CustomHeaders": {
                                "Quantity": 0
                            },
                            "S3OriginConfig": {
                                "OriginAccessIdentity": ""
                            },
                            "ConnectionAttempts": 3,
                            "ConnectionTimeout": 10,
                            "OriginShield": {
                                "Enabled": false
                            },
                            "OriginAccessControlId": "EIAP8PEXAMPLE"
                        }
                    ]
```

```
                },
                "OriginGroups": {
                    "Quantity": 0
                },
                "DefaultCacheBehavior": {
                    "TargetOriginId": "amzn-s3-demo-bucket.s3.us-
east-1.amazonaws.com",
                    "TrustedSigners": {
                        "Enabled": false,
                        "Quantity": 0
                    },
                    "TrustedKeyGroups": {
                        "Enabled": false,
                        "Quantity": 0
                    },
                    "ViewerProtocolPolicy": "allow-all",
                    "AllowedMethods": {
                        "Quantity": 2,
                        "Items": [
                            "HEAD",
                            "GET"
                        ],
                        "CachedMethods": {
                            "Quantity": 2,
                            "Items": [
                                "HEAD",
                                "GET"
                            ]
                        }
                    },
                    "SmoothStreaming": false,
                    "Compress": true,
                    "LambdaFunctionAssociations": {
                        "Quantity": 0
                    },
                    "FunctionAssociations": {
                        "Quantity": 0
                    },
                    "FieldLevelEncryptionId": "",
                    "CachePolicyId": "658327ea-f89d-4fab-a63d-7e886EXAMPLE"
                },
                "CacheBehaviors": {
                    "Quantity": 0
                },
```

```
                "CustomErrorResponses": {
                    "Quantity": 0
                },
                "Comment": "",
                "PriceClass": "PriceClass_All",
                "Enabled": true,
                "ViewerCertificate": {
                    "CloudFrontDefaultCertificate": true,
                    "SSLSupportMethod": "vip",
                    "MinimumProtocolVersion": "TLSv1",
                    "CertificateSource": "cloudfront"
                },
                "Restrictions": {
                    "GeoRestriction": {
                        "RestrictionType": "none",
                        "Quantity": 0
                    }
                },
                "WebACLId": "",
                "HttpVersion": "HTTP2",
                "IsIPV6Enabled": true,
                "Staging": false
            }
        ]
    }
}
```

- Einzelheiten zur API finden Sie ListDistributionsin der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: Gibt Distributionen zurück.

```
Get-CFDistributionList
```

- Einzelheiten zur API finden Sie unter ListDistributions AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: Gibt Distributionen zurück.

```
Get-CFDistributionList
```

- Einzelheiten zur API finden Sie unter ListDistributions AWS -Tools für PowerShellCmdlet-Referenz (V5).

Python

SDK für Python (Boto3)

> **Note**
>
> Es gibt noch mehr dazu. GitHub Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS-Code-Beispiel- einrichten und ausführen.

```
class CloudFrontWrapper:
    """Encapsulates Amazon CloudFront operations."""

    def __init__(self, cloudfront_client):
        """
        :param cloudfront_client: A Boto3 CloudFront client
        """
        self.cloudfront_client = cloudfront_client


    def list_distributions(self):
        print("CloudFront distributions:\n")
        distributions = self.cloudfront_client.list_distributions()
        if distributions["DistributionList"]["Quantity"] > 0:
            for distribution in distributions["DistributionList"]["Items"]:
                print(f"Domain: {distribution['DomainName']}")
                print(f"Distribution Id: {distribution['Id']}")
                print(
                    f"Certificate Source: "
                    f"{distribution['ViewerCertificate']['CertificateSource']}"
                )
                if distribution["ViewerCertificate"]["CertificateSource"] ==
 "acm":
                    print(
                        f"Certificate: {distribution['ViewerCertificate']
['Certificate']}"
```

```
                    )
                print("")
        else:
            print("No CloudFront distributions detected.")
```

- Einzelheiten zur API finden Sie ListDistributionsin AWS SDK for Python (Boto3) API Reference.

SAP ABAP

SDK für SAP ABAP

> **Note**
>
> Es gibt noch mehr dazu. GitHub Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS-Code-Beispiel- einrichten und ausführen.

```
    TRY.
        oo_result = lo_fnt->listdistributions( ). " oo_result is returned for
testing purposes. "
        MESSAGE 'Retrieved list of CloudFront distributions.' TYPE 'I'.
      CATCH /aws1/cx_fntinvalidargument.
        MESSAGE 'Invalid argument provided.' TYPE 'E'.
    ENDTRY.
```

- Einzelheiten zur API finden Sie ListDistributionsin der API-Referenz zum AWS SDK für SAP ABAP.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unterVerwendung CloudFront mit einem AWS SDK. Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **`UpdateDistribution`**mit einem AWS SDK oder CLI

Die folgenden Code-Beispiele zeigen, wie `UpdateDistribution` verwendet wird.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- Beginnen Sie mit einem grundlegenden Netzwerk zur Verteilung von Inhalten

CLI

AWS CLI

Beispiel 1: Um das Standard-Root-Objekt einer CloudFront Distribution zu aktualisieren

Im folgenden Beispiel wird das Standard-Stammobjekt `index.html` für die CloudFront Distribution mit der ID aktualisiert`EDFDVBD6EXAMPLE`.

```
aws cloudfront update-distribution \
    --id EDFDVBD6EXAMPLE \
    --default-root-object index.html
```

Ausgabe:

```
{
    "ETag": "E2QWRUHEXAMPLE",
    "Distribution": {
        "Id": "EDFDVBD6EXAMPLE",
        "ARN": "arn:aws:cloudfront::123456789012:distribution/EDFDVBD6EXAMPLE",
        "Status": "InProgress",
        "LastModifiedTime": "2019-12-06T18:55:39.870Z",
        "InProgressInvalidationBatches": 0,
        "DomainName": "d111111abcdef8.cloudfront.net",
        "ActiveTrustedSigners": {
            "Enabled": false,
            "Quantity": 0
        },
        "DistributionConfig": {
            "CallerReference": "6b10378d-49be-4c4b-a642-419ccaf8f3b5",
            "Aliases": {
                "Quantity": 0
            },
            "DefaultRootObject": "index.html",
            "Origins": {
                "Quantity": 1,
                "Items": [
```

```
                    {
                        "Id": "example-website",
                        "DomainName": "www.example.com",
                        "OriginPath": "",
                        "CustomHeaders": {
                            "Quantity": 0
                        },
                        "CustomOriginConfig": {
                            "HTTPPort": 80,
                            "HTTPSPort": 443,
                            "OriginProtocolPolicy": "match-viewer",
                            "OriginSslProtocols": {
                                "Quantity": 2,
                                "Items": [
                                    "SSLv3",
                                    "TLSv1"
                                ]
                            },
                            "OriginReadTimeout": 30,
                            "OriginKeepaliveTimeout": 5
                        }
                    }
                ]
            },
            "OriginGroups": {
                "Quantity": 0
            },
            "DefaultCacheBehavior": {
                "TargetOriginId": "example-website",
                "ForwardedValues": {
                    "QueryString": false,
                    "Cookies": {
                        "Forward": "none"
                    },
                    "Headers": {
                        "Quantity": 1,
                        "Items": [
                            "*"
                        ]
                    },
                    "QueryStringCacheKeys": {
                        "Quantity": 0
                    }
                },
```

```
                "TrustedSigners": {
                    "Enabled": false,
                    "Quantity": 0
                },
                "ViewerProtocolPolicy": "allow-all",
                "MinTTL": 0,
                "AllowedMethods": {
                    "Quantity": 2,
                    "Items": [
                        "HEAD",
                        "GET"
                    ],
                    "CachedMethods": {
                        "Quantity": 2,
                        "Items": [
                            "HEAD",
                            "GET"
                        ]
                    }
                },
                "SmoothStreaming": false,
                "DefaultTTL": 86400,
                "MaxTTL": 31536000,
                "Compress": false,
                "LambdaFunctionAssociations": {
                    "Quantity": 0
                },
                "FieldLevelEncryptionId": ""
            },
            "CacheBehaviors": {
                "Quantity": 0
            },
            "CustomErrorResponses": {
                "Quantity": 0
            },
            "Comment": "",
            "Logging": {
                "Enabled": false,
                "IncludeCookies": false,
                "Bucket": "",
                "Prefix": ""
            },
            "PriceClass": "PriceClass_All",
            "Enabled": true,
```

```
            "ViewerCertificate": {
                "CloudFrontDefaultCertificate": true,
                "MinimumProtocolVersion": "TLSv1",
                "CertificateSource": "cloudfront"
            },
            "Restrictions": {
                "GeoRestriction": {
                    "RestrictionType": "none",
                    "Quantity": 0
                }
            },
            "WebACLId": "",
            "HttpVersion": "http1.1",
            "IsIPV6Enabled": true
        }
    }
}
```

Beispiel 2: Um eine CloudFront Distribution zu aktualisieren

Im folgenden Beispiel wird die CloudFront Distribution mit der ID deaktiviert, `EMLARXS9EXAMPLE` indem die Verteilungskonfiguration in einer JSON-Datei mit dem Namen `dist-config-disable.json` bereitgestellt wird. Um eine Distribution zu aktualisieren, müssen Sie die Option `--if-match` verwenden, um das `ETag` der Distribution anzugeben. Um den `ETag` zu erhalten, verwenden Sie bitte den Befehl „get-Distribution“ oder „get-Distribution-Config“. Beachten Sie, dass das Feld `Enabled` in der JSON-Datei auf `false` gesetzt ist.

Nachdem Sie das folgende Beispiel verwendet haben, um die Distribution zu deaktivieren, können Sie den Befehl „delete-distribution“ verwenden, um sie zu löschen.

```
aws cloudfront update-distribution \
    --id EMLARXS9EXAMPLE \
    --if-match E2QWRUHEXAMPLE \
    --distribution-config file://dist-config-disable.json
```

Inhalt von `dist-config-disable.json`:

```
{
    "CallerReference": "cli-1574382155-496510",
    "Aliases": {
        "Quantity": 0
```

```
    },
    "DefaultRootObject": "index.html",
    "Origins": {
        "Quantity": 1,
        "Items": [
            {
                "Id": "amzn-s3-demo-bucket.s3.amazonaws.com-1574382155-273939",
                "DomainName": "amzn-s3-demo-bucket.s3.amazonaws.com",
                "OriginPath": "",
                "CustomHeaders": {
                    "Quantity": 0
                },
                "S3OriginConfig": {
                    "OriginAccessIdentity": ""
                }
            }
        ]
    },
    "OriginGroups": {
        "Quantity": 0
    },
    "DefaultCacheBehavior": {
        "TargetOriginId": "amzn-s3-demo-
bucket.s3.amazonaws.com-1574382155-273939",
        "ForwardedValues": {
            "QueryString": false,
            "Cookies": {
                "Forward": "none"
            },
            "Headers": {
                "Quantity": 0
            },
            "QueryStringCacheKeys": {
                "Quantity": 0
            }
        },
        "TrustedSigners": {
            "Enabled": false,
            "Quantity": 0
        },
        "ViewerProtocolPolicy": "allow-all",
        "MinTTL": 0,
        "AllowedMethods": {
            "Quantity": 2,
```

```
            "Items": [
                "HEAD",
                "GET"
            ],
            "CachedMethods": {
                "Quantity": 2,
                "Items": [
                    "HEAD",
                    "GET"
                ]
            }
        },
        "SmoothStreaming": false,
        "DefaultTTL": 86400,
        "MaxTTL": 31536000,
        "Compress": false,
        "LambdaFunctionAssociations": {
            "Quantity": 0
        },
        "FieldLevelEncryptionId": ""
    },
    "CacheBehaviors": {
        "Quantity": 0
    },
    "CustomErrorResponses": {
        "Quantity": 0
    },
    "Comment": "",
    "Logging": {
        "Enabled": false,
        "IncludeCookies": false,
        "Bucket": "",
        "Prefix": ""
    },
    "PriceClass": "PriceClass_All",
    "Enabled": false,
    "ViewerCertificate": {
        "CloudFrontDefaultCertificate": true,
        "MinimumProtocolVersion": "TLSv1",
        "CertificateSource": "cloudfront"
    },
    "Restrictions": {
        "GeoRestriction": {
            "RestrictionType": "none",
```

```
            "Quantity": 0
        }
    },
    "WebACLId": "",
    "HttpVersion": "http2",
    "IsIPV6Enabled": true
}
```

Ausgabe:

```
{
    "ETag": "E9LHASXEXAMPLE",
    "Distribution": {
        "Id": "EMLARXS9EXAMPLE",
        "ARN": "arn:aws:cloudfront::123456789012:distribution/EMLARXS9EXAMPLE",
        "Status": "InProgress",
        "LastModifiedTime": "2019-12-06T18:32:35.553Z",
        "InProgressInvalidationBatches": 0,
        "DomainName": "d111111abcdef8.cloudfront.net",
        "ActiveTrustedSigners": {
            "Enabled": false,
            "Quantity": 0
        },
        "DistributionConfig": {
            "CallerReference": "cli-1574382155-496510",
            "Aliases": {
                "Quantity": 0
            },
            "DefaultRootObject": "index.html",
            "Origins": {
                "Quantity": 1,
                "Items": [
                    {
                        "Id": "amzn-s3-demo-
bucket.s3.amazonaws.com-1574382155-273939",
                        "DomainName": "amzn-s3-demo-bucket.s3.amazonaws.com",
                        "OriginPath": "",
                        "CustomHeaders": {
                            "Quantity": 0
                        },
                        "S3OriginConfig": {
                            "OriginAccessIdentity": ""
                        }
```

```
                    }
                ]
            },
            "OriginGroups": {
                "Quantity": 0
            },
            "DefaultCacheBehavior": {
                "TargetOriginId": "amzn-s3-demo-
bucket.s3.amazonaws.com-1574382155-273939",
                "ForwardedValues": {
                    "QueryString": false,
                    "Cookies": {
                        "Forward": "none"
                    },
                    "Headers": {
                        "Quantity": 0
                    },
                    "QueryStringCacheKeys": {
                        "Quantity": 0
                    }
                },
                "TrustedSigners": {
                    "Enabled": false,
                    "Quantity": 0
                },
                "ViewerProtocolPolicy": "allow-all",
                "MinTTL": 0,
                "AllowedMethods": {
                    "Quantity": 2,
                    "Items": [
                        "HEAD",
                        "GET"
                    ],
                    "CachedMethods": {
                        "Quantity": 2,
                        "Items": [
                            "HEAD",
                            "GET"
                        ]
                    }
                },
                "SmoothStreaming": false,
                "DefaultTTL": 86400,
                "MaxTTL": 31536000,
```

```
                "Compress": false,
                "LambdaFunctionAssociations": {
                    "Quantity": 0
                },
                "FieldLevelEncryptionId": ""
            },
            "CacheBehaviors": {
                "Quantity": 0
            },
            "CustomErrorResponses": {
                "Quantity": 0
            },
            "Comment": "",
            "Logging": {
                "Enabled": false,
                "IncludeCookies": false,
                "Bucket": "",
                "Prefix": ""
            },
            "PriceClass": "PriceClass_All",
            "Enabled": false,
            "ViewerCertificate": {
                "CloudFrontDefaultCertificate": true,
                "MinimumProtocolVersion": "TLSv1",
                "CertificateSource": "cloudfront"
            },
            "Restrictions": {
                "GeoRestriction": {
                    "RestrictionType": "none",
                    "Quantity": 0
                }
            },
            "WebACLId": "",
            "HttpVersion": "http2",
            "IsIPV6Enabled": true
        }
    }
}
```

- Einzelheiten zur API finden Sie unter UpdateDistribution AWS CLIBefehlsreferenz.

Java

SDK für Java 2.x

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS-Code-Beispiel- einrichten und ausführen.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.cloudfront.CloudFrontClient;
import software.amazon.awssdk.services.cloudfront.model.GetDistributionRequest;
import software.amazon.awssdk.services.cloudfront.model.GetDistributionResponse;
import software.amazon.awssdk.services.cloudfront.model.Distribution;
import software.amazon.awssdk.services.cloudfront.model.DistributionConfig;
import
 software.amazon.awssdk.services.cloudfront.model.UpdateDistributionRequest;
import software.amazon.awssdk.services.cloudfront.model.CloudFrontException;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class ModifyDistribution {
    public static void main(String[] args) {
        final String usage = """

                Usage:
                    <id>\s

                Where:
                    id - the id value of the distribution.\s
                """;

        if (args.length != 1) {
            System.out.println(usage);
```

```
            System.exit(1);
        }

        String id = args[0];
        CloudFrontClient cloudFrontClient = CloudFrontClient.builder()
                .region(Region.AWS_GLOBAL)
                .build();

        modDistribution(cloudFrontClient, id);
        cloudFrontClient.close();
    }

    public static void modDistribution(CloudFrontClient cloudFrontClient, String
idVal) {
        try {
            // Get the Distribution to modify.
            GetDistributionRequest disRequest = GetDistributionRequest.builder()
                    .id(idVal)
                    .build();

            GetDistributionResponse response =
cloudFrontClient.getDistribution(disRequest);
            Distribution disObject = response.distribution();
            DistributionConfig config = disObject.distributionConfig();

            // Create a new DistributionConfig object and add new values to
comment and
            // aliases
            DistributionConfig config1 = DistributionConfig.builder()
                    .aliases(config.aliases()) // You can pass in new values here
                    .comment("New Comment")
                    .cacheBehaviors(config.cacheBehaviors())
                    .priceClass(config.priceClass())
                    .defaultCacheBehavior(config.defaultCacheBehavior())
                    .enabled(config.enabled())
                    .callerReference(config.callerReference())
                    .logging(config.logging())
                    .originGroups(config.originGroups())
                    .origins(config.origins())
                    .restrictions(config.restrictions())
                    .defaultRootObject(config.defaultRootObject())
                    .webACLId(config.webACLId())
                    .httpVersion(config.httpVersion())
                    .viewerCertificate(config.viewerCertificate())
```

```
                    .customErrorResponses(config.customErrorResponses())
                    .build();

            UpdateDistributionRequest updateDistributionRequest =
 UpdateDistributionRequest.builder()
                    .distributionConfig(config1)
                    .id(disObject.id())
                    .ifMatch(response.eTag())
                    .build();

            cloudFrontClient.updateDistribution(updateDistributionRequest);

        } catch (CloudFrontException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Einzelheiten zur API finden Sie UpdateDistributionin der AWS SDK for Java 2.x API-Referenz.

Python

SDK für Python (Boto3)

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS-Code-Beispiel- einrichten und ausführen.

