Panduan Pengguna

AWS CodeBuild



Versi API 2016-10-06

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

AWS CodeBuild: Panduan Pengguna

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

Merek dagang dan tampilan dagang Amazon tidak boleh digunakan sehubungan dengan produk atau layanan apa pun yang bukan milik Amazon, dengan cara apa pun yang dapat menyebabkan kebingungan di antara pelanggan, atau dengan cara apa pun yang menghina atau mendiskreditkan Amazon. Semua merek dagang lain yang tidak dimiliki oleh Amazon merupakan hak milik masing-masing pemiliknya, yang mungkin atau mungkin tidak terafiliasi, terkait dengan, atau disponsori oleh Amazon.

Table of Contents

Apa itu AWS CodeBuild? 1
Mendaftar untuk Akun AWS 1
Bagaimana cara menjalankan CodeBuild 1
Harga untuk CodeBuild 3
Bagaimana saya memulai CodeBuild? 3
Konsep 3
Bagaimana cara CodeBuild kerja 3
Langkah selanjutnya 5
Memulai 6
Memulai menggunakan konsol 6
Langkah 1: Buat kode sumber 7
Langkah 2: Buat file buildspec 10
Langkah 3: Buat dua bucket S3 12
Langkah 4: Unggah kode sumber dan file buildspec 13
Langkah 5: Buat proyek pembangunan 14
Langkah 6: Jalankan build 16
Langkah 7: Lihat informasi build yang diringkas 17
Langkah 8: Lihat informasi build terperinci 18
Langkah 9: Dapatkan artefak keluaran build 19
Langkah 10: Hapus bucket S3 19
Membungkus 20
Memulai menggunakan AWS CLI 21
Langkah 1: Buat kode sumber 22
Langkah 2: Buat file buildspec 24
Langkah 3: Buat dua bucket S3 27
Langkah 4: Unggah kode sumber dan file buildspec 27
Langkah 5: Buat proyek pembangunan 29
Langkah 6: Jalankan build 33
Langkah 7: Lihat informasi build yang diringkas 34
Langkah 8: Lihat informasi build terperinci 37
Langkah 9: Dapatkan artefak keluaran build 40
Langkah 10: Hapus bucket S3 40
Membungkus 41
Gunakan sampel berbasis kasus 42

Cross-service sampel 43
Sampel Amazon ECR 44
Sampel Amazon EFS 50
AWS CodePipeline sampel 57
AWS Config sampel 68
Buat contoh notifikasi 69
Bangun contoh lencana 85
Buat proyek build dengan lencana build 85
Akses AWS CodeBuild membangun lencana 88
Publikasikan CodeBuild lencana build 89
CodeBuild status lencana 89
Contoh laporan pengujian 89
Jalankan sampel laporan pengujian 90
Sampel Docker untuk CodeBuild 96
Docker dalam sampel gambar khusus 97
Contoh server pembuatan gambar Docker 100
Windows Docker membangun sampel 103
Contoh 'Publikasikan gambar Docker ke Amazon ECR' 105
Registri pribadi dengan AWS Secrets Manager sampel 114
Host build output dalam bucket S3 118
Beberapa sampel input dan output 121
Buat proyek build dengan beberapa input dan output 122
Membuat proyek tanpa sumber 125
Versi runtime dalam contoh file buildspec 125
Perbarui versi runtime dalam file buildspec 126
Tentukan dua runtime 130
Contoh versi sumber 134
Tentukan versi GitHub repositori dengan ID komit 136
Tentukan versi GitHub repositori dengan referensi dan ID komit 137
Third-party sampel repositori sumber 138
Jalankan sampel Bitbucket 139
Jalankan sampel Server GitHub Perusahaan 144
Jalankan permintaan GitHub tarik dan sampel filter webhook 150
Tutorial: Penandatanganan kode Apple dengan Fastlane dalam CodeBuild menggunakan S3 untuk penyimpanan sertifikat 154

Tutorial: Penandatanganan kode Apple dengan Fastlane di CodeBuild Menggunakan GitHub untuk penyimpanan sertifikat 161
Tetapkan nama artefak pada waktu pembuatan 167
Jalankan sampel Windows 170
Jalankan sampel Windows 170
Struktur direktori 171
F # dan .NET Framework 171
Visual Basic dan .NET Framework 172
Berkas 172
F # dan .NET Framework 172
Visual Basic dan .NET Framework 177
Merencanakan build 191
Referensi Buildspec 194
Nama file buildspec dan lokasi penyimpanan 194
Sintaks Buildspec 195
versi 198
berjalan seperti 198
env 199
proxy 204
fase 204
melaporkan 209
artefak 211
cache 217
Contoh Buildspec 219
Versi Buildspec 222
Referensi buildspec batch 222
batch 223
`batch/build-grafik` 223
`batch/build-daftar` 226
`batch/build-matriks` 228
`batch/build-fanout` 230
Membangun referensi lingkungan 233
Gambar Docker disediakan oleh CodeBuild 234
Dapatkan daftar gambar Docker saat ini 234
Gambar komputasi EC2 235
Gambar komputasi Lambda 237

Gambar yang tidak digunakan lagi CodeBuild 243
Runtime yang tersedia 244
Versi waktu aktif 271
Membangun mode dan tipe komputasi lingkungan 275
Tentang komputasi 276
Tentang jenis lingkungan kapasitas yang dicadangkan 276
Tentang jenis lingkungan sesuai permintaan 340
Shell dan perintah di lingkungan build 352
Variabel lingkungan di lingkungan build 353
Tugas latar belakang di lingkungan build 359
Proyek build 360
Membuat proyek build 360
Prasyarat 361
Buat proyek build (konsol) 361
Buat proyek build (AWS CLI) 382
Buat proyek build (AWS SDK) 402
Buat proyek build (CloudFormation) 403
Membuat aturan notifikasi 403
Ubah pengaturan proyek build 406
Mengubah setelan proyek build (konsol) 406
Mengubah setelan proyek build (AWS CLI) 429
Mengubah setelan proyek build (AWS SDK) 430
Beberapa token akses 430
Langkah 1: Buat rahasia Secrets Manager atau AWS CodeConnections koneksi 431
Langkah 2: Berikan akses peran IAM CodeBuild proyek ke rahasia Secrets Manager 432
Langkah 3: Konfigurasikan Secrets Manager atau AWS CodeConnections token 434
Opsi penyiapan tambahan 438
Hapus proyek pembangunan 441
Hapus proyek build (konsol) 441
Menghapus proyek build (AWS CLI) 442
Menghapus proyek build (AWS SDK) 442
Dapatkan URL proyek build publik 443
Bagikan proyek pembangunan 444
Bagikan proyek 444
Layanan terkait 447
Akses proyek bersama 448

Batalkan berbagi proyek bersama 448
Mengidentifikasi proyek bersama 449
Izin proyek bersama 449
Tag membangun proyek 449
Menambahkan tag ke proyek 450
Lihat tag untuk proyek 452
Mengedit tag untuk proyek 453
Menghapus tag dari proyek 454
Gunakan pelari 455
GitHub Tindakan 455
GitLab pelari 480
Pelari buildkite 495
Gunakan webhook 517
Praktik terbaik untuk menggunakan webhook 518
Acara webhook Bitbucket 519
GitHub webhook global dan organisasi 532
GitHub webhook manual 539
GitHub acara webhook 540
GitLab kelompok webhooks 557
GitLab webhook manual 562
GitLab acara webhook 563
Buildkite webhook manual 578
Persetujuan komentar permintaan tarik 579
Lihat detail proyek build 585
Melihat detail proyek build (konsol) 586
Lihat detail proyek build (AWS CLI) 586
Melihat detail proyek build (AWS SDK) 588
Lihat nama proyek build 589
Melihat daftar nama proyek build (konsol) 589
Lihat daftar nama proyek build (AWS CLI) 589
Melihat daftar nama proyek build (AWS SDK) 591
Bangunan 592
Jalankan build secara manual 593
Jalankan build secara lokal 594
Jalankan build (konsol) 597
Jalankan build (AWS CLI) 598