```
class CloudFrontWrapper:
    """Encapsulates Amazon CloudFront operations."""

    def __init__(self, cloudfront_client):
        """
        :param cloudfront_client: A Boto3 CloudFront client
        """
        self.cloudfront_client = cloudfront_client
```

```
    def update_distribution(self):
        distribution_id = input(
            "This script updates the comment for a CloudFront distribution.\n"
            "Enter a CloudFront distribution ID: "
        )

        distribution_config_response =
self.cloudfront_client.get_distribution_config(
            Id=distribution_id
        )
        distribution_config = distribution_config_response["DistributionConfig"]
        distribution_etag = distribution_config_response["ETag"]

        distribution_config["Comment"] = input(
            f"\nThe current comment for distribution {distribution_id} is "
            f"'{distribution_config['Comment']}'.\n"
            f"Enter a new comment: "
        )
        self.cloudfront_client.update_distribution(
            DistributionConfig=distribution_config,
            Id=distribution_id,
            IfMatch=distribution_etag,
        )
        print("Done!")
```

- Einzelheiten zur API finden Sie UpdateDistributionin AWS SDK for Python (Boto3) API Reference.

SAP ABAP

SDK für SAP ABAP

> **Note**
>
> Es gibt noch mehr dazu. GitHub Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS-Code-Beispiel- einrichten und ausführen.

```
    TRY.
        " Get the current distribution configuration and ETag "
        DATA(lo_distribution_config_result) = lo_fnt-
>getdistributionconfig( iv_id = iv_distribution_id ).
        DATA(lo_old_config) = lo_distribution_config_result-
>get_distributionconfig( ).
        DATA(lv_etag) = lo_distribution_config_result->get_etag( ).

        " Create a new distribution config with the updated comment "
        " Since the config object is immutable, we need to create a new one with
 all existing values "
        DATA(lo_new_config) = NEW /aws1/cl_fntdistributionconfig(
          iv_callerreference = lo_old_config->get_callerreference( )
          io_aliases = lo_old_config->get_aliases( )
          iv_defaultrootobject = lo_old_config->get_defaultrootobject( )
          io_origins = lo_old_config->get_origins( )
          io_origingroups = lo_old_config->get_origingroups( )
          io_defaultcachebehavior = lo_old_config->get_defaultcachebehavior( )
          io_cachebehaviors = lo_old_config->get_cachebehaviors( )
          io_customerrorresponses = lo_old_config->get_customerrorresponses( )
          iv_comment = iv_comment
          io_logging = lo_old_config->get_logging( )
          iv_priceclass = lo_old_config->get_priceclass( )
          iv_enabled = lo_old_config->get_enabled( )
          io_viewercertificate = lo_old_config->get_viewercertificate( )
          io_restrictions = lo_old_config->get_restrictions( )
          iv_webaclid = lo_old_config->get_webaclid( )
          iv_httpversion = lo_old_config->get_httpversion( )
          iv_isipv6enabled = lo_old_config->get_isipv6enabled( ) ).

        " Update the distribution with the modified configuration "
        lo_fnt->updatedistribution(
          io_distributionconfig = lo_new_config
          iv_id = iv_distribution_id
          iv_ifmatch = lv_etag ).
        MESSAGE 'CloudFront distribution updated successfully.' TYPE 'I'.
      CATCH /aws1/cx_fntnosuchdistribution.
        MESSAGE 'Distribution does not exist.' TYPE 'E'.
      CATCH /aws1/cx_fntpreconditionfailed.
        MESSAGE 'Precondition failed - ETag mismatch.' TYPE 'E'.
      CATCH /aws1/cx_fntinvalidifmatchvrs.
        MESSAGE 'Invalid If-Match version.' TYPE 'E'.
    ENDTRY.
```

- Einzelheiten zur API finden Sie UpdateDistributionin der API-Referenz zum AWS SDK für SAP ABAP.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unterVerwendung CloudFront mit einem AWS SDK. Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Szenarien für die CloudFront Verwendung AWS SDKs

Die folgenden Codebeispiele zeigen Ihnen, wie Sie allgemeine Szenarien CloudFront mit AWS SDKs implementieren. Diese Szenarien zeigen Ihnen, wie Sie bestimmte Aufgaben erledigen können, indem Sie mehrere Funktionen innerhalb CloudFront oder in Kombination mit anderen AWS-Services aufrufen. Jedes Szenario enthält einen Link zum vollständigen Quell-Code, wo Sie Anleitungen zum Einrichten und Ausführen des Codes finden.

Szenarien zielen auf eine mittlere Erfahrungsebene ab, um Ihnen zu helfen, Service-Aktionen im Kontext zu verstehen.

Beispiele

- SaaS-Manager-Ressourcen erstellen AWS SDK
- Löschen Sie CloudFront Signaturressourcen mit AWS SDK
- Beginnen Sie mit einem einfachen Netzwerk zur Verteilung von Inhalten
- Erste Schritte mit Firewalls für Webanwendungen
- Erstellen Sie signierte URLs und Cookies mit einem AWS SDK

SaaS-Manager-Ressourcen erstellen AWS SDK

Das folgende Codebeispiel zeigt, wie Sie eine Multi-Tenant-Distribution und einen Distributionsmandanten mit verschiedenen Konfigurationen erstellen.

Java

SDK für Java 2.x

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS-Code-Beispiel- einrichten und ausführen.

Das folgende Beispiel zeigt, wie Sie eine Multi-Tenant-Distribution mit Parametern und einem Platzhalterzertifikat erstellen.

```
import software.amazon.awssdk.core.internal.waiters.ResponseOrException;
import software.amazon.awssdk.services.cloudfront.CloudFrontClient;
import software.amazon.awssdk.services.cloudfront.model.ConnectionMode;
import
 software.amazon.awssdk.services.cloudfront.model.CreateDistributionResponse;
import software.amazon.awssdk.services.cloudfront.model.Distribution;
import software.amazon.awssdk.services.cloudfront.model.GetDistributionResponse;
import software.amazon.awssdk.services.cloudfront.model.HttpVersion;
import software.amazon.awssdk.services.cloudfront.model.Method;
import software.amazon.awssdk.services.cloudfront.model.SSLSupportMethod;
import software.amazon.awssdk.services.cloudfront.model.ViewerProtocolPolicy;
import software.amazon.awssdk.services.cloudfront.waiters.CloudFrontWaiter;
import software.amazon.awssdk.services.s3.S3Client;

import java.time.Instant;

public class CreateMultiTenantDistribution {
    public static Distribution
 CreateMultiTenantDistributionWithCert(CloudFrontClient cloudFrontClient,
                                                                    S3Client
 s3Client,
                                                                    final String
 bucketName,
                                                                    final String
 certificateArn) {
        // fetch the origin info if necessary
        final String region = s3Client.headBucket(b ->
 b.bucket(bucketName)).sdkHttpResponse().headers()
```

```
                .get("x-amz-bucket-region").get(0);
        final String originDomain = bucketName + ".s3." + region +
 ".amazonaws.com";
        String originId = originDomain; // Use the originDomain value for the
 originId.

        CreateDistributionResponse createDistResponse =
 cloudFrontClient.createDistribution(builder -> builder
                .distributionConfig(b1 -> b1
                        .httpVersion(HttpVersion.HTTP2)
                        .enabled(true)
                        .comment("Template Distribution with cert built with
 java")
                        .connectionMode(ConnectionMode.TENANT_ONLY)
                        .callerReference(Instant.now().toString())
                        .viewerCertificate(certBuilder -> certBuilder
                                .acmCertificateArn(certificateArn)
                                .sslSupportMethod(SSLSupportMethod.SNI_ONLY))
                        .origins(b2 -> b2
                                .quantity(1)
                                .items(b3 -> b3
                                        .domainName(originDomain)
                                        .id(originId)
                                        .originPath("/{{tenantName}}")
                                        .s3OriginConfig(builder4 -> builder4
                                                .originAccessIdentity(
                                                        ""))))
                        .tenantConfig(b5 -> b5
                                .parameterDefinitions(b6 -> b6
                                        .name("tenantName")
                                        .definition(b7 -> b7
                                                .stringSchema(b8 -> b8
                                                        .comment("tenantName
 value")
                                                        .defaultValue("root")
                                                        .required(false)))))
                        .defaultCacheBehavior(b2 -> b2

 .viewerProtocolPolicy(ViewerProtocolPolicy.ALLOW_ALL)
                                .targetOriginId(originId)
                                .cachePolicyId("658327ea-f89d-4fab-
a63d-7e88639e58f6") // CachingOptimized Policy
                                .allowedMethods(b4 -> b4
                                        .quantity(2)
```

```
                                        .items(Method.HEAD, Method.GET)))
                ));

        final Distribution distribution = createDistResponse.distribution();
        try (CloudFrontWaiter cfWaiter =
CloudFrontWaiter.builder().client(cloudFrontClient).build()) {
            ResponseOrException<GetDistributionResponse> responseOrException =
cfWaiter
                    .waitUntilDistributionDeployed(builder ->
builder.id(distribution.id()))
                    .matched();
            responseOrException.response()
                    .orElseThrow(() -> new RuntimeException("Distribution not
created"));
        }
        return distribution;
    }

    public static Distribution
CreateMultiTenantDistributionNoCert(CloudFrontClient cloudFrontClient,
                                                                S3Client s3Client,
                                                                final String
bucketName) {
        // fetch the origin info if necessary
        final String region = s3Client.headBucket(b ->
b.bucket(bucketName)).sdkHttpResponse().headers()
                .get("x-amz-bucket-region").get(0);
        final String originDomain = bucketName + ".s3." + region +
".amazonaws.com";
        String originId = originDomain; // Use the originDomain value for the
originId.

        CreateDistributionResponse createDistResponse =
cloudFrontClient.createDistribution(builder -> builder
                .distributionConfig(b1 -> b1
                        .httpVersion(HttpVersion.HTTP2)
                        .enabled(true)
                        .comment("Template Distribution with cert built with
java")
                        .connectionMode(ConnectionMode.TENANT_ONLY)
                        .callerReference(Instant.now().toString())
                        .origins(b2 -> b2
                                .quantity(1)
                                .items(b3 -> b3
```

```
                                        .domainName(originDomain)
                                        .id(originId)
                                        .originPath("/{{tenantName}}")
                                        .s3OriginConfig(builder4 -> builder4
                                                .originAccessIdentity(
                                                        ""))))
                        .tenantConfig(b5 -> b5
                                .parameterDefinitions(b6 -> b6
                                        .name("tenantName")
                                        .definition(b7 -> b7
                                                .stringSchema(b8 -> b8
                                                        .comment("tenantName
 value")
                                                        .defaultValue("root")
                                                        .required(false)))))
                        .defaultCacheBehavior(b2 -> b2

 .viewerProtocolPolicy(ViewerProtocolPolicy.ALLOW_ALL)
                                .targetOriginId(originId)
                                .cachePolicyId("658327ea-f89d-4fab-
a63d-7e88639e58f6") // CachingOptimized Policy
                                .allowedMethods(b4 -> b4
                                        .quantity(2)
                                        .items(Method.HEAD, Method.GET)))
                ));

        final Distribution distribution = createDistResponse.distribution();
        try (CloudFrontWaiter cfWaiter =
 CloudFrontWaiter.builder().client(cloudFrontClient).build()) {
            ResponseOrException<GetDistributionResponse> responseOrException =
 cfWaiter
                    .waitUntilDistributionDeployed(builder ->
 builder.id(distribution.id()))
                    .matched();
            responseOrException.response()
                    .orElseThrow(() -> new RuntimeException("Distribution not
 created"));
        }
        return distribution;
    }
}
```

Das folgende Beispiel zeigt, wie Sie einen dieser Vorlage zugeordneten Distributionsmandanten erstellen und dabei den oben definierten Parameter verwenden. Beachten Sie, dass Sie hier keine Zertifikatsinformationen hinzufügen müssen, da Ihre Domain bereits durch die übergeordnete Vorlage abgedeckt ist.

```
import software.amazon.awssdk.services.cloudfront.CloudFrontClient;
import
 software.amazon.awssdk.services.cloudfront.model.CreateConnectionGroupResponse;
import
 software.amazon.awssdk.services.cloudfront.model.CreateDistributionTenantResponse;
import software.amazon.awssdk.services.cloudfront.model.DistributionTenant;
import
 software.amazon.awssdk.services.cloudfront.model.GetConnectionGroupResponse;
import software.amazon.awssdk.services.cloudfront.model.ValidationTokenHost;
import software.amazon.awssdk.services.route53.Route53Client;
import software.amazon.awssdk.services.route53.model.RRType;

import java.time.Instant;

public class CreateDistributionTenant {

    public static DistributionTenant
 createDistributionTenantNoCert(CloudFrontClient cloudFrontClient,
                                                                    Route53Client
 route53Client,
                                                                    String
 distributionId,
                                                                    String
 domain,
                                                                    String
 hostedZoneId) {
        CreateDistributionTenantResponse createResponse =
 cloudFrontClient.createDistributionTenant(builder -> builder
                .distributionId(distributionId)
                .domains(b1 -> b1
                        .domain(domain))
                .parameters(b2 -> b2
                        .name("tenantName")
                        .value("myTenant"))
                .enabled(false)
                .name("no-cert-tenant")
        );
```

```
        final DistributionTenant distributionTenant =
 createResponse.distributionTenant();

        // Then update the Route53 hosted zone to point your domain at the
 distribution tenant
        // We fetch the RoutingEndpoint to point to via the default connection
 group that was created for your tenant
        final GetConnectionGroupResponse fetchedConnectionGroup =
 cloudFrontClient.getConnectionGroup(builder -> builder
                .identifier(distributionTenant.connectionGroupId()));

        route53Client.changeResourceRecordSets(builder -> builder
                .hostedZoneId(hostedZoneId)
                .changeBatch(b1 -> b1
                        .comment("ChangeBatch comment")
                        .changes(b2 -> b2
                                .resourceRecordSet(b3 -> b3
                                        .name(domain)
                                        .type("CNAME")
                                        .ttl(300L)
                                        .resourceRecords(b4 -> b4

 .value(fetchedConnectionGroup.connectionGroup().routingEndpoint())))
                                .action("CREATE"))
                ));
        return distributionTenant;
    }

}
```

Wenn das Viewer-Zertifikat in der übergeordneten Vorlage weggelassen wurde, müssen Sie stattdessen Zertifikatsinformationen für die damit verknüpften Mandanten hinzufügen. Das folgende Beispiel zeigt, wie Sie dies mithilfe eines ACM-Zertifikats-ARN tun, der die erforderliche Domain für den Mandanten abdeckt.

```
import software.amazon.awssdk.services.cloudfront.CloudFrontClient;
import
 software.amazon.awssdk.services.cloudfront.model.CreateConnectionGroupResponse;
import
 software.amazon.awssdk.services.cloudfront.model.CreateDistributionTenantResponse;
```

```
import software.amazon.awssdk.services.cloudfront.model.DistributionTenant;
import
 software.amazon.awssdk.services.cloudfront.model.GetConnectionGroupResponse;
import software.amazon.awssdk.services.cloudfront.model.ValidationTokenHost;
import software.amazon.awssdk.services.route53.Route53Client;
import software.amazon.awssdk.services.route53.model.RRType;

import java.time.Instant;

public class CreateDistributionTenant {

    public static DistributionTenant
 createDistributionTenantWithCert(CloudFrontClient cloudFrontClient,

 Route53Client route53Client,
                                                                    String
 distributionId,
                                                                    String
 domain,
                                                                    String
 hostedZoneId,
                                                                    String
 certificateArn) {
        CreateDistributionTenantResponse createResponse =
 cloudFrontClient.createDistributionTenant(builder -> builder
                .distributionId(distributionId)
                .domains(b1 -> b1
                        .domain(domain))
                .enabled(false)
                .name("tenant-with-cert")
                .parameters(b2 -> b2
                        .name("tenantName")
                        .value("myTenant"))
                .customizations(b3 -> b3
                        .certificate(b4 -> b4
                                .arn(certificateArn))) // NOTE: Cert must be in
 Us-East-1 and cover the domain provided in this request

        );

        final DistributionTenant distributionTenant =
 createResponse.distributionTenant();
```

```
        // Then update the Route53 hosted zone to point your domain at the
 distribution tenant
        // We fetch the RoutingEndpoint to point to via the default connection
 group that was created for your tenant
        final GetConnectionGroupResponse fetchedConnectionGroup =
 cloudFrontClient.getConnectionGroup(builder -> builder
                .identifier(distributionTenant.connectionGroupId()));

        route53Client.changeResourceRecordSets(builder -> builder
                .hostedZoneId(hostedZoneId)
                .changeBatch(b1 -> b1
                        .comment("ChangeBatch comment")
                        .changes(b2 -> b2
                                .resourceRecordSet(b3 -> b3
                                        .name(domain)
                                        .type("CNAME")
                                        .ttl(300L)
                                        .resourceRecords(b4 -> b4

 .value(fetchedConnectionGroup.connectionGroup().routingEndpoint())))
                                .action("CREATE"))
                ));
        return distributionTenant;
    }

}
```

Das folgende Beispiel zeigt, wie Sie dies mit einer CloudFront-hosted verwalteten Zertifikatsanforderung tun können. Diese Methode eignet sich ideal, wenn noch kein Datenverkehr zu Ihrer Domain besteht. In diesem Fall erstellen wir eine, ConnectionGroup um eine zu generieren RoutingEndpoint. Dann verwenden wir das, RoutingEndpoint um DNS-Einträge zu erstellen, die den Domainbesitz verifizieren und auf die wir verweisen CloudFront. CloudFront stellt dann automatisch ein Token bereit, um den Domainbesitz zu validieren und ein verwaltetes Zertifikat zu erstellen.