Jalankan build batch (AWS CLI) 605
Mulai menjalankan build secara otomatis ()AWS CLI 606
Berhenti menjalankan build secara otomatis ()AWS CLI 607
Jalankan build (AWS SDK) 608
Jalankan build pada komputasi Lambda 608
Alat dan runtime mana yang akan disertakan dalam gambar docker lingkungan runtime yang dikuratori yang dijalankan AWS Lambda? 609
Bagaimana jika gambar yang dikuratori tidak menyertakan alat yang saya butuhkan? 609
Daerah mana yang mendukung AWS Lambda menghitung di? CodeBuild 610
Pembatasan AWS Lambda komputasi 610
Menyebarkan fungsi Lambda menggunakan AWS SAM dengan CodeBuild Lambda Java ... 611
Buat aplikasi React satu halaman dengan CodeBuild Lambda Node.js 615
Perbarui konfigurasi fungsi Lambda dengan CodeBuild Lambda Python 618
Jalankan build pada armada kapasitas yang dicadangkan 622
Buat armada kapasitas yang dicadangkan 623
Praktik terbaik 625
Dapatkah saya berbagi armada kapasitas yang dipesan di beberapa CodeBuild proyek? 626
Bagaimana cara kerja komputasi berbasis atribut? 626
Dapatkah saya menentukan instans Amazon EC2 secara manual untuk armada saya? 627
Wilayah mana yang mendukung armada kapasitas cadangan? 627
Bagaimana cara mengonfigurasi armada macOS kapasitas cadangan? 627
Bagaimana cara mengonfigurasi Image Mesin Amazon (AMI) khusus untuk armada kapasitas cadangan? 628
Keterbatasan armada kapasitas cadangan 630
Properti armada kapasitas cadangan 630
Sampel kapasitas cadangan 635
Jalankan build batch 637
Peran keamanan 637
Jenis build batch 637
Mode laporan batch 642
Informasi selengkapnya 642
Jalankan tes paralel 642
Dukungan di AWS CodeBuild 644
Aktifkan eksekusi pengujian paralel dalam build batch 646
Gunakan perintah `codebuild-` tests-run CLI 647
Gunakan perintah `codebuild-` glob-search CLI 650

Tentang pemisahan tes 652
Gabungkan laporan build individual secara otomatis 653
Sampel eksekusi uji paralel 655
Membangun cache 665
Penyimpanan cache Amazon S3 666
Caching lokal 672
Tentukan cache lokal 674
Membangun debug 676
Debug dibangun dengan kotak pasir CodeBuild 676
Debug dibangun dengan Session Manager 677
Debug build dengan kotak pasir CodeBuild 677
Debug dibangun dengan Session Manager 707
Hapus build 712
Hapus build ()AWS CLI 712
Hapus build (AWS SDK) 713
Coba ulang pembuatan secara manual 713
Coba ulang build secara manual (konsol) 713
Coba ulang build secara manual (AWS CLI) 714
Coba ulang build secara manual (AWS SDK) 715
Coba lagi membangun secara otomatis 715
Coba lagi build secara otomatis (konsol) 715
Coba lagi build secara otomatis ()AWS CLI 716
Coba ulang build (AWS SDK) secara otomatis 716
Hentikan pembangunan 716
Menghentikan build (konsol) 717
Hentikan membangun (AWS CLI) 717
Hentikan membangun (AWS SDK) 718
Hentikan pembuatan batch 718
Menghentikan pembuatan batch (konsol) 718
Hentikan pembuatan batch (AWS CLI) 719
Hentikan pembuatan batch (AWS SDK) 719
Pemicu dibangun secara otomatis 719
Buat pemicu build 720
Edit pemicu build 723
Lihat detail build 726
Lihat detail build (konsol) 726

Lihat detail build (AWS CLI) 727
Lihat detail build (AWS SDK) 727
Bangun transisi fase 727
Lihat ID build 728
Melihat daftar ID build (konsol) 728
Melihat daftar ID build (AWS CLI) 729
Melihat daftar ID build batch (AWS CLI) 730
Melihat daftar ID build (AWS SDK) 731
Melihat ID build untuk proyek build 732
Melihat daftar ID build untuk proyek build (konsol) 732
Melihat daftar ID build untuk proyek build (AWS CLI) 732
Lihat daftar ID build batch untuk proyek build (AWS CLI) 734
Melihat daftar ID build untuk proyek build (AWS SDK) 735
Laporan pengujian 736
Buat laporan pengujian 737
Buat laporan cakupan kode 738
Buat laporan cakupan kode 739
Temukan laporan secara otomatis 740
Konfigurasikan pencarian otomatis laporan menggunakan konsol 741
Konfigurasikan deteksi otomatis laporan menggunakan variabel lingkungan proyek 742
Grup laporan 742
Membuat grup laporan 743
Penamaan grup laporan 749
Bagikan grup laporan 750
Tentukan file pengujian 756
Tentukan perintah pengujian 756
Menandai grup laporan 757
Memperbarui grup laporan 763
Kerangka uji 766
Siapkan Jasmine 766
Mengatur Jest 769
Siapkan pytest 770
Mengatur RSpec 771
Lihat laporan pengujian 772
Melihat laporan pengujian untuk build 772
Melihat laporan pengujian untuk grup laporan 773

Lihat laporan pengujian di AWS akun ... 773
Izin laporan uji ... 773
Peran IAM untuk laporan pengujian ... 774
Izin untuk operasi pelaporan pengujian ... 776
Contoh izin pelaporan uji ... 776
Status laporan pengujian ... 777
Dukungan VPC ... 778
Kasus penggunaan ... 778
Praktik terbaik untuk VPC ... 779
Keterbatasan VPC ... 780
Izinkan akses Amazon VPC di proyek Anda CodeBuild ... 780
Memecahkan masalah pengaturan VPC Anda ... 781
Gunakan titik akhir VPC ... 782
Sebelum Anda membuat titik akhir VPC ... 782
Buat titik akhir VPC untuk CodeBuild ... 783
Membuat kebijakan titik akhir VPC untuk CodeBuild ... 783
Menggunakan server proxy CodeBuild terkelola ... 784
Konfigurasikan konfigurasi proxy terkelola untuk armada kapasitas cadangan ... 785
Jalankan armada kapasitas CodeBuild cadangan ... 786
Gunakan server proxy ... 786
Menyiapkan komponen yang diperlukan untuk berjalan CodeBuild di server proxy ... 787
Jalan CodeBuild kan di server proxy eksplisit ... 790
J CodeBuild alankan di server proxy transparan ... 794
Jalankan manajer paket dan alat lainnya di server proxy ... 796
CloudFormation Template VPC ... 798
Pencatatan log dan pemantauan ... 805
Log panggilan CodeBuild API ... 805
Tentang AWS CodeBuild informasi di CloudTrail ... 805
Tentang AWS CodeBuild entri file log ... 806
Memantau pembangunan ... 809
CloudWatch metrik ... 809
CloudWatch metrik pemanfaatan sumber daya ... 812
CloudWatch dimensi ... 814
CloudWatch alarm ... 814
Lihat CodeBuild metrik ... 815
Lihat metrik pemanfaatan CodeBuild sumber daya ... 817