```
import software.amazon.awssdk.services.cloudfront.CloudFrontClient;
import
 software.amazon.awssdk.services.cloudfront.model.CreateConnectionGroupResponse;
import
 software.amazon.awssdk.services.cloudfront.model.CreateDistributionTenantResponse;
```

```
import software.amazon.awssdk.services.cloudfront.model.DistributionTenant;
import
 software.amazon.awssdk.services.cloudfront.model.GetConnectionGroupResponse;
import software.amazon.awssdk.services.cloudfront.model.ValidationTokenHost;
import software.amazon.awssdk.services.route53.Route53Client;
import software.amazon.awssdk.services.route53.model.RRType;

import java.time.Instant;

public class CreateDistributionTenant {

    public static DistributionTenant
 createDistributionTenantCfHosted(CloudFrontClient cloudFrontClient,

 Route53Client route53Client,
                                                                    String
 distributionId,
                                                                    String
 domain,
                                                                    String
 hostedZoneId) throws InterruptedException {
        CreateConnectionGroupResponse createConnectionGroupResponse =
 cloudFrontClient.createConnectionGroup(builder -> builder
                .ipv6Enabled(true)
                .name("cf-hosted-connection-group")
                .enabled(true));

        route53Client.changeResourceRecordSets(builder -> builder
                .hostedZoneId(hostedZoneId)
                .changeBatch(b1 -> b1
                        .comment("cf-hosted domain validation record")
                        .changes(b2 -> b2
                                .resourceRecordSet(b3 -> b3
                                        .name(domain)
                                        .type(RRType.CNAME)
                                        .ttl(300L)
                                        .resourceRecords(b4 -> b4

 .value(createConnectionGroupResponse.connectionGroup().routingEndpoint())))
                                .action("CREATE"))
                ));

        // Give the R53 record time to propagate, if it isn't being returned by
 servers yet, the following call will fail
```

```
        Thread.sleep(60000);

        CreateDistributionTenantResponse createResponse =
 cloudFrontClient.createDistributionTenant(builder -> builder
                .distributionId(distributionId)
                .domains(b1 -> b1
                        .domain(domain))

 .connectionGroupId(createConnectionGroupResponse.connectionGroup().id())
                .enabled(false)
                .name("cf-hosted-tenant")
                .parameters(b2 -> b2
                        .name("tenantName")
                        .value("myTenant"))
                .managedCertificateRequest(b3 -> b3
                        .validationTokenHost(ValidationTokenHost.CLOUDFRONT)
                )
        );

        return createResponse.distributionTenant();
    }

}
```

Das folgende Beispiel zeigt, wie Sie dies mit einer selbst gehosteten verwalteten Zertifikatsanforderung durchführen können. Dies ist ideal, wenn bereits Datenverkehr zu Ihrer Domain besteht und Sie während einer Migration keine Ausfallzeiten in Kauf nehmen können. Am Ende dieses Beispiels wird der Mandant in einem Zustand erstellt, in dem die Domainvalidierung und die DNS-Einrichtung noch ausstehen. Folgen Sie den Schritten [hier] (https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/managed-cloudfront-certificates.html#complete -domain-ownership), um die Einrichtung abzuschließen, wenn Sie bereit sind, Traffic zu migrieren.

```
import software.amazon.awssdk.services.cloudfront.CloudFrontClient;
import
 software.amazon.awssdk.services.cloudfront.model.CreateConnectionGroupResponse;
import
 software.amazon.awssdk.services.cloudfront.model.CreateDistributionTenantResponse;
import software.amazon.awssdk.services.cloudfront.model.DistributionTenant;
```

```
import
 software.amazon.awssdk.services.cloudfront.model.GetConnectionGroupResponse;
import software.amazon.awssdk.services.cloudfront.model.ValidationTokenHost;
import software.amazon.awssdk.services.route53.Route53Client;
import software.amazon.awssdk.services.route53.model.RRType;

import java.time.Instant;

public class CreateDistributionTenant {

    public static DistributionTenant
 createDistributionTenantSelfHosted(CloudFrontClient cloudFrontClient,
                                                                    String
 distributionId,
                                                                    String
 domain) {
        CreateDistributionTenantResponse createResponse =
 cloudFrontClient.createDistributionTenant(builder -> builder
                .distributionId(distributionId)
                .domains(b1 -> b1
                        .domain(domain))
                .parameters(b2 -> b2
                        .name("tenantName")
                        .value("myTenant"))
                .enabled(false)
                .name("self-hosted-tenant")
                .managedCertificateRequest(b3 -> b3
                        .validationTokenHost(ValidationTokenHost.SELF_HOSTED)
                        .primaryDomainName(domain)
                )
        );

        return createResponse.distributionTenant();
    }

}
```

- Weitere API-Informationen finden Sie in den folgenden Themen der AWS SDK for Java 2.x-API-Referenz.
  - CreateDistribution
  - CreateDistributionTenant

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwendung CloudFront mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Löschen Sie CloudFront Signaturressourcen mit AWS SDK

Das folgende Codebeispiel zeigt, wie Ressourcen gelöscht werden, die für den Zugriff auf eingeschränkte Inhalte in einem Bucket von Amazon Simple Storage Service (Amazon S3) verwendet werden.

Java

SDK für Java 2.x

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS-Code-Beispiel- einrichten und ausführen.

```
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import software.amazon.awssdk.services.cloudfront.CloudFrontClient;
import software.amazon.awssdk.services.cloudfront.model.DeleteKeyGroupResponse;
import
 software.amazon.awssdk.services.cloudfront.model.DeleteOriginAccessControlResponse;
import software.amazon.awssdk.services.cloudfront.model.DeletePublicKeyResponse;
import software.amazon.awssdk.services.cloudfront.model.GetKeyGroupResponse;
import
 software.amazon.awssdk.services.cloudfront.model.GetOriginAccessControlResponse;
import software.amazon.awssdk.services.cloudfront.model.GetPublicKeyResponse;

public class DeleteSigningResources {
    private static final Logger logger =
 LoggerFactory.getLogger(DeleteSigningResources.class);

    public static void deleteOriginAccessControl(final CloudFrontClient
 cloudFrontClient,
            final String originAccessControlId) {
        GetOriginAccessControlResponse getResponse = cloudFrontClient
                .getOriginAccessControl(b -> b.id(originAccessControlId));
```

```
        DeleteOriginAccessControlResponse deleteResponse =
 cloudFrontClient.deleteOriginAccessControl(builder -> builder
                .id(originAccessControlId)
                .ifMatch(getResponse.eTag()));
        if (deleteResponse.sdkHttpResponse().isSuccessful()) {
            logger.info("Successfully deleted Origin Access Control [{}]",
 originAccessControlId);
        }
    }

    public static void deleteKeyGroup(final CloudFrontClient cloudFrontClient,
 final String keyGroupId) {

        GetKeyGroupResponse getResponse = cloudFrontClient.getKeyGroup(b ->
 b.id(keyGroupId));
        DeleteKeyGroupResponse deleteResponse =
 cloudFrontClient.deleteKeyGroup(builder -> builder
                .id(keyGroupId)
                .ifMatch(getResponse.eTag()));
        if (deleteResponse.sdkHttpResponse().isSuccessful()) {
            logger.info("Successfully deleted Key Group [{}]", keyGroupId);
        }
    }

    public static void deletePublicKey(final CloudFrontClient cloudFrontClient,
 final String publicKeyId) {
        GetPublicKeyResponse getResponse = cloudFrontClient.getPublicKey(b ->
 b.id(publicKeyId));

        DeletePublicKeyResponse deleteResponse =
 cloudFrontClient.deletePublicKey(builder -> builder
                .id(publicKeyId)
                .ifMatch(getResponse.eTag()));

        if (deleteResponse.sdkHttpResponse().isSuccessful()) {
            logger.info("Successfully deleted Public Key [{}]", publicKeyId);
        }
    }
}
```

- Weitere API-Informationen finden Sie in den folgenden Themen der AWS SDK for Java 2.x-API-Referenz.

- DeleteKeyGroup
- DeleteOriginAccessControl
- DeletePublicKey

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unterVerwendung CloudFront mit einem AWS SDK. Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Beginnen Sie mit einem einfachen Netzwerk zur Verteilung von Inhalten

Wie das aussehen kann, sehen Sie am nachfolgenden Beispielcode:

- Erstellen Sie einen Amazon S3 S3-Bucket
- Laden Sie Inhalte in den Bucket hoch
- Erstellen Sie eine CloudFront Distribution mit OAC
- Bereinigen von Ressourcen

Bash

AWS CLI mit Bash-Skript

> **Note**
>
> Es gibt noch mehr dazu. GitHub Hier finden Sie das vollständige Beispiel und erfahren, wie Sie es im Tutorials und Beispiele-Repository für Entwickler einrichten und ausführen.

```
#!/bin/bash

# CloudFront Getting Started Tutorial Script
# This script creates an S3 bucket, uploads sample content, creates a CloudFront
 distribution with OAC,
# and demonstrates how to access content through CloudFront.

set -euo pipefail

# Set up logging
```

```
LOG_FILE="cloudfront-tutorial.log"
exec > >(tee -a "$LOG_FILE") 2>&1

echo "Starting CloudFront Getting Started Tutorial at $(date)"

# Function to handle errors
handle_error() {
    echo "ERROR: $1" >&2
    echo "Resources created before error:"
    if [ -n "${BUCKET_NAME:-}" ]; then
        echo "- S3 Bucket: $BUCKET_NAME"
    fi
    if [ -n "${OAC_ID:-}" ]; then
        echo "- CloudFront Origin Access Control: $OAC_ID"
    fi
    if [ -n "${DISTRIBUTION_ID:-}" ]; then
        echo "- CloudFront Distribution: $DISTRIBUTION_ID"
    fi

    echo "Attempting to clean up resources..."
    cleanup
    exit 1
}

# Function to clean up resources
cleanup() {
    echo "Cleaning up resources..."

    if [ -n "${DISTRIBUTION_ID:-}" ]; then
        echo "Disabling CloudFront distribution $DISTRIBUTION_ID..."

        # Get the current configuration and ETag in one call
        DIST_CONFIG=$(aws cloudfront get-distribution-config --id
 "$DISTRIBUTION_ID" 2>/dev/null) || {
            echo "Failed to get distribution config. Continuing with cleanup..."
            DIST_CONFIG=""
        }

        if [ -n "$DIST_CONFIG" ]; then
            ETAG=$(echo "$DIST_CONFIG" | jq -r '.ETag')

            # Modify and update distribution in one pipeline
            if echo "$DIST_CONFIG" | jq '.DistributionConfig.Enabled = false' | \
                aws cloudfront update-distribution \
```

```
                    --id "$DISTRIBUTION_ID" \
                    --distribution-config "$(cat)" \
                    --if-match "$ETAG" 2>/dev/null; then

                echo "Waiting for distribution to be disabled (this may take
 several minutes)..."
                aws cloudfront wait distribution-deployed --id "$DISTRIBUTION_ID"
 2>/dev/null || {
                    echo "Distribution deployment wait timed out. Proceeding with
 deletion..."
                }

                # Get fresh ETag for deletion
                DIST_CONFIG=$(aws cloudfront get-distribution-config --id
 "$DISTRIBUTION_ID" 2>/dev/null) || {
                    echo "Failed to get updated config. Skipping distribution
 deletion..."
                    DIST_CONFIG=""
                }

                if [ -n "$DIST_CONFIG" ]; then
                    ETAG=$(echo "$DIST_CONFIG" | jq -r '.ETag')
                    aws cloudfront delete-distribution --id "$DISTRIBUTION_ID" --
if-match "$ETAG" 2>/dev/null && \
                        echo "CloudFront distribution deleted." || \
                        echo "Failed to delete distribution. You may need to
 delete it manually."
                fi
            else
                echo "Failed to disable distribution. Continuing with cleanup..."
            fi
        fi
    fi

    if [ -n "${OAC_ID:-}" ]; then
        echo "Deleting Origin Access Control $OAC_ID..."
        OAC_DATA=$(aws cloudfront get-origin-access-control --id "$OAC_ID" 2>/
dev/null) || {
            echo "Failed to get Origin Access Control. You may need to delete it
 manually."
            OAC_DATA=""
        }

        if [ -n "$OAC_DATA" ]; then
```

```
            OAC_ETAG=$(echo "$OAC_DATA" | jq -r '.ETag')
            aws cloudfront delete-origin-access-control --id "$OAC_ID" --if-match
 "$OAC_ETAG" 2>/dev/null && \
                echo "Origin Access Control deleted." || \
                echo "Failed to delete Origin Access Control. You may need to
 delete it manually."
        fi
    fi

    if [ -n "${BUCKET_NAME:-}" ] && [ "$BUCKET_IS_SHARED" != "true" ]; then
        echo "Deleting S3 bucket $BUCKET_NAME and its contents..."
        aws s3 rm "s3://$BUCKET_NAME" --recursive 2>/dev/null || {
            echo "Failed to remove bucket contents. Continuing with bucket
 deletion..."
        }

        aws s3 rb "s3://$BUCKET_NAME" 2>/dev/null && \
            echo "S3 bucket deleted." || \
            echo "Failed to delete bucket. You may need to delete it manually."
    fi

    # Clean up temporary files
    rm -f temp_disabled_config.json distribution-config.json bucket-policy.json
 2>/dev/null || true
    rm -rf temp_content 2>/dev/null || true
}

# Trap errors and cleanup
trap 'handle_error "Script interrupted"' INT TERM

# Validate AWS CLI is available
if ! command -v aws &> /dev/null; then
    handle_error "AWS CLI is not installed or not in PATH"
fi

# Validate jq is available
if ! command -v jq &> /dev/null; then
    handle_error "jq is not installed or not in PATH"
fi

# Validate AWS credentials are configured
if ! aws sts get-caller-identity &> /dev/null; then
    handle_error "AWS credentials are not configured or invalid"
fi
```

```
# Initialize variables
BUCKET_NAME=""
OAC_ID=""
DISTRIBUTION_ID=""
BUCKET_IS_SHARED=false

# Generate a random identifier for the bucket name using secure random
RANDOM_ID=$(openssl rand -hex 6)
if [ -z "$RANDOM_ID" ]; then
    handle_error "Failed to generate random identifier"
fi

# Check for shared prereq bucket and get account ID in parallel calls
PREREQ_BUCKET=$(aws cloudformation describe-stacks --stack-name tutorial-prereqs-
bucket \
    --query 'Stacks[0].Outputs[?OutputKey==`BucketName`].OutputValue' --output
 text 2>/dev/null) || PREREQ_BUCKET=""

if [ -n "$PREREQ_BUCKET" ] && [ "$PREREQ_BUCKET" != "None" ]; then
    BUCKET_NAME="$PREREQ_BUCKET"
    BUCKET_IS_SHARED=true
    echo "Using shared bucket: $BUCKET_NAME"
else
    BUCKET_IS_SHARED=false
    BUCKET_NAME="cloudfront-${RANDOM_ID}"
fi
echo "Using bucket name: $BUCKET_NAME"

# Get AWS account ID early
ACCOUNT_ID=$(aws sts get-caller-identity --query 'Account' --output text)
if [ $? -ne 0 ]; then
    handle_error "Failed to get AWS account ID"
fi

# Validate account ID format
if ! [[ "$ACCOUNT_ID" =~ ^[0-9]{12}$ ]]; then
    handle_error "Invalid AWS account ID format: $ACCOUNT_ID"
fi

# Create a temporary directory for content with restrictive permissions
TEMP_DIR="temp_content"
mkdir -p "$TEMP_DIR/css"
chmod 700 "$TEMP_DIR"
```

```
if [ $? -ne 0 ]; then
    handle_error "Failed to create temporary directory"
fi

# Step 1: Create an S3 bucket (only if not shared)
if [ "$BUCKET_IS_SHARED" != "true" ]; then
    echo "Creating S3 bucket: $BUCKET_NAME"
    aws s3 mb "s3://$BUCKET_NAME" --region us-east-1
    if [ $? -ne 0 ]; then
        handle_error "Failed to create S3 bucket"
    fi

    aws s3api put-bucket-tagging --bucket "$BUCKET_NAME" --tagging
 'TagSet=[{Key=project,Value=doc-smith},{Key=tutorial,Value=cloudfront-
gettingstarted}]'

    # Batch bucket configuration calls for efficiency
    aws s3api put-bucket-versioning --bucket "$BUCKET_NAME" --versioning-
configuration Status=Enabled &
    aws s3api put-public-access-block \
        --bucket "$BUCKET_NAME" \
        --public-access-block-configuration
 "BlockPublicAcls=true,IgnorePublicAcls=true,BlockPublicPolicy=true,RestrictPublicBuckets
 &
    aws s3api put-bucket-encryption \
        --bucket "$BUCKET_NAME" \
        --server-side-encryption-configuration '{
            "Rules": [
                {
                    "ApplyServerSideEncryptionByDefault": {
                        "SSEAlgorithm": "AES256"
                    }
                }
            ]
        }' &
    wait
fi

# Step 2: Create sample content
echo "Creating sample content..."
cat > "$TEMP_DIR/index.html" << 'EOF'
<!DOCTYPE html>
<html>
<head>
```

```
    <title>Hello World</title>
    <link rel="stylesheet" type="text/css" href="css/styles.css">
</head>
<body>
    <h1>Hello world!</h1>
</body>
</html>
EOF

cat > "$TEMP_DIR/css/styles.css" << 'EOF'
body {
    font-family: Arial, sans-serif;
    margin: 40px;
    background-color: #f5f5f5;
}
h1 {
    color: #333;
    text-align: center;
}
EOF