Buat CodeBuild alarm di CloudWatch 821
Keamanan 822
Perlindungan Data 823
Enkripsi data 825
Manajemen kunci 826
Privasi lalu lintas 826
Manajemen Identitas dan Akses 826
Ikhtisar mengenai pengelolaan akses 826
Menggunakan kebijakan berbasis identitas 831
AWS CodeBuild referensi izin 860
Menggunakan tag untuk mengontrol akses AWS CodeBuild sumber daya 867
Menampilkan sumber daya di konsol 872
Validasi Kepatuhan 872
Ketahanan 873
Keamanan Infrastruktur 873
Akses penyedia sumber 874
Buat dan simpan token dalam rahasia Secrets Manager 874
GitHub dan akses GitHub Enterprise Server 877
Akses Bitbucket 889
GitLab akses 900
Cross-service pencegahan wakil bingung 907
Topik lanjutan 909
Izinkan pengguna berinteraksi dengan CodeBuild 909
Memungkinkan CodeBuild untuk berinteraksi dengan orang lain AWS layanan 916
Enkripsi keluaran build 925
Berinteraksi dengan CodeBuild menggunakan AWS CLI 927
Referensi baris perintah 928
AWS Referensi SDK dan alat 929
Didukung AWS SDK dan alat untuk AWS CodeBuild 929
Bekerja dengan AWS SDK 930
Tentukan titik CodeBuild akhir 931
Tentukan AWS CodeBuild titik akhir (AWS CLI) 932
Tentukan AWS CodeBuild titik akhir (AWS SDK) 932
Gunakan CodeBuild dengan CodePipeline 934
Prasyarat 936
Buat pipeline (konsol) 937

Buat pipeline (AWS CLI) 942
Menambahkan tindakan build 946
Menambahkan tindakan pengujian 950
Gunakan CodeBuild dengan Codecov 954
Integrasikan Codecov ke dalam proyek build 954
Gunakan CodeBuild dengan Jenkins 957
Siapkan Jenkins 957
Pasang plugin 958
Gunakan plugin 958
Gunakan CodeBuild dengan aplikasi tanpa server 960
Sumber daya terkait 961
Pemberitahuan pihak ketiga 961
1) gambar dasar Docker — windowsservercore 962
2) gambar Docker berbasis windows-choco 963
3) gambar Docker berbasis windows-git - versi 2.16.2 963
4) gambar Docker berbasis windows-microsoft-build-tools - versi 15.0.26320.2 963
5) gambar Docker berbasis windows-nuget.commandline -- versi 4.5.1 968
7) gambar Docker berbasis windows-netfx-4.6.2-devpack 968
8) gambar Docker berbasis windows-visualfsharptools, v 4.0 970
9) gambar Docker berbasis windows-netfx-pcl-rakitan referensi-4.6 970
10) gambar Docker berbasis windows-visualcppbuildtools v 14.0.25420.1 974
11) gambar Docker berbasis windows-microsoft-windows-netfx3-ondemand-package.cab .. 978
12) gambar Docker berbasis windows-dotnet-sdk 980
Gunakan tombol CodeBuild kondisi sebagai variabel peran layanan IAM 980
AWS CodeBuild kunci kondisi 981
Menerapkan pengaturan konektivitas VPC pada proyek dan armada Anda 982
Mencegah modifikasi tidak sah pada spesifikasi build proyek 982
Batasi jenis komputasi untuk build Anda 983
Kontrol pengaturan variabel lingkungan 984
Gunakan variabel dalam nama kunci kondisi 985
Periksa keberadaan atribut dalam permintaan API 986
Contoh kode 987
Hal-hal mendasar 987
Tindakan 988
API build publik 1005
Tindakan membangun publik 1005

DescribeBuildBatchesForPublicProject 1006
DescribeBuildsForPublicProject 1009
GetCloudWatchLogsForPublicBuild 1014
GetPresignedUrlsForPublicBuild 1017
GetPublicBuild 1020
GetPublicBuildBatch 1024
GetPublicProject 1029
Tipe data build publik 1032
BuildBatchForDescribeBuildBatchesPublic 1034
PublicBuild 1036
PublicBuildArtifacts 1040
PublicBuildBatch 1041
PublicBuildGroup 1045
PublicBuildSummary 1047
PublicLogsStatus 1049
PublicProject 1050
PublicProjectArtifacts 1053
PublicProjectBuildBatchConfig 1054
PublicProjectEnvironment 1055
PublicProjectSource 1057
PublicWebhook 1059
S3Dapat Diunduh 1061
Pemecahan Masalah 1063
Apache Maven membangun artefak referensi dari repositori yang salah 1064
Perintah build dijalankan sebagai root secara default 1066
Build mungkin gagal ketika nama file tidak memiliki. U.S Karakter bahasa Inggris 1066
Build mungkin gagal saat mendapatkan parameter dari Amazon EC2 Parameter Store 1067
Tidak dapat mengakses filter cabang di CodeBuild konsol 1068
Tidak dapat melihat keberhasilan atau kegagalan bangunan 1068
Status build tidak dilaporkan ke penyedia sumber 1069
Tidak dapat menemukan dan memilih gambar dasar platform Windows Server Core 2019 1069
Perintah sebelumnya dalam file buildspec tidak dikenali oleh perintah selanjutnya 1069
Kesalahan: “Akses ditolak” saat mencoba mengunduh cache 1070
Kesalahan: “BUILD_CONTAINER_UNABLE_TO_PULL_IMAGE” saat menggunakan gambar build khusus 1071

Kesalahan: “Build container ditemukan mati sebelum menyelesaikan build. build container mati karena kehabisan memori, atau image Docker tidak didukung. ErrorCode: 500” 1072
Kesalahan: “Tidak dapat terhubung ke daemon Docker” saat menjalankan build 1072
Kesalahan: "CodeBuild tidak diizinkan untuk melakukan: sts:AssumeRole" saat membuat atau memperbarui proyek build 1073
Kesalahan: “Kesalahan memanggil GetBucketAcl: Pemilik bucket telah berubah atau peran layanan tidak lagi memiliki izin untuk memanggil s3:GetBucketAcl” 1074
Kesalahan: “Gagal mengunggah artefak: Arn tidak valid” saat menjalankan build 1075
Kesalahan: “Gagal klon Git: Tidak dapat mengakses `'URL repositori-Anda'`: Masalah sertifikat SSL: Sertifikat yang ditandatangani sendiri” 1075
Kesalahan: “Bucket yang Anda coba akses harus ditangani menggunakan titik akhir yang ditentukan” saat menjalankan build 1076
Kesalahan: “Gambar build ini memerlukan pemilihan setidaknya satu versi runtime.” 1076
Kesalahan: “QUEUED: INSUFFICIENT_SUBNET” saat build dalam antrian build gagal 1077
Kesalahan: “Tidak dapat mengunduh cache: RequestError: Permintaan kirim gagal disebabkan oleh: x509: Gagal memuat akar sistem dan tidak ada akar yang disediakan” 1078
Kesalahan: “Tidak dapat mengunduh sertifikat dari S3. AccessDenied” 1078
Kesalahan: “Tidak dapat menemukan kredensil” 1078
RequestError kesalahan batas waktu saat berjalan CodeBuild di server proxy 1080
Cangkang bourne (sh) harus ada dalam gambar build 1081
Peringatan: “Melewatkan pemasangan runtime. pemilihan versi runtime tidak didukung oleh image build ini” saat menjalankan build 1082
Kesalahan: “Tidak dapat memverifikasi JobWorker identitas” 1082
Build gagal untuk memulai 1082
Mengakses GitHub metadata dalam build yang di-cache secara lokal 1083
AccessDenied: Pemilik bucket untuk grup laporan tidak cocok dengan pemilik bucket S3... ... 1083
Kesalahan: “Kredensyal Anda tidak memiliki satu atau lebih cakupan hak istimewa yang diperlukan” saat membuat proyek dengan CodeBuild AWS CodeConnections 1084
Kesalahan: “Maaf, tidak ada terminal sama sekali yang diminta - tidak bisa mendapatkan input” saat membangun dengan perintah instal Ubuntu 1085
Bitbucket build gagal dengan kesalahan otentikasi setelah menggunakan Secrets Manager OAuth 1086
Kuota 1087
Kuota layanan 1087
Batas lainnya 1094
Proyek build 1094

Bangunan 1094
Hitung armada 1095
Laporan 1096
Tanda 1097
Riwayat dokumen 1098
Pembaruan lebih awal 1124
..... mcxxxvii

Apa itu AWS CodeBuild?