# Set restrictive permissions on content files
chmod 600 "$TEMP_DIR/index.html" "$TEMP_DIR/css/styles.css"

# Step 3: Upload content to the S3 bucket with encryption and metadata
echo "Uploading content to S3 bucket..."
aws s3 cp "$TEMP_DIR/" "s3://$BUCKET_NAME/" --recursive \
    --sse AES256 \
    --metadata "Source=CloudFrontTutorial"
if [ $? -ne 0 ]; then
    handle_error "Failed to upload content to S3 bucket"
fi

# Step 4: Create Origin Access Control
echo "Creating Origin Access Control..."
OAC_RESPONSE=$(aws cloudfront create-origin-access-control \
    --origin-access-control-config Name="oac-for-
$BUCKET_NAME",SigningProtocol=sigv4,SigningBehavior=always,OriginAccessControlOriginType=

if [ $? -ne 0 ]; then
    handle_error "Failed to create Origin Access Control"
fi

OAC_ID=$(echo "$OAC_RESPONSE" | jq -r '.OriginAccessControl.Id')
```

```
if [ -z "$OAC_ID" ] || [ "$OAC_ID" = "null" ]; then
    handle_error "Failed to extract OAC ID from response"
fi

# Validate OAC ID format (alphanumeric and hyphens)
if ! [[ "$OAC_ID" =~ ^[A-Z0-9]+$ ]]; then
    handle_error "Invalid OAC ID format: $OAC_ID"
fi

echo "Created Origin Access Control with ID: $OAC_ID"

# Step 5: Create CloudFront distribution
echo "Creating CloudFront distribution..."

# Validate bucket name format
if ! [[ "$BUCKET_NAME" =~ ^[a-z0-9][a-z0-9.-]*[a-z0-9]$ ]]; then
    handle_error "Invalid S3 bucket name format: $BUCKET_NAME"
fi

# Create distribution configuration with improved security settings
cat > distribution-config.json << EOF
{
    "CallerReference": "cli-tutorial-$(date +%s)",
    "Origins": {
        "Quantity": 1,
        "Items": [
            {
                "Id": "S3-$BUCKET_NAME",
                "DomainName": "$BUCKET_NAME.s3.amazonaws.com",
                "S3OriginConfig": {
                    "OriginAccessIdentity": ""
                },
                "OriginAccessControlId": "$OAC_ID"
            }
        ]
    },
    "DefaultCacheBehavior": {
        "TargetOriginId": "S3-$BUCKET_NAME",
        "ViewerProtocolPolicy": "redirect-to-https",
        "AllowedMethods": {
            "Quantity": 2,
            "Items": ["GET", "HEAD"],
            "CachedMethods": {
                "Quantity": 2,
```

```
                "Items": ["GET", "HEAD"]
            }
        },
        "DefaultTTL": 86400,
        "MinTTL": 0,
        "MaxTTL": 31536000,
        "Compress": true,
        "ForwardedValues": {
            "QueryString": false,
            "Cookies": {
                "Forward": "none"
            }
        }
    },
    "Comment": "CloudFront distribution for tutorial",
    "Enabled": true,
    "WebACLId": "",
    "HttpVersion": "http2and3"
}
EOF

# Set restrictive permissions on config file before passing credentials
chmod 600 distribution-config.json

DIST_RESPONSE=$(aws cloudfront create-distribution --distribution-config file://
distribution-config.json)
if [ $? -ne 0 ]; then
    handle_error "Failed to create CloudFront distribution"
fi

DISTRIBUTION_ID=$(echo "$DIST_RESPONSE" | jq -r '.Distribution.Id')
DOMAIN_NAME=$(echo "$DIST_RESPONSE" | jq -r '.Distribution.DomainName')

if [ -z "$DISTRIBUTION_ID" ] || [ "$DISTRIBUTION_ID" = "null" ] || [ -z
 "$DOMAIN_NAME" ] || [ "$DOMAIN_NAME" = "null" ]; then
    handle_error "Failed to extract distribution ID or domain name from response"
fi

# Validate distribution ID format
if ! [[ "$DISTRIBUTION_ID" =~ ^[A-Z0-9]+$ ]]; then
    handle_error "Invalid distribution ID format: $DISTRIBUTION_ID"
fi

echo "Created CloudFront distribution with ID: $DISTRIBUTION_ID"
```

```
echo "CloudFront domain name: $DOMAIN_NAME"

# Tag the CloudFront distribution
aws cloudfront tag-resource --resource "arn:aws:cloudfront::
$ACCOUNT_ID:distribution/$DISTRIBUTION_ID" --tags 'Items=[{Key=project,Value=doc-
smith},{Key=tutorial,Value=cloudfront-gettingstarted}]'

# Step 6: Update S3 bucket policy
echo "Updating S3 bucket policy..."

cat > bucket-policy.json << EOF
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowCloudFrontServicePrincipal",
            "Effect": "Allow",
            "Principal": {
                "Service": "cloudfront.amazonaws.com"
            },
            "Action": "s3:GetObject",
            "Resource": "arn:aws:s3:::$BUCKET_NAME/*",
            "Condition": {
                "StringEquals": {
                    "AWS:SourceArn": "arn:aws:cloudfront::
$ACCOUNT_ID:distribution/$DISTRIBUTION_ID"
                }
            }
        }
    ]
}
EOF

# Set restrictive permissions on policy file
chmod 600 bucket-policy.json

aws s3api put-bucket-policy --bucket "$BUCKET_NAME" --policy file://bucket-
policy.json
if [ $? -ne 0 ]; then
    handle_error "Failed to update S3 bucket policy"
fi

# Step 7: Wait for distribution to deploy
```

```
echo "Waiting for CloudFront distribution to deploy (this may take 5-10
 minutes)..."
aws cloudfront wait distribution-deployed --id "$DISTRIBUTION_ID" 2>/dev/null ||
 {
    echo "Warning: Distribution deployment wait timed out. The distribution may
 still be deploying."
}

echo "CloudFront distribution is now deployed."

# Step 8: Display access information
echo ""
echo "===== CloudFront Distribution Setup Complete ====="
echo "You can access your content at: https://$DOMAIN_NAME/index.html"
echo ""
echo "Resources created:"
echo "- S3 Bucket: $BUCKET_NAME"
echo "- CloudFront Origin Access Control: $OAC_ID"
echo "- CloudFront Distribution: $DISTRIBUTION_ID"
echo ""
echo "To clean up resources, run: cleanup"
echo ""

echo "Tutorial completed at $(date)"
```

- Weitere API-Informationen finden Sie in den folgenden Themen der AWS CLI-Befehlsreferenz.
  - CreateDistribution
  - CreateOriginAccessControl
  - DeleteDistribution
  - DeleteOriginAccessControl
  - GetDistribution
  - GetDistributionConfig
  - GetOriginAccessControl
  - UpdateDistribution
  - WaitDistributionDeployed

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unterVerwendung CloudFront mit einem AWS SDK. Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Erste Schritte mit Firewalls für Webanwendungen

Wie das aussehen kann, sehen Sie am nachfolgenden Beispielcode:

- Erstellen einer Web-ACL
- Fügen Sie eine Regel zum Abgleich von Zeichenketten hinzu
- Fügen Sie verwaltete Regeln hinzu
- Konfigurieren der -Protokollierung
- Bereinigen von Ressourcen

Bash

AWS CLI mit Bash-Skript

> **Note**
>
> Es gibt noch mehr dazu. GitHub Hier finden Sie das vollständige Beispiel und erfahren, wie Sie es im Tutorials und Beispiele-Repository für Entwickler einrichten und ausführen.

```
#!/bin/bash

# AWS WAF Getting Started Script
# This script creates a Web ACL with a string match rule and AWS Managed Rules,
# associates it with a CloudFront distribution, and then cleans up all resources.

set -euo pipefail

# Security: Restrict file permissions
umask 077

# Set up logging with secure file handling
LOG_FILE="waf-tutorial.log"
touch "$LOG_FILE"
```

```
chmod 600 "$LOG_FILE"
exec > >(tee -a "$LOG_FILE") 2>&1

echo "================================================="
echo "AWS WAF Getting Started Tutorial"
echo "================================================="
echo "This script will create AWS WAF resources and associate"
echo "them with a CloudFront distribution."
echo ""

# Maximum number of retries for operations
MAX_RETRIES=3

# Function to handle errors securely
handle_error() {
    local error_msg="$1"
    echo "ERROR: $error_msg" >&2
    echo "Check the log file for details: $LOG_FILE" >&2
    cleanup_resources
    exit 1
}

# Function to validate AWS CLI JSON output
validate_json() {
    local json_string="$1"
    if ! echo "$json_string" | jq empty 2>/dev/null; then
        return 1
    fi
    return 0
}

# Function to safely extract JSON values
extract_json_value() {
    local json_string="$1"
    local key_path="$2"

    if ! validate_json "$json_string"; then
        return 1
    fi

    echo "$json_string" | jq -r "$key_path" 2>/dev/null || return 1
}

# Function to clean up resources securely
```

```
cleanup_resources() {
    echo ""
    echo "=================================================="
    echo "CLEANING UP RESOURCES"
    echo "=================================================="

    if [ -n "${DISTRIBUTION_ID:-}" ] && [ -n "${WEB_ACL_ARN:-}" ]; then
        echo "Disassociating Web ACL from CloudFront distribution..."
        local account_id
        account_id=$(aws sts get-caller-identity --query Account --output text
 2>/dev/null) || account_id=""

        if [ -z "$account_id" ]; then
            echo "Warning: Could not retrieve AWS account ID"
            return
        fi

        local disassociate_result
        disassociate_result=$(aws wafv2 disassociate-web-acl \
            --resource-arn "arn:aws:cloudfront::${account_id}:distribution/
${DISTRIBUTION_ID}" \
            --region us-east-1 2>&1) || true

        if echo "$disassociate_result" | grep -qi "error"; then
            echo "Warning: Failed to disassociate Web ACL: $disassociate_result"
        else
            echo "Web ACL disassociated successfully."
        fi
    fi

    if [ -n "${WEB_ACL_ID:-}" ] && [ -n "${WEB_ACL_NAME:-}" ]; then
        echo "Deleting Web ACL..."

        local get_result
        get_result=$(aws wafv2 get-web-acl \
            --name "$WEB_ACL_NAME" \
            --scope CLOUDFRONT \
            --id "$WEB_ACL_ID" \
            --region us-east-1 2>&1) || true

        if echo "$get_result" | grep -qi "error"; then
            echo "Warning: Failed to get Web ACL for deletion: $get_result"
            echo "You may need to manually delete the Web ACL using the AWS
 Console."
```

```
        else
            local latest_token
            latest_token=$(extract_json_value "$get_result" '.WebACL.LockToken'
 2>/dev/null) || latest_token=""

            if [ -n "$latest_token" ]; then
                local delete_result
                delete_result=$(aws wafv2 delete-web-acl \
                    --name "$WEB_ACL_NAME" \
                    --scope CLOUDFRONT \
                    --id "$WEB_ACL_ID" \
                    --lock-token "$latest_token" \
                    --region us-east-1 2>&1) || true

                if echo "$delete_result" | grep -qi "error"; then
                    echo "Warning: Failed to delete Web ACL: $delete_result"
                    echo "You may need to manually delete the Web ACL using the
 AWS Console."
                else
                    echo "Web ACL deleted successfully."
                fi
            else
                echo "Warning: Could not extract lock token for deletion. You may
 need to manually delete the Web ACL."
            fi
        fi
    fi

    echo "Cleanup process completed."
}

# Security: Trap EXIT to ensure cleanup on any exit
trap cleanup_resources EXIT

# Generate a random identifier for resource names using secure method
RANDOM_ID=$(openssl rand -hex 4) || handle_error "Failed to generate random ID"
WEB_ACL_NAME="MyWebACL-${RANDOM_ID}"
METRIC_NAME="MyWebACLMetrics-${RANDOM_ID}"

echo "Using Web ACL name: $WEB_ACL_NAME"

# Step 1: Create a Web ACL
echo ""
echo "================================================="
```

```
echo "STEP 1: Creating Web ACL"
echo "================================================="

local create_result
create_result=$(aws wafv2 create-web-acl \
    --name "$WEB_ACL_NAME" \
    --scope "CLOUDFRONT" \
    --default-action Allow={} \
    --visibility-config
 "SampledRequestsEnabled=true,CloudWatchMetricsEnabled=true,MetricName=
$METRIC_NAME" \
    --tags Key=project,Value=doc-smith Key=tutorial,Value=aws-waf-gs \
    --region us-east-1 2>&1) || handle_error "Failed to create Web ACL"

if ! validate_json "$create_result"; then
    handle_error "Invalid JSON response from create-web-acl"
fi

# Extract Web ACL ID, ARN, and Lock Token from the response
WEB_ACL_ID=$(extract_json_value "$create_result" '.Summary.Id') || handle_error
 "Failed to extract Web ACL ID"
WEB_ACL_ARN=$(extract_json_value "$create_result" '.Summary.ARN') || handle_error
 "Failed to extract Web ACL ARN"
LOCK_TOKEN=$(extract_json_value "$create_result" '.Summary.LockToken') ||
 handle_error "Failed to extract Lock Token"

echo "Web ACL created successfully with ID: $WEB_ACL_ID"
echo "Lock Token: [REDACTED]"

# Step 2: Add a String Match Rule
echo ""
echo "================================================="
echo "STEP 2: Adding String Match Rule"
echo "================================================="

for ((i=1; i<=MAX_RETRIES; i++)); do
    echo "Attempt $i to add string match rule..."

    # Get the latest lock token before updating
    local get_result
    get_result=$(aws wafv2 get-web-acl \
        --name "$WEB_ACL_NAME" \
        --scope CLOUDFRONT \
        --id "$WEB_ACL_ID" \
```

```
        --region us-east-1 2>&1) || true

    if echo "$get_result" | grep -qi "error"; then
        echo "Warning: Failed to get Web ACL for update: $get_result"
        if [ "$i" -eq "$MAX_RETRIES" ]; then
            handle_error "Failed to get Web ACL after $MAX_RETRIES attempts"
        fi
        sleep 2
        continue
    fi

    if ! validate_json "$get_result"; then
        echo "Warning: Invalid JSON response from get-web-acl"
        if [ "$i" -eq "$MAX_RETRIES" ]; then
            handle_error "Invalid JSON response after $MAX_RETRIES attempts"
        fi
        sleep 2
        continue
    fi

    local latest_token
    latest_token=$(extract_json_value "$get_result" '.WebACL.LockToken' 2>/dev/
null) || true

    if [ -z "$latest_token" ]; then
        echo "Warning: Could not extract lock token for update"
        if [ "$i" -eq "$MAX_RETRIES" ]; then
            handle_error "Failed to extract lock token after $MAX_RETRIES
 attempts"
        fi
        sleep 2
        continue
    fi

    local update_result
    update_result=$(aws wafv2 update-web-acl \
        --name "$WEB_ACL_NAME" \
        --scope "CLOUDFRONT" \
        --id "$WEB_ACL_ID" \
        --lock-token "$latest_token" \
        --default-action Allow={} \
        --rules '[{
            "Name": "UserAgentRule",
            "Priority": 0,
```

```
            "Statement": {
                "ByteMatchStatement": {
                    "SearchString": "MyAgent",
                    "FieldToMatch": {
                        "SingleHeader": {
                            "Name": "user-agent"
                        }
                    },
                    "TextTransformations": [
                        {
                            "Priority": 0,
                            "Type": "NONE"
                        }
                    ],
                    "PositionalConstraint": "EXACTLY"
                }
            },
            "Action": {
                "Count": {}
            },
            "VisibilityConfig": {
                "SampledRequestsEnabled": true,
                "CloudWatchMetricsEnabled": true,
                "MetricName": "UserAgentRuleMetric"
            }
        }]' \
        --visibility-config
 "SampledRequestsEnabled=true,CloudWatchMetricsEnabled=true,MetricName=
$METRIC_NAME" \
        --region us-east-1 2>&1) || true

    if echo "$update_result" | grep -qi "WAFOptimisticLockException"; then
        echo "Optimistic lock exception encountered. Will retry with new lock
 token."
        if [ "$i" -eq "$MAX_RETRIES" ]; then
            handle_error "Failed to add string match rule after $MAX_RETRIES
 attempts"
        fi
        sleep 2
        continue
    elif echo "$update_result" | grep -qi "error"; then
        handle_error "Failed to add string match rule: $update_result"
    else
        echo "String match rule added successfully."
```

```
        break
    fi
done

# Step 3: Add AWS Managed Rules
echo ""
echo "================================================="
echo "STEP 3: Adding AWS Managed Rules"
echo "================================================="

for ((i=1; i<=MAX_RETRIES; i++)); do
    echo "Attempt $i to add AWS Managed Rules..."