AWS CodeBuild adalah layanan build yang dikelola sepenuhnya di cloud. CodeBuild mengkompilasi kode sumber Anda, menjalankan pengujian unit, dan menghasilkan artefak yang siap digunakan. CodeBuild menghilangkan kebutuhan untuk menyediakan, mengelola, dan menskalakan server build Anda sendiri. Ini menyediakan lingkungan bangunan yang dikemas untuk bahasa pemrograman populer dan membangun alat seperti Apache Maven, Gradle, dan banyak lagi. Anda juga dapat menyesuaikan lingkungan build CodeBuild untuk menggunakan alat build Anda sendiri. CodeBuild skala secara otomatis untuk memenuhi permintaan build puncak.

CodeBuild memberikan manfaat ini:

- Dikelola sepenuhnya — CodeBuild menghilangkan kebutuhan untuk mengatur, menambal, memperbarui, dan mengelola server build Anda sendiri.
- Sesuai permintaan — CodeBuild skala sesuai permintaan untuk memenuhi kebutuhan bangunan Anda. Anda hanya membayar untuk jumlah menit build yang Anda konsumsi.
- Out of the box - CodeBuild menyediakan lingkungan build yang telah dikonfigurasi untuk bahasa pemrograman paling populer. Yang perlu Anda lakukan adalah mengarahkan ke skrip build Anda untuk memulai build pertama Anda.

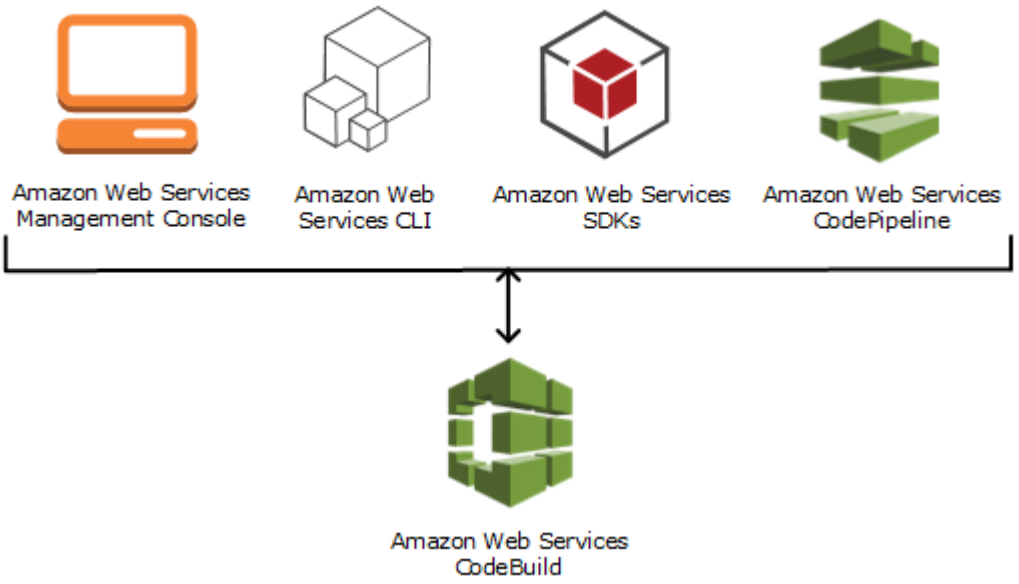
Untuk informasi selengkapnya, lihat AWS CodeBuild.