    # Get the latest lock token before updating
    local get_result
    get_result=$(aws wafv2 get-web-acl \
        --name "$WEB_ACL_NAME" \
        --scope CLOUDFRONT \
        --id "$WEB_ACL_ID" \
        --region us-east-1 2>&1) || true

    if echo "$get_result" | grep -qi "error"; then
        echo "Warning: Failed to get Web ACL for update: $get_result"
        if [ "$i" -eq "$MAX_RETRIES" ]; then
            handle_error "Failed to get Web ACL after $MAX_RETRIES attempts"
        fi
        sleep 2
        continue
    fi

    if ! validate_json "$get_result"; then
        echo "Warning: Invalid JSON response from get-web-acl"
        if [ "$i" -eq "$MAX_RETRIES" ]; then
            handle_error "Invalid JSON response after $MAX_RETRIES attempts"
        fi
        sleep 2
        continue
    fi

    local latest_token
    latest_token=$(extract_json_value "$get_result" '.WebACL.LockToken' 2>/dev/
null) || true

    if [ -z "$latest_token" ]; then
```

```
        echo "Warning: Could not extract lock token for update"
        if [ "$i" -eq "$MAX_RETRIES" ]; then
            handle_error "Failed to extract lock token after $MAX_RETRIES
attempts"
        fi
        sleep 2
        continue
    fi

    local update_result
    update_result=$(aws wafv2 update-web-acl \
        --name "$WEB_ACL_NAME" \
        --scope "CLOUDFRONT" \
        --id "$WEB_ACL_ID" \
        --lock-token "$latest_token" \
        --default-action Allow={} \
        --rules '[{
            "Name": "UserAgentRule",
            "Priority": 0,
            "Statement": {
                "ByteMatchStatement": {
                    "SearchString": "MyAgent",
                    "FieldToMatch": {
                        "SingleHeader": {
                            "Name": "user-agent"
                        }
                    },
                    "TextTransformations": [
                        {
                            "Priority": 0,
                            "Type": "NONE"
                        }
                    ],
                    "PositionalConstraint": "EXACTLY"
                }
            },
            "Action": {
                "Count": {}
            },
            "VisibilityConfig": {
                "SampledRequestsEnabled": true,
                "CloudWatchMetricsEnabled": true,
                "MetricName": "UserAgentRuleMetric"
            }
```

```
        },
        {
            "Name": "AWS-AWSManagedRulesCommonRuleSet",
            "Priority": 1,
            "Statement": {
                "ManagedRuleGroupStatement": {
                    "VendorName": "AWS",
                    "Name": "AWSManagedRulesCommonRuleSet",
                    "ExcludedRules": []
                }
            },
            "OverrideAction": {
                "Count": {}
            },
            "VisibilityConfig": {
                "SampledRequestsEnabled": true,
                "CloudWatchMetricsEnabled": true,
                "MetricName": "AWS-AWSManagedRulesCommonRuleSet"
            }
        }]' \
        --visibility-config
 "SampledRequestsEnabled=true,CloudWatchMetricsEnabled=true,MetricName=
$METRIC_NAME" \
        --region us-east-1 2>&1) || true

    if echo "$update_result" | grep -qi "WAFOptimisticLockException"; then
        echo "Optimistic lock exception encountered. Will retry with new lock
 token."
        if [ "$i" -eq "$MAX_RETRIES" ]; then
            handle_error "Failed to add AWS Managed Rules after $MAX_RETRIES
 attempts"
        fi
        sleep 2
        continue
    elif echo "$update_result" | grep -qi "error"; then
        handle_error "Failed to add AWS Managed Rules: $update_result"
    else
        echo "AWS Managed Rules added successfully."
        break
    fi
done

# Step 4: List CloudFront distributions
echo ""
```

```
echo "=================================================="
echo "STEP 4: Listing CloudFront Distributions"
echo "=================================================="

local cf_result
cf_result=$(aws cloudfront list-distributions --query "DistributionList.Items[*].
{Id:Id,DomainName:DomainName}" --output table 2>&1) || cf_result=""

if echo "$cf_result" | grep -qi "error"; then
    echo "Warning: Failed to list CloudFront distributions: $cf_result"
    echo "Continuing without CloudFront association."
    DISTRIBUTION_ID=""
else
    echo "$cf_result"

    # Auto-select first CloudFront distribution if available
    echo ""
    echo "=================================================="
    echo "STEP 5: Associate Web ACL with CloudFront Distribution"
    echo "=================================================="

    local first_dist
    first_dist=$(aws cloudfront list-distributions --query
 "DistributionList.Items[0].Id" --output text 2>&1) || first_dist=""

    if [ -n "$first_dist" ] && [ "$first_dist" != "None" ] && ! echo
 "$first_dist" | grep -qi "error"; then
        DISTRIBUTION_ID="$first_dist"
        echo "Using CloudFront distribution: $DISTRIBUTION_ID"

        local account_id
        account_id=$(aws sts get-caller-identity --query Account --output text
 2>/dev/null) || account_id=""

        if [ -z "$account_id" ]; then
            echo "Warning: Could not retrieve AWS account ID for association"
            DISTRIBUTION_ID=""
        else
            local associate_result
            associate_result=$(aws wafv2 associate-web-acl \
                --web-acl-arn "$WEB_ACL_ARN" \
                --resource-arn "arn:aws:cloudfront::${account_id}:distribution/
${DISTRIBUTION_ID}" \
                --region us-east-1 2>&1) || true
```

```
            if echo "$associate_result" | grep -qi "error"; then
                echo "Warning: Failed to associate Web ACL with CloudFront
 distribution: $associate_result"
                echo "Continuing without CloudFront association."
                DISTRIBUTION_ID=""
            else
                echo "Web ACL associated with CloudFront distribution
 successfully."
            fi
        fi
    else
        echo "No CloudFront distributions available. Skipping association."
        DISTRIBUTION_ID=""
    fi
fi

# Display summary of created resources
echo ""
echo "=================================================="
echo "RESOURCE SUMMARY"
echo "=================================================="
echo "Web ACL Name: $WEB_ACL_NAME"
echo "Web ACL ID: $WEB_ACL_ID"
echo "Web ACL ARN: $WEB_ACL_ARN"
if [ -n "${DISTRIBUTION_ID:-}" ]; then
    echo "Associated CloudFront Distribution: $DISTRIBUTION_ID"
fi
echo ""

# Auto-confirm cleanup
echo "=================================================="
echo "CLEANUP CONFIRMATION"
echo "=================================================="
echo "Proceeding with automatic cleanup of all created resources..."

echo ""
echo "=================================================="
echo "Tutorial completed!"
echo "=================================================="
echo "Log file: $LOG_FILE"
```

- Weitere API-Informationen finden Sie in den folgenden Themen der AWS CLI-Befehlsreferenz.
  - AssociateWebAcl
  - CreateWebAcl
  - DeleteWebAcl
  - DisassociateWebAcl
  - GetCallerIdentity
  - GetWebAcl
  - ListDistributions
  - UpdateWebAcl

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unterVerwendung CloudFront mit einem AWS SDK. Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Erstellen Sie signierte URLs und Cookies mit einem AWS SDK

Das folgende Codebeispiel zeigt, wie signierte URLs und Cookies erstellt werden, die den Zugriff auf eingeschränkte Ressourcen ermöglichen.

Java

SDK für Java 2.x

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS-Code-Beispiel- einrichten und ausführen.

Verwenden Sie den CannedSignerRequestKurs, um URLs oder Cookies mit einer vorgefertigten Richtlinie zu signieren.

```
import software.amazon.awssdk.services.cloudfront.model.CannedSignerRequest;

import java.net.URL;
import java.nio.file.Path;
```

```
import java.nio.file.Paths;
import java.time.Instant;
import java.time.temporal.ChronoUnit;

public class CreateCannedPolicyRequest {

    public static CannedSignerRequest createRequestForCannedPolicy(String
 distributionDomainName,
            String fileNameToUpload,
            String privateKeyFullPath, String publicKeyId) throws Exception {
        String protocol = "https";
        String resourcePath = "/" + fileNameToUpload;

        String cloudFrontUrl = new URL(protocol, distributionDomainName,
 resourcePath).toString();
        Instant expirationDate = Instant.now().plus(7, ChronoUnit.DAYS);
        Path path = Paths.get(privateKeyFullPath);

        return CannedSignerRequest.builder()
                .resourceUrl(cloudFrontUrl)
                .privateKey(path)
                .keyPairId(publicKeyId)
                .expirationDate(expirationDate)
                .build();
    }
}
```

Verwenden Sie die CustomSignerRequestKlasse, um URLs oder Cookies mit einer benutzerdefinierten Richtlinie zu signieren. Die Methoden `activeDate` und `ipRange` sind optional.

```
import software.amazon.awssdk.services.cloudfront.model.CustomSignerRequest;

import java.net.URL;
import java.nio.file.Path;
import java.nio.file.Paths;
import java.time.Instant;
import java.time.temporal.ChronoUnit;

public class CreateCustomPolicyRequest {
```

```
    public static CustomSignerRequest createRequestForCustomPolicy(String
 distributionDomainName,
            String fileNameToUpload,
            String privateKeyFullPath, String publicKeyId) throws Exception {
        String protocol = "https";
        String resourcePath = "/" + fileNameToUpload;

        String cloudFrontUrl = new URL(protocol, distributionDomainName,
 resourcePath).toString();
        Instant expireDate = Instant.now().plus(7, ChronoUnit.DAYS);
        // URL will be accessible tomorrow using the signed URL.
        Instant activeDate = Instant.now().plus(1, ChronoUnit.DAYS);
        Path path = Paths.get(privateKeyFullPath);

        return CustomSignerRequest.builder()
                .resourceUrl(cloudFrontUrl)
                // .resourceUrlPattern("https://*.example.com/*")  // Optional.
                .privateKey(path)
                .keyPairId(publicKeyId)
                .expirationDate(expireDate)
                .activeDate(activeDate) // Optional.
                // .ipRange("192.168.0.1/24") // Optional.
                .build();
    }
}
```

Das folgende Beispiel zeigt, wie die CloudFrontUtilitiesKlasse verwendet wird, um signierte Cookies und URLs zu erzeugen. Sehen Sie sich dieses Codebeispiel unter an GitHub.

```
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import software.amazon.awssdk.services.cloudfront.CloudFrontUtilities;
import software.amazon.awssdk.services.cloudfront.cookie.CookiesForCannedPolicy;
import software.amazon.awssdk.services.cloudfront.cookie.CookiesForCustomPolicy;
import software.amazon.awssdk.services.cloudfront.model.CannedSignerRequest;
import software.amazon.awssdk.services.cloudfront.model.CustomSignerRequest;
import software.amazon.awssdk.services.cloudfront.url.SignedUrl;

public class SigningUtilities {
    private static final Logger logger =
 LoggerFactory.getLogger(SigningUtilities.class);
```

```
   private static final CloudFrontUtilities cloudFrontUtilities =
CloudFrontUtilities.create();

   public static SignedUrl signUrlForCannedPolicy(CannedSignerRequest
cannedSignerRequest) {
       SignedUrl signedUrl =
cloudFrontUtilities.getSignedUrlWithCannedPolicy(cannedSignerRequest);
       logger.info("Signed URL: [{}]", signedUrl.url());
       return signedUrl;
   }

   public static SignedUrl signUrlForCustomPolicy(CustomSignerRequest
customSignerRequest) {
       SignedUrl signedUrl =
cloudFrontUtilities.getSignedUrlWithCustomPolicy(customSignerRequest);
       logger.info("Signed URL: [{}]", signedUrl.url());
       return signedUrl;
   }

   public static CookiesForCannedPolicy
getCookiesForCannedPolicy(CannedSignerRequest cannedSignerRequest) {
       CookiesForCannedPolicy cookiesForCannedPolicy = cloudFrontUtilities
               .getCookiesForCannedPolicy(cannedSignerRequest);
       logger.info("Cookie EXPIRES header [{}]",
cookiesForCannedPolicy.expiresHeaderValue());
       logger.info("Cookie KEYPAIR header [{}]",
cookiesForCannedPolicy.keyPairIdHeaderValue());
       logger.info("Cookie SIGNATURE header [{}]",
cookiesForCannedPolicy.signatureHeaderValue());
       return cookiesForCannedPolicy;
   }

   public static CookiesForCustomPolicy
getCookiesForCustomPolicy(CustomSignerRequest customSignerRequest) {
       CookiesForCustomPolicy cookiesForCustomPolicy = cloudFrontUtilities
               .getCookiesForCustomPolicy(customSignerRequest);
       logger.info("Cookie POLICY header [{}]",
cookiesForCustomPolicy.policyHeaderValue());
       logger.info("Cookie KEYPAIR header [{}]",
cookiesForCustomPolicy.keyPairIdHeaderValue());
       logger.info("Cookie SIGNATURE header [{}]",
cookiesForCustomPolicy.signatureHeaderValue());
       return cookiesForCustomPolicy;
   }
```

```
}
```

- Einzelheiten zur API finden Sie CloudFrontUtilitiesin der AWS SDK for Java 2.x API-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unterVerwendung CloudFront mit einem AWS SDK. Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

CloudFront Funktionen, Beispiele für CloudFront

Die folgenden Codebeispiele zeigen, wie die Verwendung CloudFront mit AWS SDKs funktioniert.

Beispiele

- Fügen Sie einem CloudFront Functions-Viewer-Antwortereignis HTTP-Sicherheitsheader hinzu
- Fügen Sie einem CloudFront Functions-Viewer-Antwortereignis einen CORS-Header hinzu
- Fügen Sie einem Antwortereignis des CloudFront Functions-Viewers einen Cache-Control-Header hinzu
- Fügen Sie einem CloudFront Functions-Viewer-Anforderungsereignis einen echten Client-IP-Header hinzu
- Fügen Sie einem CloudFront Functions-Viewer-Anforderungsereignis einen Origin-Header hinzu
- Fügen Sie index.html hinzu, um URLs ohne Dateinamen in einem CloudFront Functions-Viewer-Anforderungsereignis anzufordern
- Normalisieren Sie die Parameter der Abfragezeichenfolge in einer CloudFront Functions Viewer-Anforderung
- In einem CloudFront Functions-Viewer-Anforderungsereignis zu einer neuen URL umleiten
- Schreiben Sie einen Anforderungs-URI auf der Grundlage der KeyValueStore Konfiguration für ein CloudFront Functions-Viewer-Anforderungsereignis neu
- Leiten Sie Anfragen in einem CloudFront Functions-Viewer-Anforderungsereignis an einen Ursprung weiter, der näher am Betrachter liegt
- Verwenden Sie Schlüssel-Wert-Paare in einer CloudFront Functions-Viewer-Anfrage
- Validieren Sie ein einfaches Token in einer CloudFront Functions-Viewer-Anfrage

Fügen Sie einem CloudFront Functions-Viewer-Antwortereignis HTTP-Sicherheitsheader hinzu

Das folgende Codebeispiel zeigt, wie einem Antwortereignis des CloudFront Functions-Viewers HTTP-Sicherheitsheader hinzugefügt werden.

JavaScript

JavaScript Runtime 2.0 für CloudFront Funktionen

> **Note**
>
> Es gibt noch mehr dazu GitHub. Das vollständige Beispiel und Informationen zur Einrichtung und Ausführung finden Sie im CloudFront Functions-Beispiel-Repository.

```
async function handler(event) {
    var response = event.response;
    var headers = response.headers;

    // Set HTTP security headers
    // Since JavaScript doesn't allow for hyphens in variable names, we use the
 dict["key"] notation
    headers['strict-transport-security'] = { value: 'max-age=63072000;
 includeSubdomains; preload'};
    headers['content-security-policy'] = { value: "default-src 'none'; img-src
 'self'; script-src 'self'; style-src 'self'; object-src 'none'; frame-ancestors
 'none'"};
    headers['x-content-type-options'] = { value: 'nosniff'};
    headers['x-frame-options'] = {value: 'DENY'};
    headers['x-xss-protection'] = {value: '1; mode=block'};
    headers['referrer-policy'] = {value: 'same-origin'};

    // Return the response to viewers
    return response;
}
```

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unterVerwendung CloudFront mit einem AWS SDK. Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Fügen Sie einem CloudFront Functions-Viewer-Antwortereignis einen CORS-Header hinzu

Das folgende Codebeispiel zeigt, wie ein CORS-Header zu einem Antwortereignis des CloudFront Functions-Viewers hinzugefügt wird.

JavaScript

JavaScript Runtime 2.0 für CloudFront Funktionen

> **Note**
>
> Es gibt noch mehr dazu GitHub. Das vollständige Beispiel und Informationen zur Einrichtung und Ausführung finden Sie im CloudFront Functions-Beispiel-Repository.

```
async function handler(event)  {
    var request = event.request;
    var response  = event.response;

    // If Access-Control-Allow-Origin CORS header is missing, add it.
    // Since JavaScript doesn't allow for hyphens in variable names, we use the
 dict["key"] notation.
    if (!response.headers['access-control-allow-origin'] &&
 request.headers['origin']) {
        response.headers['access-control-allow-origin'] = {value:
 request.headers['origin'].value};
        console.log("Access-Control-Allow-Origin was missing, adding it now.");
    }

    return response;
}
```

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unterVerwendung CloudFront mit einem AWS SDK. Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Fügen Sie einem Antwortereignis des CloudFront Functions-Viewers einen Cache-Control-Header hinzu

Das folgende Codebeispiel zeigt, wie ein Cache-Control-Header zu einem Antwortereignis im CloudFront Functions-Viewer hinzugefügt wird.

JavaScript

JavaScript Runtime 2.0 für CloudFront Funktionen

> **Note**
>
> Es gibt noch mehr dazu GitHub. Das vollständige Beispiel und Informationen zur Einrichtung und Ausführung finden Sie im CloudFront Functions-Beispiel-Repository.

```
async function handler(event) {
    var response = event.response;
    var headers = response.headers;

    if (response.statusCode >= 200 && response.statusCode < 400) {
        // Set the cache-control header
        headers['cache-control'] = {value: 'public, max-age=63072000'};
    }

    // Return response to viewers
    return response;
}
```

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unterVerwendung CloudFront mit einem AWS SDK. Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Fügen Sie einem CloudFront Functions-Viewer-Anforderungsereignis einen echten Client-IP-Header hinzu

Das folgende Codebeispiel zeigt, wie einem CloudFront Functions-Viewer-Anforderungsereignis ein echter Client-IP-Header hinzugefügt wird.

JavaScript

JavaScript Runtime 2.0 für CloudFront Funktionen

> **Note**
>
> Es gibt noch mehr dazu GitHub. Das vollständige Beispiel und Informationen zur Einrichtung und Ausführung finden Sie im CloudFront Functions-Beispiel-Repository.

```
async function handler(event) {
    var request = event.request;
    var clientIP = event.viewer.ip;

    //Add the true-client-ip header to the incoming request
    request.headers['true-client-ip'] = {value: clientIP};

    return request;
}
```

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unterVerwendung CloudFront mit einem AWS SDK. Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Fügen Sie einem CloudFront Functions-Viewer-Anforderungsereignis einen Origin-Header hinzu

Das folgende Codebeispiel zeigt, wie ein Origin-Header zu einem CloudFront Functions-Viewer-Anforderungsereignis hinzugefügt wird.

JavaScript

JavaScript Runtime 2.0 für CloudFront Funktionen

> **Note**
>
> Es gibt noch mehr dazu GitHub. Das vollständige Beispiel und Informationen zur Einrichtung und Ausführung finden Sie im CloudFront Functions-Beispiel-Repository.

```
async function handler(event) {
    var request = event.request;
    var headers = request.headers;
    var host = request.headers.host.value;

   // If origin header is missing, set it equal to the host header.
   if (!headers.origin)
       headers.origin = {value:`https://${host}`};

   return request;
}
```

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unterVerwendung CloudFront mit einem AWS SDK. Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Fügen Sie index.html hinzu, um URLs ohne Dateinamen in einem CloudFront Functions-Viewer-Anforderungsereignis anzufordern

Das folgende Codebeispiel zeigt, wie index.html hinzugefügt wird, um URLs ohne Dateinamen in einem CloudFront Functions-Viewer-Anforderungsereignis anzufordern.

JavaScript

JavaScript Runtime 2.0 für CloudFront Funktionen

> **Note**
>
> Es gibt noch mehr dazu GitHub. Das vollständige Beispiel und Informationen zur Einrichtung und Ausführung finden Sie im CloudFront Functions-Beispiel-Repository.

```
async function handler(event) {
    var request = event.request;
    var uri = request.uri;

    // Check whether the URI is missing a file name.
    if (uri.endsWith('/')) {
        request.uri += 'index.html';
    }
    // Check whether the URI is missing a file extension.
    else if (!uri.includes('.')) {
        request.uri += '/index.html';
    }

    return request;
}
```

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unterVerwendung CloudFront mit einem AWS SDK. Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Normalisieren Sie die Parameter der Abfragezeichenfolge in einer CloudFront Functions Viewer-Anforderung

Das folgende Codebeispiel zeigt, wie Abfragezeichenfolgenparameter in einer CloudFront Functions Viewer-Anfrage normalisiert werden.

JavaScript

JavaScript Runtime 2.0 für CloudFront Funktionen

> **Note**
>
> Es gibt noch mehr dazu GitHub. Das vollständige Beispiel und Informationen zur Einrichtung und Ausführung finden Sie im CloudFront Functions-Beispiel-Repository.

```
function handler(event) {
     var qs=[];
     for (var key in event.request.querystring) {
         if (event.request.querystring[key].multiValue) {
             event.request.querystring[key].multiValue.forEach((mv) =>
 {qs.push(key + "=" + mv.value)});
         } else {
             qs.push(key + "=" + event.request.querystring[key].value);
         }
     };

     event.request.querystring = qs.sort().join('&');


     return event.request;
}
```

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unterVerwendung CloudFront mit einem AWS SDK. Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

In einem CloudFront Functions-Viewer-Anforderungsereignis zu einer neuen URL umleiten

Das folgende Codebeispiel zeigt, wie in einem CloudFront Functions-Viewer-Anforderungsereignis zu einer neuen URL umgeleitet wird.

JavaScript

JavaScript Runtime 2.0 für CloudFront Funktionen

> **Note**
>
> Es gibt noch mehr dazu GitHub. Das vollständige Beispiel und Informationen zur Einrichtung und Ausführung finden Sie im CloudFront Functions-Beispiel-Repository.

```
async function handler(event) {
    var request = event.request;
    var headers = request.headers;
    var host = request.headers.host.value;
    var country = 'DE' // Choose a country code
    var newurl = `https://${host}/de/index.html`; // Change the redirect URL to
 your choice

    if (headers['cloudfront-viewer-country']) {
        var countryCode = headers['cloudfront-viewer-country'].value;
        if (countryCode === country) {
            var response = {
                statusCode: 302,
                statusDescription: 'Found',
                headers:
                    { "location": { "value": newurl } }
                }

            return response;
        }
    }
    return request;
}
```

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unterVerwendung CloudFront mit einem AWS SDK. Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Schreiben Sie einen Anforderungs-URI auf der Grundlage der KeyValueStore Konfiguration für ein CloudFront Functions-Viewer-Anforderungsereignis neu

Das folgende Codebeispiel zeigt, wie eine Anforderungs-URI auf der Grundlage der KeyValueStore Konfiguration für ein CloudFront Functions-Viewer-Anforderungsereignis neu geschrieben wird.

JavaScript

JavaScript Runtime 2.0 für CloudFront Funktionen

> **Note**
>
> Es gibt noch mehr dazu GitHub. Das vollständige Beispiel und Informationen zur Einrichtung und Ausführung finden Sie im CloudFront Functions-Beispiel-Repository.

```
import cf from 'cloudfront';

// (Optional) Replace KVS_ID with actual KVS ID
const kvsId = "KVS_ID";
// enable stickiness by setting a cookie from origin or using another edge
 function
const stickinessCookieName = "appversion";
// set to true to enable console logging
const loggingEnabled = false;

// function rewrites the request uri based on configuration in KVS
// example config in KVS in key:value format
// "latest": {"a_weightage": .8, "a_url": "v1", "b_url": "v2"}
// given above key and value in KVS the request uri will be rewritten
// for example http(s)://domain/latest/something/else will be rewritten as
 http(s)://domain/v1/something/else or http(s)://domain/v2/something/else
 depending on weightage
// if no configuration is found, then the request is returned as is
async function handler(event) {
    // NOTE: This example function is for a viewer request event trigger.
    // Choose viewer request for event trigger when you associate this function
 with a distribution.
    const request = event.request;
    const pathSegments = request.uri.split('/');
```

```
    const key = pathSegments[1];

    // if empty path segment or if there is valid stickiness cookie
    // then skip call to KVS and let the request continue.
    if (!key || hasValidSticknessCookie(request.cookies[stickinessCookieName],
 key)) {
        return event.request;
    }

    try {
        // get the prefix replacement from KVS
        const replacement = await getPathPrefixByWeightage(key);
        if (!replacement) {
            return event.request;
        }
        //Replace the first path with the replacement
        pathSegments[1] = replacement;
        log(`using prefix ${pathSegments[1]}`)
        const newUri = pathSegments.join('/');
        log(`${request.uri} -> ${newUri}`);
        request.uri = newUri;

        return request;
    } catch (err) {
        // No change to the path if the key is not found or any other error
        log(`request uri: ${request.uri}, error: ${err}`);
    }
    // no change to path - return request
    return event.request;
}

// function to get the prefix from KVS
async function getPathPrefixByWeightage(key) {
    const kvsHandle = cf.kvs(kvsId);
    // get the weightage config from KVS
    const kvsResponse = await kvsHandle.get(key);
    const weightageConfig = JSON.parse(kvsResponse);
    // no configuration - return null
    if (!weightageConfig || !isFinite(weightageConfig.a_weightage)) {
        return null;
    }
    // return the url based on weightage
    // return null if no url is configured
    if (Math.random() <= weightageConfig.a_weightage) {
```

```
        return weightageConfig.a_url ? weightageConfig.a_url: null;
    } else {
        return weightageConfig.b_url ? weightageConfig.b_url : null;
    }
}

// function to check if the stickiness cookie is valid
function hasValidStickinessCookie(stickinessCookie, pathSegment) {
    // if the value exists and it matches pathSegment
    return (stickinessCookie && stickinessCookie.value === pathSegment)
}

function log(message) {
    if (loggingEnabled) {
        console.log(message);
    }
}
```

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unterVerwendung CloudFront mit einem AWS SDK. Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Leiten Sie Anfragen in einem CloudFront Functions-Viewer-Anforderungsereignis an einen Ursprung weiter, der näher am Betrachter liegt

Das folgende Codebeispiel zeigt, wie Anfragen in einem CloudFront Functions-Viewer-Anforderungsereignis an einen Ursprung weitergeleitet werden, der näher am Betrachter liegt.

JavaScript

JavaScript Runtime 2.0 für CloudFront Funktionen

> **Note**
>
> Es gibt noch mehr dazu GitHub. Das vollständige Beispiel und Informationen zur Einrichtung und Ausführung finden Sie im CloudFront Functions-Beispiel-Repository.

```
import cf from 'cloudfront';

function handler(event) {
    const request = event.request;
    const headers = request.headers;
    const country = headers['cloudfront-viewer-country'] &&
        headers['cloudfront-viewer-country'].value;

    //List of Regions with S3 buckets containing content
    const countryToRegion = {
        'DE': 'eu-central-1',
        'IE': 'eu-west-1',
        'GB': 'eu-west-2',
        'FR': 'eu-west-3',
        'JP': 'ap-northeast-1',
        'IN': 'ap-south-1'
    };

    const DEFAULT_REGION = 'us-east-1';

    const selectedRegion = (country && countryToRegion[country]) ||
 DEFAULT_REGION;

    const domainName =
        `cloudfront-functions-demo-bucket-in-${selectedRegion}.s3.
${selectedRegion}.amazonaws.com`;

    cf.updateRequestOrigin({
        "domainName": domainName,
        "originAccessControlConfig": {
            "enabled": true,
            "region": selectedRegion,
            "signingBehavior": "always",
            "signingProtocol": "sigv4",
            "originType": "s3"
        },
    });

    return request;
}
```

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unterVerwendung CloudFront mit einem AWS SDK. Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie Schlüssel-Wert-Paare in einer CloudFront Functions-Viewer-Anfrage

Das folgende Codebeispiel zeigt, wie Schlüssel-Wert-Paare in einer CloudFront Functions-Viewer-Anfrage verwendet werden.

JavaScript

JavaScript Runtime 2.0 für CloudFront Funktionen

> **Note**
>
> Es gibt noch mehr dazu GitHub. Das vollständige Beispiel und Informationen zur Einrichtung und Ausführung finden Sie im CloudFront Functions-Beispiel-Repository.

```
import cf from 'cloudfront';

// This fails if there is no key value store associated with the function
const kvsHandle = cf.kvs();

// Remember to associate the KVS with your function before referencing KVS in
 your code.
// https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/kvs-with-
functions-associate.html
async function handler(event) {
    const request = event.request;
    // Use the first segment of the pathname as key
    // For example http(s)://domain/<key>/something/else
    const pathSegments = request.uri.split('/')
    const key = pathSegments[1]
    try {
        // Replace the first path of the pathname with the value of the key
        // For example http(s)://domain/<value>/something/else
        pathSegments[1] = await kvsHandle.get(key);
        const newUri = pathSegments.join('/');
        console.log(`${request.uri} -> ${newUri}`)
```

```
        request.uri = newUri;
    } catch (err) {
        // No change to the pathname if the key is not found
        console.log(`${request.uri} | ${err}`);
    }
    return request;
}
```

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unterVerwendung CloudFront mit einem AWS SDK. Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Validieren Sie ein einfaches Token in einer CloudFront Functions-Viewer-Anfrage

Das folgende Codebeispiel zeigt, wie ein einfaches Token in einer CloudFront Functions Viewer-Anfrage validiert wird.

JavaScript

JavaScript Runtime 2.0 für CloudFront Funktionen

> **Note**
>
> Es gibt noch mehr dazu GitHub. Das vollständige Beispiel und Informationen zur Einrichtung und Ausführung finden Sie im CloudFront Functions-Beispiel-Repository.

```
import crypto from 'crypto';
import cf from 'cloudfront';

//Response when JWT is not valid.
const response401 = {
    statusCode: 401,
    statusDescription: 'Unauthorized'
};

// Remember to associate the KVS with your function before calling the const
 kvsKey = 'jwt.secret'.
```

```
// https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/kvs-with-
functions-associate.html
const kvsKey = 'jwt.secret';
// set to true to enable console logging
const loggingEnabled = false;


function jwt_decode(token, key, noVerify, algorithm) {
    // check token
    if (!token) {
        throw new Error('No token supplied');
    }
    // check segments
    const segments = token.split('.');
    if (segments.length !== 3) {
        throw new Error('Not enough or too many segments');
    }

    // All segment should be base64
    const headerSeg = segments[0];
    const payloadSeg = segments[1];
    const signatureSeg = segments[2];

    // base64 decode and parse JSON
    const payload = JSON.parse(_base64urlDecode(payloadSeg));

    if (!noVerify) {
        const signingMethod = 'sha256';
        const signingType = 'hmac';

        // Verify signature. `sign` will return base64 string.
        const signingInput = [headerSeg, payloadSeg].join('.');

        if (!_verify(signingInput, key, signingMethod, signingType,
 signatureSeg)) {
            throw new Error('Signature verification failed');
        }

        // Support for nbf and exp claims.
        // According to the RFC, they should be in seconds.
        if (payload.nbf && Date.now() < payload.nbf*1000) {
            throw new Error('Token not yet active');
        }
```

```
        if (payload.exp && Date.now() > payload.exp*1000) {
            throw new Error('Token expired');
        }
    }

    return payload;
}

//Function to ensure a constant time comparison to prevent
//timing side channels.
function _constantTimeEquals(a, b) {
    if (a.length != b.length) {
        return false;
    }

    let xor = 0;
    for (let i = 0; i < a.length; i++) {
    xor |= (a.charCodeAt(i) ^ b.charCodeAt(i));
    }

    return 0 === xor;
}

function _verify(input, key, method, type, signature) {
    if(type === "hmac") {
        return _constantTimeEquals(signature, _sign(input, key, method));
    }
    else {
        throw new Error('Algorithm type not recognized');
    }
}

function _sign(input, key, method) {
    return crypto.createHmac(method, key).update(input).digest('base64url');
}

function _base64urlDecode(str) {
    return Buffer.from(str, 'base64url')
}

async function handler(event) {
    let request = event.request;

    //Secret key used to verify JWT token.
```

```
    //Update with your own key.
    const secret_key = await getSecret()

    if(!secret_key) {
        return response401;
    }

    // If no JWT token, then generate HTTP redirect 401 response.
    if(!request.querystring.jwt) {
        log("Error: No JWT in the querystring");
        return response401;
    }

    const jwtToken = request.querystring.jwt.value;

    try{
        jwt_decode(jwtToken, secret_key);
    }
    catch(e) {
        log(e);
        return response401;
    }

    //Remove the JWT from the query string if valid and return.
    delete request.querystring.jwt;
    log("Valid JWT token");
    return request;
}

// get secret from key value store
async function getSecret() {
    // initialize cloudfront kv store and get the key value
    try {
        const kvsHandle = cf.kvs();
        return await kvsHandle.get(kvsKey);
    } catch (err) {
        log(`Error reading value for key: ${kvsKey}, error: ${err}`);
        return null;
    }

}

function log(message) {
    if (loggingEnabled) {
```

```
        console.log(message);
    }
}
```

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unterVerwendung CloudFront mit einem AWS SDK. Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Dokumentverlauf

In der folgenden Tabelle werden die wichtigen Änderungen an der Dokumentation beschrieben. CloudFront Um Benachrichtigungen über Aktualisierungen zu erhalten, können Sie den RSS-Feed abonnieren.

Änderung	Beschreibung	Datum
Gegenseitiges TLS (Viewer) hinzugefügt	CloudFront unterstützt gegenseitiges TLS (Viewer).	24. November 2025
Protokollfeld für Zugriffsprotokolle hinzugefügt	Das `connection-id` Feld für Zugriffsprotokolle (Standardprotokolle) und Echtzeitzugriffsprotokolle wurde hinzugefügt.	24. November 2025
Verbindungsprotokolle hinzugefügt	Verbindungsprotokolle als neue Protokollierungsfunktion für gegenseitiges TLS (Viewer) hinzugefügt.	24. November 2025
Verbindungsfunktionen hinzugefügt	CloudFront unterstützt Verbindungsfunktionen für gegenseitiges TLS (Viewer).	24. November 2025
Fügen Sie Unterstützung für die Verwendung von IPAM hinzu, um Ihre eigene IP-Adresse zu CloudFront verwenden	CloudFront unterstützt das Einbinden Ihrer eigenen IPv4-Adressen über das BYOIP von IPAM für globale Dienste.	24. November 2025
AWS verwaltetes Richtlinien-Update	Unterstützung `CloudFrontReadOnlyAccess` und `ec2:GetIpamPoolCidrs` Aktionen für die Richtlinien und `CloudFrontFullAcce`	24. November 2025

	`ss` IAM sind jetzt `ec2:Descr ibeIpamPools` verfügbar.	
Aktualisierungen der CloudFront Funktionen	In dieser Version werden Parameter für die Hilfsmeth oden zur Änderung des Ursprungs in CloudFron t Functions hinzugefü gt. Sie können die `origin0verrides` Parameter `hostHeade r sni`,`allowedCe rtificateNames` , und verwenden.	20. November 2025
Aktualisierungen der CloudFront Funktionen	Unterstützung für CBOR Web Tokens (CWT) für CloudFront Funktionen hinzugefügt.	20. November 2025
Aktualisierungen der CloudFront Funktionen	Allgemeine Hilfsmethoden für CloudFront Funktionen hinzugefügt.	20. November 2025
Aktualisierte AWS verwaltete Richtlinie	Die Aktualisierung `CloudFron tFullAccess` gewährt nun die Berechtigung, eine AWS WAF Web-ACL-Ressource zu erstellen und den AWS Pricing Plan Manager zu erstellen, zu aktualisieren, zu löschen und zu lesen.	18. November 2025
Aktualisierte AWS verwaltete Richtlinie	Die Aktualisierung `CloudFron tReadOnlyAccess` gewährt AWS Pricing Plan Manager nun nur noch Lesezugriff.	18. November 2025

Aktualisierte AWS verwaltete Richtlinie	Die Aktualisierung `CloudFrontFullAccess` gewährt nun die Berechtigung, eine AWS WAF Web-ACL-Ressource zu erstellen und den AWS Pricing Plan Manager zu erstellen, zu aktualisieren, zu löschen und zu lesen.	18. November 2025
Aktualisierte AWS verwaltete Richtlinie	Die Aktualisierung `CloudFrontReadOnlyAccess` gewährt AWS Pricing Plan Manager nun nur noch Lesezugriff.	18. November 2025
CloudFront unterstützt Flatrate-Preispläne	Sie können jetzt für Ihre Distributionen einen CloudFront Pauschalpreis abonnieren.	18. November 2025
Statische Anycast-IPs	Sie können jetzt zwischen nur IPv4-Adressen oder sowohl IPv4- als auch IPv6-Adressen (Dualstack) wählen.	5. November 2025

Unterstützung für die gemeinsame Nutzung von VPC-Ursprüngen zwischen AWS-Konten	Sie können eine Ressource nfreigabe erstellen und ihr VPC-Ursprünge hinzufüge n. Auf diese Weise können Sie Ihre VPC-Ursprünge und CloudFront -Distributionen getrennt halten AWS-Konte n, sodass Ihr Unternehmen die Anforderungen mehrerer Konten erfüllen kann. Andere AWS-Konten können die gemeinsamen VPC-Urspr ünge für ihre Distributionen zuordnen. CloudFront	5. November 2025
Sicherheitsrichtlinie für Viewer hinzugefügt	Die Sicherheitsrichtlinie TLSv1.2 _2025 wurde hinzugefügt. • Unterstützte Protokolle und Chiffren zwischen Zuschauern und CloudFront • Sicherheitsrichtlinie (SSL/ TLS Mindestversion)	10. Oktober 2025
Post Quantum (PQ)-Algo rithmen für den Schlüssel austausch hinzugefügt	Den CloudFront TLS-Richt linien wurden zwei neue PQ-Algorithmen für den Schlüssel austausch hinzugefügt.	4. September 2025

IPv6-Ursprungsanforderungen	Wenn Sie einen benutzerd efinierten Ursprung verwenden (mit Ausnahme von Amazon S3- und VPC-Ursprüngen), können Sie die Ursprungs einstellungen für Ihre Distribut ion anpassen, um auszuwähl en, wie die CloudFront Verbindung zu Ihrem Ursprung mithilfe von IPv4- oder IPv6-Adressen hergestellt wird.	3. September 2025
Neue verwaltete Ursprungs anforderungsrichtlinie hinzugefügt	Die neue verwaltete Ursprungs anforderungsrichtlinie `HostHeaderOnly` wurde hinzugefügt.	29. August 2025
CloudFront Öffentliche Endpunkte unterstützen jetzt IPv6	Weitere Informationen zu CloudFront Amazon-En dpunkten und -Kontingenten finden Sie im Amazon VPC-Benutzerhandbuch in der Allgemeine AWS-Referenz und AWS-Services die IPv6 unterstützen.	21. August 2025
Sicherheitsrichtlinie für Viewer hinzugefügt	TLSv1.3_2025, eine neue Sicherheitsrichtlinie nur für TLS 1.3, hinzugefügt.	7. August 2025
Neue Einstellungen für das Ursprungs-Timeout hinzugefü gt	Das Timeout für den Abschluss der Antwort wurde für alle Ursprünge hinzugefü gt und das Antwort-Timeout (Ursprungs-Lese-Timeout) für S3-Ursprünge.	30. Juli 2025

Vorkonfigurierte Standardd istributionseinstellungen hinzugefügt	Vorkonfigurierte Einstellungen für Standarddistributionen wurden hinzugefügt.	17. Juni 2025
Neuer Konsolen-Workflow für die Einrichtung von Standardd istributionsdomains hinzugefü gt	Neuer Konsolen-Workflow für die Einrichtung von Standardd istributionsdomains wurde hinzugefügt.	17. Juni 2025
Beispielparameter hinzugefügt	Es wurden Beispiele für die Verwendung von Parameter n mit Domainnamen und Ursprungspfaden in Distribut ionsmandanten hinzugefügt.	17. Juni 2025
Unterstützung für CloudFront Funktionen für SaaS Manager hinzugefügt CloudFront	Hilfsfunktionen und das Feld `endpoint` für das Objekt `context` wurden hinzugefügt.	2. Mai 2025
Aktualisierungen der Standardprotokollierung (v2)	Die Partitionsvariable `{distributionid}` wurde hinzugefügt, um das Senden von Zugriffsprotokollen zu AWS Glue zu unterstützen.	01. Mai 2025
Aktualisierungen der CloudFront verwalteten Richtlinien	ACM-Berechtigungen wurden den verwalteten Richtlinien `CloudFrontReadOnly Access` und `CloudFron tFullAccess` hinzugefügt.	28. April 2025

Unterstützung für Multi-Tenant-Distribution und Distributionsmandanten hinzugefügt	Sie können eine Multi-Tenant-Distribution erstellen, um allgemeine Distributionseinstellungen auf der Grundlage Ihres Ursprungstyps festzulegen. Anschließend können Sie die Multi-Tenant-Distribution wiederverwenden, um mehrere Distributionsmandanten zu erstellen, die diese Einstellungen gemeinsam nutzen. Sie können dann bestimmte Distributionsmandanten anpassen, wenn Sie zusätzliche Websites oder Anwendungen hinzufügen.	28. April 2025
Aktualisierungen für Lambda@Edge-Funktionen	Die Lambda @Edge -Funktionen unterstützen jetzt erweiterte Protokollierungsseuerungen und das Anpassen des CloudWatch Protokoll gruppennamens.	7. April 2025
Statische Anycast-IPs	Sie können statische Anycast-IPs verwenden, um das Routing von Apex-Domains direkt an Ihre CloudFront Distributionen zu aktivieren.	4. April 2025
Zusätzliche Hilfsmethoden für die Änderung des Ursprungs hinzugefügt	Die Hilfsmethoden `selectRequestOriginById()` und `createRequestOriginGroup()` CloudFront Functions wurden hinzugefügt.	2. April 2025

Aktualisierungen der Standardprotokollierung (v2)	Die `{accountid}` -Partitionsvariable und Beispiel-Suffixpfade für die Übermittlung von Zugriffsprotokollen an Amazon S3 wurden hinzugefügt.	14. Februar 2025
Zusätzliche Protokollfelder für den Echtzeitzugriff für die Standardprotokollierung hinzugefügt (v2)	Sie können die Logfelder `c-country` und die `cache-behavior-path-pattern` Echtzeitzugriffs-Log-Felder angeben, wenn Sie die Standardprotokollierung (v2) aktivieren.	31. Januar 2025
Lambda@Edge unterstützt neuere Laufzeitversion	Lambda @Edge unterstützt jetzt Lambda-Funktionen mit der Node.js 22-Laufzeit.	22. November 2024
Unterstützung für die Stabilität, die die Medienqualität berücksichtigt CloudFront	Sie können die Funktion Media Quality-Aware Resiliency (MQAR) verwenden, sodass CloudFront automatisch der Ursprung in einer Ursprungsgruppe mit dem höchsten Medienqualitätsfaktor ausgewählt wird.	21. November 2024
Hilfsmethode für die Änderung des Ursprungs	Eine neue CloudFront Functions-Hilfsmethode zur Änderung des Ursprungs wurde hinzugefügt.	21. November 2024

VPC-Ursprünge	Verwenden Sie CloudFront VPC-Ursprünge, um den Zugriff auf einen Application Load Balancer-, Network Load Balancer- oder EC2-Instance-Ursprung einzuschränken.	20. November 2024
Aktualisierungen für verwaltete Richtlinie	Von `CloudFrontFullAccess` verwaltete Richtlinie aktualisiert.	20. November 2024
Statische Anycast-IPs	Sie können statische Anycast-IP-Adressen CloudFront zur Verwendung mit Ihren Distributionen anfordern.	20. November 2024
Unterstützung für Standardprotokollierung hinzugefügt	CloudFront unterstützt die Standardprotokollierung (v2) und das Senden Ihrer Protokolle an Amazon CloudWatch Logs, Amazon Data Firehose und Amazon Simple Storage Service (Amazon S3).	20. November 2024
Unterstützung für gRPC hinzugefügt	CloudFront unterstützt jetzt gRPC-Anfragen für Ihre Distribution.	20. November 2024
Neue verwaltete Richtlinie für VPC-Ursprünge hinzugefügt	Die neue verwaltete Richtlinie `AWSCloudFrontVPCOriginServiceRolePolicy` wurde hinzugefügt.	20. November 2024
Lambda@Edge unterstützt neuere Laufzeitversion	Lambda @Edge unterstützt jetzt Lambda-Funktionen mit der Python 3.13-Laufzeit.	13. November 2024

Evaluieren Sie anhand von Regeln AWS Config	Evaluieren Sie Ihre CloudFront Konfigurationen mit AWS Config Regeln.	20. September 2024
Weitere Inhalte zur Fehlerbehebung hinzugefügt	Es wurden weitere Inhalte zur Problembehandlung für die HTTP-Fehlerstatuscodes 4xx und 5xx hinzugefügt.	26. August 2024
Neue verwaltete Cache-Richtlinien hinzugefügt	Es wurden die neuen verwalteten Cache-Richtlinien `UseOriginCacheControlHeaders` und `UseOriginCacheControlHeaders-QueryStrings` hinzugefügt.	24. Mai 2024
Unterstützung für Ursprungszugriffssteuerung hinzugefügt	Sie können jetzt eine Origin Access Control (OAC) für AWS Elemental MediaPackage V2 und eine AWS Lambda Funktions-URL erstellen.	11. April 2024
Real-time Zugriffs-Log-Felder für CMCD	18 CMCD-Felder (Common Media Client Data) für Zugriffsprotokolle in Echtzeit hinzugefügt.	9. April 2024
Erste Schritte mit einer CloudFront Standarddistribution	Das Tutorial für eine Standarddistribution, die einen Amazon-S3-Ursprung mit Ursprungszugriffssteuerung (OAC) verwendet, wurde aktualisiert.	18. März 2024

Codebeispiele für die CloudFront Verwendung von AWS SDKs	Es wurden Codebeispiele hinzugefügt, die die Verwendung CloudFront mit einem AWS Software Development Kit (SDK) zeigen. Die Beispiele sind in Codeauszüge unterteilt, die Ihnen zeigen, wie Sie einzelne Servicefunktionen aufrufen können, und Beispiele, die Ihnen zeigen, wie Sie eine bestimmte Aufgabe ausführen können, indem Sie mehrere Funktionen innerhalb desselben Services aufrufen.	16. Februar 2024
AWS Verwaltete Richtlinie	Die `CloudFrontReadOnlyAccess` - und `CloudFrontFullAccess` -IAM-Richtlinien unterstützen jetzt `KeyValueStore` -Vorgänge.	19. Dezember 2023
JavaScript Runtime 2.0	JavaScript Runtime 2.0-Funktionen für CloudFront Funktionen hinzugefügt.	21. November 2023

CloudFront KeyValueStore	Amazon unterstützt CloudFront jetzt CloudFront KeyValueStore. Bei dieser Funktion handelt es sich um einen sicheren, globalen Schlüssel wert-Datenspeicher mit niedriger Latenz, der den Lesezugriff innerhalb von Functions ermöglicht. CloudFront Sie können erweiterte anpassbare Logik an CloudFront Edge-Stan dorten aktivieren.	21. November 2023
Lambda@Edge unterstützt neuere Laufzeitversion	Lambda @Edge unterstützt jetzt Lambda-Funktionen mit der Node.js 20-Laufzeit.	15. November 2023
Sicherheits-Dashboard	CloudFront erstellt ein Sicherheits-Dashboard, wenn Sie eine Distribution erstellen . Aktivieren AWS WAF und verwalten Sie geografische Einschränkungen und lassen Sie sich allgemeine Daten für Anfragen, Bots und Protokolle anzeigen.	8. November 2023
Sortieren von Abfrageze ichenfolgen in Funktionen	CloudFront unterstützt jetzt das Sortieren von Abfrageze ichenfolgen mithilfe von CloudFront Funktionen.	3. Oktober 2023
AWS WAF Sicherheitsempfehl ungen	Amazon zeigt CloudFront jetzt AWS WAF Sicherheitsempfehl ungen auf der CloudFront Konsole an.	26. September 2023

Unterstützung für die Bereitstellung veralteter (abgelaufener) Cache-Inhalte	CloudFront unterstützt die Direktiven `Stale-While-Revalidate` und `Stale-If-Error` Cache Control.	15. Mai 2023
Aktiviere AWS WAF Schutzmaßnahmen mit einer Auswahl	Eine optimierte Methode zum Hinzufügen von AWS WAF Sicherheitsvorkehrungen zu Distributionen. CloudFront	10. Mai 2023
Aktivieren von ACLs für neue S3-Buckets, die für Standardprotokolle verwendet werden	Hinweis und Links für die ACL-Standardeinstellung für neue S3-Buckets hinzugefügt.	11. April 2023
Erstellen eines Ursprungs mit Amazon S3 Object Lambda	Sie können ein Alias eines Zugriffspunkts von Amazon S3 Object Lambda als Ursprung für Ihre Verteilung verwenden.	31. März 2023
Passen Sie den HTTP-Status und den Text mithilfe von Funktionen an CloudFront	Sie können CloudFront Functions verwenden, um den Statuscode der Viewer-Antwort zu aktualisieren und den Antworttext zu ersetzen oder zu entfernen.	29. März 2023
Platzhalteroptionen für CORS-Header für Ports hinzugefügt	In CORS-Zugriffskontroll-Headern können Sie jetzt Platzhalterkonfigurationen für Ports verwenden.	20. März 2023
Neuer Link für das AWS Security Hub CSPM Benutzerhandbuch hinzugefügt	Die Sprache wurde aktualisiert und ein Link zu den neu organisierten CloudFront Amazon-Steuerelementen im AWS Security Hub CSPM Benutzerhandbuch hinzugefügt.	9. März 2023

CloudFront unterstützt jetzt Blocklisten („alle außer“) in den Richtlinien für Originalanfragen	Verwenden Sie Blocklisten in den Richtlinien für Quellanfragen, um alle Abfrageze ichenfolgen, HTTP-Head er oder Cookies, mit Ausnahme der angegebenen, in Anfragen einzubeziehen, die CloudFront an den Ursprung gesendet werden.	22. Februar 2023
CloudFront fügt eine neue Richtlinie für verwaltete Ursprungsanfragen hinzu, um alle Viewer-Header mit Ausnahme des Host-Headers weiterzuleiten	Verwenden Sie die CloudFron t neue Richtlinie für verwaltet e Ursprungsanfragen, um alle Header aus der Viewer-An frage, mit Ausnahme des `Host` Headers, in Anfragen einzubeziehen, die CloudFron t an den Ursprung gesendet werden.	22. Februar 2023
Die Einschränkungen für Lambda@Edge wurden aktualisiert	Lambda@Edge unterstützt Lambda-Laufzeit-Verwaltungs konfigurationen, die auf Auto (Automatisch) festgelegt sind.	16. Februar 2023
Die IAM-Anleitung für wurde aktualisiert CloudFront	Aktualisierung des Leitfadens zur Ausrichtung an bewährten IAM-Methoden. Weitere Informationen finden Sie unter Bewährte Sicherheitsmethode n in IAM.	15. Februar 2023
Verbesserte Sicherheit mit Ursprungszugriffssteuerung	Sie können jetzt MediaStor e Ursprünge sichern, indem Sie nur den Zugriff auf die angegebenen CloudFront Distributionen zulassen.	9. Februar 2023

Neue Header zur Bestimmung der Header-Struktur des Viewers	Sie können jetzt die Header-Reihenfolge und die Anzahl der Header hinzufügen, um den Viewer leichter anhand der gesendeten Header identifizieren zu können.	13. Januar 2023
Lambda@Edge unterstützt neuere Laufzeitversion	Lambda @Edge unterstützt jetzt Lambda-Funktionen mit der 18-Laufzeit. Node.js	12. Januar 2023
Entfernen von Antwort-Headern mithilfe einer Antwort-Header-Richtlinie	Sie können jetzt eine Richtlinie für CloudFront Antwort-Header verwenden, um Header, die in der Antwort CloudFront empfangen wurden, vom Ursprung zu entfernen. Die angegebenen Header sind nicht in der Antwort enthalten, die CloudFront an die Zuschauer gesendet wird.	3. Januar 2023
Kontinuierliche Bereitstellung zum sicheren Testen von Konfigurationsänderungen	Sie können jetzt Änderungen an Ihrer CDN-Konfiguration bereitstellen, indem Sie Tests unter Verwendung eines Teils des Produktionsdatenverkehrs durchführen.	18. November 2022
Veröffentlichung des Headers `CloudFront-Viewer-JA3-Fingerprint`	Sie können jetzt mithilfe des JA3-Fingerprints feststellen, ob die Anforderung von einem bekannten Client stammt.	16. November 2022

Platzhalteroptionen für CORS-Header hinzugefügt	In einigen CORS-Zugriffskontroll-Headern können Sie jetzt verschiedene Platzhalterkonfigurationen verwenden.	11. November 2022
Zusätzliche Metriken für CloudFront Verteilungen	Unterstützung für `MonitoringSubscription` in der CloudFront API und Cloud Formation.	3. Oktober 2022
Verbesserte Sicherheit mit Ursprungszugriffssteuerung	Sie können jetzt Amazon S3-Ursprünge sichern, indem Sie nur den Zugriff auf die angegebenen CloudFront Distributionen zulassen.	24. August 2022
HTTP/3 Unterstützung für Distributionen CloudFront	Sie können jetzt HTTP/3 für Ihre CloudFront Distribution wählen.	15. August 2022
Fügen Sie der Kopfzeile Handshake-Details hinzu CloudFront-Viewer-TLS	Sie können nun Informationen über den verwendeten SSL/TLS Handshake anzeigen.	27. Juni 2022
Neue Metrik in der Kopfzeile Server-Timing	Die neue Metrik `cdn-downstream-fbl` wurde zu den `Server-Timing` -Headern hinzugefügt.	13. Juni 2022
Neuer Header zum Abrufen von Informationen über die TLS-Version und Verschlüsselung	Sie können den `CloudFront-Viewer-TLS` Header jetzt verwenden, um Informationen über die Version von TLS (oder SSL) und die Chiffre abzurufen, die für die Verbindung zwischen dem Viewer und verwendet wurde. CloudFront	23. Mai 2022

Neue FunctionThrottles Metrik für Funktionen CloudFront	Mit Amazon können Sie jetzt überwachen CloudWatch, wie oft eine CloudFront Funktion in einem bestimmten Zeitraum gedrosselt wurde.	4. Mai 2022
CloudFront unterstützt Lambda-Funktions-URLs	Wenn Sie eine serverlos e Webanwendung mithilfe von Lambda-Funktionen mit Funktions-URLs erstellen, können Sie jetzt eine Reihe von CloudFront Vorteilen hinzufügen.	6. April 2022
Server-Timing Header in HTTP-Antworten	Sie können jetzt den `Server-Timing` Header in HTTP-Antworten aktivieren, die von gesendet werden, CloudFron t um Kennzahlen anzuzeige n, die Ihnen helfen können, Einblicke in das Verhalten und die Leistung von zu gewinnen CloudFront.	30. März 2022
Verwenden Sie die AWS verwaltete Präfixliste, um den eingehenden Verkehr zu begrenzen	Sie können jetzt den eingehenden HTTP- und HTTPS-Verkehr zu Ihren Ursprüngen beschränken, und zwar nur für die IP-Adressen, die zu Servern gehören, die mit dem Absender verbunden CloudFront sind.	7. Februar 2022

Neues Feature	CloudFront fügt Unterstützung für Antwortheader-Richtlinien hinzu, mit denen Sie die HTTP-Header angeben können, die zu den HTTP-Antworten CloudFront hinzugefügt werden, die an Betrachter (Webbrowser oder andere Clients) gesendet werden. Sie können die gewünschten Header (und die zugehörigen Werte) angeben, ohne Änderungen am Ursprung vorzunehmen oder Code zu schreiben. Weitere Informationen finden Sie unter Hinzufügen oder Entfernen von HTTP-Headern in Antworten. CloudFront	2. November 2021
Neuer `CloudFront-Viewer-Address` Anforderungsheader	CloudFront fügt Unterstützung für einen neuen Header hinzu`CloudFront-Viewer-Address`, der die IP-Adresse des Viewers enthält, an den die HTTP-Anfrage gesendet wurde CloudFront. Weitere Informationen finden Sie unter Hinzufügen von CloudFront Anforderungsheadern.	25. Oktober 2021

Lambda @Edge unterstützt die neue Runtime-Version	Lambda@Edge unterstützt jetzt Lambda-Funktionen mit der Python-3.9-Laufzeit. Weitere Informationen finden Sie unter Unterstützte Laufzeiten.	22. September 2021
AWS verwaltete Richtlinienaktualisierung	CloudFront hat die CloudFrontReadOnlyAccess Richtlinie aktualisiert. Weitere Informationen finden Sie unter CloudFront Aktualisierungen AWS verwalteter Richtlinien.	8. September 2021
Neues Feature	CloudFront unterstützt jetzt ECDSA-Zertifikate für HTTPS-Verbindungen, die dem Betrachter zugewandt sind. Weitere Informationen finden Sie unter Unterstützte Protokolle und Chiffren zwischen Viewern und CloudFront Anforderungen für die Verwendung von Zertifikaten mit. SSL/TLS CloudFront	14. Juli 2021
Neues Feature	CloudFront unterstützt jetzt mehr Möglichkeiten, einen alternativen Domainnamen von einer Distribution in eine andere zu verschieben, ohne Kontakt aufnehmen zu müssen. Support Weitere Informationen finden Sie unter Verschieben eines alternativen Domainnamens in eine andere Distribution.	7. Juli 2021

Neue Sicherheitsrichtlinie	CloudFront unterstützt jetzt eine neue Sicherheitsrichtlinie, TLSv1.2 _2021, mit einer kleineren Anzahl unterstützter Chiffren. Weitere Informationen finden Sie unter Unterstützte Protokolle und Chiffren zwischen Viewern und. CloudFront	23. Juni 2021
Neues Feature	Amazon unterstützt CloudFront jetzt CloudFront Functions, eine native Funktion CloudFront, mit der Sie einfache Funktionen JavaScript für umfangreiche, latenzempfindliche CDN-Anpassungen einschreiben können. Weitere Informationen finden Sie unter Customizing at the Edge mit Funktionen. CloudFront	3. Mai 2021
Lambda @Edge unterstützt neuere Runtime-Versionen	Lambda @Edge unterstützt jetzt Lambda-Funktionen mit der Node.js 14-Laufzeit. Weitere Informationen finden Sie unter Unterstützte Laufzeiten.	29. April 2021

Entfernen Sie die Dokumentation für RTMP-Distributionen	Amazon CloudFront hat RTMP-Distributionen (Real-Time Messaging Protocol) am 31. Dezember 2020 eingestellt. Die Dokumentation für RTMP-Distributionen wurde jetzt aus dem Amazon CloudFront Developer Guide entfernt.	10. Februar 2021
Neue Preisoption	Amazon CloudFront führt das CloudFront Sicherheitssparpaket ein, mit dem Sie auf einfache Weise bis zu 30% der CloudFront Gebühren auf Ihrer AWS Rechnung sparen können. Weitere Informationen finden Sie unter Häufig gestellte Fragen zum Sicherheitssparpaket.	5. Februar 2021
Neues Tutorial	Der Amazon CloudFront Developer Guide enthält jetzt ein Tutorial zur Verwendung von Amazon CloudFront zur Beschränkung des Zugriffs auf einen Application Load Balancer in Elastic Load Balancing. Weitere Informationen finden Sie unter Beschränken des Zugriffs auf Application Load Balancer.	18. Dezember 2020

Neue Option für die Verwaltung öffentlicher Schlüssel	CloudFront unterstützt jetzt die Verwaltung öffentlicher Schlüssel für signierte URLs und signierte Cookies über die CloudFront Konsole und die API, ohne dass Zugriff auf den AWS-Konto Root-Benutzer erforderlich ist. Weitere Informationen finden Sie unter Festlegen der Aussteller, die signierte URLs und signierte Cookies erstellen können.	22. Oktober 2020
Neues Feature — Origin Shield	CloudFront unterstützt jetzt CloudFront Origin Shield, eine zusätzliche Ebene in der CloudFront Caching-Infrastruktur, die dazu beiträgt, die Auslastung deines Origins zu minimieren, seine Verfügbarkeit zu verbessern und die Betriebskosten zu senken. Weitere Informationen finden Sie unter Amazon CloudFront Origin Shield https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/origin-shield.html verwenden.	20. Oktober 2020

Neues Komprimierungsformat	CloudFront unterstützt jetzt die Brotli-Komprimierungsformat ion, wenn Sie so konfigurieren CloudFront , dass Objekte an CloudFront Kantenpositionen komprimiert werden. Sie können auch so konfigurieren CloudFront , dass Brotli-Ob jekte mithilfe eines normalisi erten Headers zwischeng espeichert werden. `Accept-Encoding` Weitere Informati onen finden Sie unter Bereitste llen komprimierter Dateien und Unterstützte Komprimierung.	14. September 2020
Neues TLS-Protokoll	CloudFront unterstützt jetzt das TLS 1.3-Protokoll für HTTPS-Verbindungen zwischen Viewern und CloudFront Distributionen. TLS 1.3 ist standardmäßig in allen CloudFront Sicherhei tsrichtlinien aktiviert. Weitere Informationen finden Sie unter Unterstützte Protokolle und Chiffren zwischen Viewern und. CloudFront	3. September 2020

Neue Zugriffsprotokolle in Echtzeit	CloudFront unterstützt jetzt konfigurierbare Zugriffsprotokolle in Echtzeit. Mithilfe von Zugriffsprotokollen in Echtzeit können Sie Informationen über Anfragen abrufen, die an eine Distribution gestellt wurden, in Echtzeit. Sie können Zugriffsprotokolle in Echtzeit verwenden, um die Leistung der Inhaltsbereitstellung zu überwachen, zu analysieren und Maßnahmen zu ergreifen. Weitere Informationen finden Sie unter Real-time Protokolle.	31. August 2020
API-Unterstützung für zusätzliche Metriken	CloudFront unterstützt jetzt die Aktivierung von acht zusätzlichen Echtzeit-Metriken mit der CloudFront API. Weitere Informationen finden Sie unter Zusätzliche Metriken aktivieren.	28. August 2020
Neue CloudFront HTTP-Header	CloudFront zusätzliche HTTP-Header hinzugefügt, um Informationen über den Betrachter wie Gerätetyp, geografischer Standort und mehr zu ermitteln. Weitere Informationen finden Sie unter Hinzufügen von CloudFront Anforderungsheadern.	23. Juli 2020

Neues Feature	CloudFront unterstützt jetzt Cache-Richtlinien und Origin-Request-Richtlinien, die Ihnen eine detailliertere Kontrolle über den Cache-Schlüssel und die Quellanfragen für Ihre Distributionen geben. CloudFront Weitere Informationen finden Sie unter Steuern des Cache-Schlüssels und Steuern von Ursprungsanforderungen.	22. Juli 2020
Neue Sicherheitsrichtlinie	CloudFront unterstützt jetzt die neue Sicherheitsrichtlinie TLSv1.2 _2019 mit einer kleineren Anzahl unterstützter Chiffren. Weitere Informationen finden Sie unter Unterstützte Protokolle und Chiffren zwischen Viewern und. CloudFront	8. Juli 2020
Neue Einstellungen zur Steuerung von Timeouts und Startversuchen	CloudFront neue Einstellungen hinzugefügt, die Timeouts und Versuche bei der Herkunft steuern. Weitere Informationen finden Sie unter Steuern von Timeouts und Verbindungsversuchen für Ursprünge.	5. Juni 2020

Neue Dokumentation für den Einstieg in CloudFront die Erstellung einer sicheren statischen Website	Erstellen Sie zunächst eine sichere statische Website mit CloudFront Amazon S3 CloudFront, Lambda @Edge und mehr, die alle mit CloudFormation bereitgestellt werden. Weitere Informationen finden Sie unter Erste Schritte mit einer sicheren statischen Website.	2. Juni 2020
Lambda @Edge unterstützt neuere Runtime-Versionen	Lambda @Edge unterstützt jetzt Lambda-Funktionen mit den Laufzeiten Node.js 12 und Python 3.8. Weitere Informationen finden Sie unter Unterstützte Laufzeiten.	27. Februar 2020
Neue Echtzeit-Metriken in CloudWatch	Amazon CloudFrontnow bietet acht zusätzliche Echtzeit-Metriken in Amazon an CloudWatch. Weitere Informationen finden Sie unter Zusätzliche CloudFront Vertriebsmetriken aktivieren.	19. Dezember 2019
Neue Felder in den Zugriffsprotokollen	CloudFront fügt den Zugriffsprotokollen sieben neue Felder hinzu. Weitere Informationen finden Sie unter Standard-Protokolldateifelder.	12. Dezember 2019

AWS WordPress Plugin	Sie können das AWS WordPress Plugin verwenden , um den Besuchern Ihrer WordPress Website ein beschleunigtes Seherlebnis zu bieten CloudFront. (Update: Seit dem 30. September 2022 ist das AWS WordPress For-Plugin veraltet.)	30. Oktober 2019
Tag-based und Richtlinien für IAM-Berechtigungen auf Ressourcenebene	CloudFront unterstützt jetzt zwei zusätzliche Methoden zur Angabe von IAM-Berec htigungsrichtlinien: Tag-basie rte und Richtlinienberecht igungen auf Ressource nebene. Weitere Informationen finden Sie unter Verwalten des Zugriffs auf Ressourcen.	8. August 2019
Unterstützung für die Programmiersprache Python	Zusätzlich zu können Sie jetzt die Programmiersprache Python verwenden, um Funktionen in Lambda @Edge zu Node.js entwickel n. Beispielfunktionen, die eine Vielzahl von Szenarien abdecken, finden Sie unter Lambda@Edge-Beispielfunktio nen.	1. August 2019

Die Überwachungsdiagramme wurden aktualisiert	Inhaltsaktualisierungen, um neue Möglichkeiten zu beschreiben, wie Sie Lambda-Funktionen, die Ihren CloudFront Distributionen zugeordnet sind, direkt von der CloudFront Konsole aus überwachen können, um Fehler leichter verfolgen und debuggen zu können. Weitere Informationen finden Sie unter -Überwachung CloudFront.	20. Juni 2019
Konsolidierter Sicherheitsinhalt	In einem neuen Kapitel zur Sicherheit werden Informationen zu CloudFront Funktionen und zur Implementierung von Datenschutz, IAM, Protokollierung, Compliance und mehr zusammengefasst. Weitere Informationen finden Sie unter Sicherheit.	24. Mai 2019
Die Domänenvalidierung ist jetzt erforderlich	CloudFront erfordert jetzt, dass Sie ein SSL-Zertifikat verwenden, um zu überprüfen, ob Sie die Erlaubnis haben, einen alternativen Domainnamen mit einer Distribution zu verwenden. Weitere Informationen finden Sie unter Verwenden alternativer Domainnamen in Verbindung mit HTTPS.	9. April 2019

Der PDF-Dateiname wurde aktualisiert	Der neue Dateiname für den Amazon CloudFront Developer Guide lautet: AmazonClo udFront _DevGuide. Der vorherige Name lautete: cf-dg	7. Januar 2019
Neue Features	CloudFront unterstützt jetzt ein TCP-based Protokoll WebSocket, das nützlich ist, wenn Sie langlebige Verbindungen zwischen Clients und Servern benötigen . Sie können jetzt auch Origin-Failover für Szenarien einrichten CloudFront , die eine hohe Verfügbarkeit erfordern. Weitere Informati onen finden Sie unter Verwenden WebSocket mit CloudFront Distributionen und Optimieren der Hochverfü gbarkeit mit CloudFront Origin-Failover.	20. November 2018

Neues Feature	CloudFront unterstützt jetzt eine detaillierte Fehlerprotokollierung für HTTP-Anfragen, die Lambda-Funktionen ausführen. Sie können die Protokolle speichern CloudWatch und sie zur Behebung von HTTP 5xx-Fehlern verwenden, wenn Ihre Funktion eine ungültige Antwort zurückgibt. Weitere Informationen finden Sie unter CloudWatch Metriken und CloudWatch Protokolle für Lambda-Funktionen.	8. Oktober 2018
Neues Feature	Sie können jetzt auswählen, ob Lambda@Edge den Body in einer Anforderung für eine nicht schreibgeschützte HTTP-Methode (POST, PUT, DELETE etc.) weitergeben soll, sodass Sie in Ihrer Lambda-Funktion darauf zugreifen können. Sie können den Lesezugriff wählen oder angeben, dass Sie den Textkörper ersetzen möchten. Weitere Informationen finden Sie unter Zugriff auf den Anforderungstext über die Option „Include Body“ (Text einschließen).	14. August 2018

Neues Feature	CloudFront unterstützt jetzt zusätzlich zu oder anstelle von gzip die Bereitstellung von Inhalten, die mithilfe von Brotli oder anderen Komprimierungsalgorithmen komprimiert wurden. Weitere Informationen finden Sie unter Bereitstellung komprimierter Dateien.	25. Juli 2018
Reorganisation	Der Amazon CloudFront Developer Guide wurde neu organisiert, um das Auffinden verwandter Inhalte zu vereinfachen und die Scanbarkeit und Navigation zu verbessern.	28. Juni 2018
Neue Funktion	Mit Lambda@Edge können Sie jetzt die Bereitstellung von Inhalten, die in einem Amazon-S3-Bucket gespeichert sind, weiter anpassen. Sie haben nun Zugriff auf zusätzliche Header, einschließlich benutzerdefinierter Header innerhalb von Ereignissen auf der Ursprungsseite. Weitere Informationen zur Personalisierung von Inhalten basierend auf Standort und Gerätetyp des Betrachters finden Sie in diesen Beispielen.	20. März 2018

Neue Funktion	Sie können Amazon jetzt verwenden, um HTTPS-Verbindungen CloudFront zu Startpunkten mithilfe des Elliptic Curve Digital Signature Algorithm (ECDSA) auszuhandeln. ECDSA verwendet kleinere Schlüssel, die schneller, aber genauso sicher wie der ältere RSA-Algorithmus sind. Weitere Informationen finden Sie unter Unterstützte SSL/TLS Protokolle und Chiffren für die Kommunikation zwischen und Ihrem Ursprung und Über RSA CloudFront - und ECDSA-Chiffren.	15. März 2018
Neue Funktion	Lambda @Edge ermöglicht es Ihnen, die von Ihrem Ursprung ausgehenden Fehlerantworten individuell anzupassen, indem Sie Lambda-Funktionen als Reaktion auf HTTP-Fehler ausführen können, die Amazon von Ihrem Ursprung hat. CloudFrontreceives Weitere Informationen entnehmen Sie den Beispielen für Weiterleitungen an einen anderen Standort und die Generierung einer Antwort mit dem Statuscode 200 (OK).	21. Dezember 2017

Neue Funktion	Eine neue CloudFront Funktion, die Verschlüsselung auf Feldebene, hilft Ihnen dabei, die Sicherheit vertrauli cher Daten wie Kreditkar tennummern oder persönlich identifizierbarer Informationen (PII) wie Sozialversicherung snummern weiter zu erhöhen. Weitere Informationen finden Sie unter Schutz sensibler Daten durch Verschlüsselung auf Feldebene.	14. Dezember 2017
Verlauf der Dokumente archiviert	Der Verlauf älterer Dokumente wurde archiviert.	1. Dezember 2017

Die vorliegende Übersetzung wurde maschinell erstellt. Im Falle eines Konflikts oder eines Widerspruchs zwischen dieser übersetzten Fassung und der englischen Fassung (einschließlich infolge von Verzögerungen bei der Übersetzung) ist die englische Fassung maßgeblich.