Mendaftar untuk Akun AWS

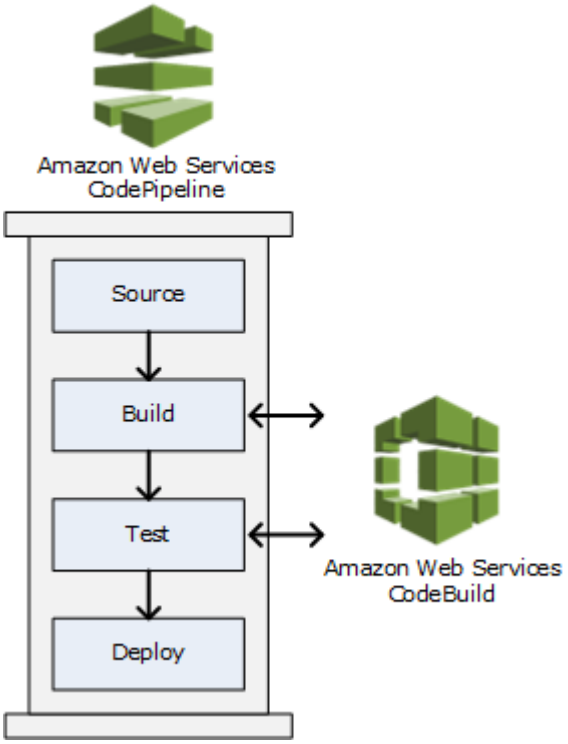
Untuk memulai AWS, Anda membutuhkan Akun AWS. Ada dua cara untuk mendaftar AWS. Jika Anda ingin kontrol penuh atas konfigurasi akun dan di mana konten Anda disimpan, atau jika Anda memiliki beban kerja yang diatur seperti HIPAA atau FedRAMP, gunakan Daftar untuk (lanjutan). AWS Jika Anda adalah pengguna baru dan ingin menggunakan login yang sudah Anda miliki untuk masuk AWS dan membuat AWS lingkungan dengan default yang telah dikonfigurasi sebelumnya di pilih Wilayah AWS, gunakan Daftar untuk. AWS

Bagaimana cara menjalankan CodeBuild

Anda dapat menggunakan AWS CodePipeline konsol AWS CodeBuild atau untuk menjalankan CodeBuild. Anda juga dapat mengotomatiskan proses CodeBuild dengan menggunakan AWS Command Line Interface (AWS CLI) atau AWS SDK.



Seperti yang ditunjukkan diagram berikut, Anda dapat menambahkan CodeBuild sebagai tindakan build atau pengujian ke tahap build atau pengujian pipeline di AWS CodePipeline. AWS CodePipeline adalah layanan pengiriman berkelanjutan yang dapat Anda gunakan untuk memodelkan, memvisualisasikan, dan mengotomatiskan langkah-langkah yang diperlukan untuk merilis kode Anda. Ini termasuk membangun kode Anda. Pipeline adalah konstruksi alur kerja yang menjelaskan bagaimana perubahan kode melalui proses rilis.



Untuk digunakan CodePipeline untuk membuat pipeline dan kemudian menambahkan tindakan CodeBuild build atau test, lihatGunakan CodeBuild dengan CodePipeline. Untuk informasi selengkapnya CodePipeline, lihat Panduan AWS CodePipeline Pengguna.

CodeBuild Konsol juga menyediakan cara untuk mencari sumber daya Anda dengan cepat, seperti repositori, membangun proyek, aplikasi penerapan, dan saluran pipa. Pilih Buka sumber daya atau tekan / tombol, lalu masukkan nama sumber daya. Setiap kecocokan akan muncul di daftar. Pencarian peka huruf besar/kecil. Anda hanya melihat sumber daya yang izin untuk menampilkannya Anda memiliki. Untuk informasi selengkapnya, lihat Menampilkan sumber daya di konsol.

Harga untuk CodeBuild

Untuk informasi, lihat hargaCodeBuild .

Bagaimana saya memulai CodeBuild?

Kami menyarankan agar Anda menyelesaikan langkah berikut:

1. Pelajari lebih lanjut CodeBuild dengan membaca informasi diKonsep.
2. Bereksperimenlah dengan CodeBuild skenario contoh dengan mengikuti instruksi diMemulai menggunakan konsol.
3. Gunakan CodeBuild dalam skenario Anda sendiri dengan mengikuti instruksi diMerencanakan build.

AWS CodeBuild konsep

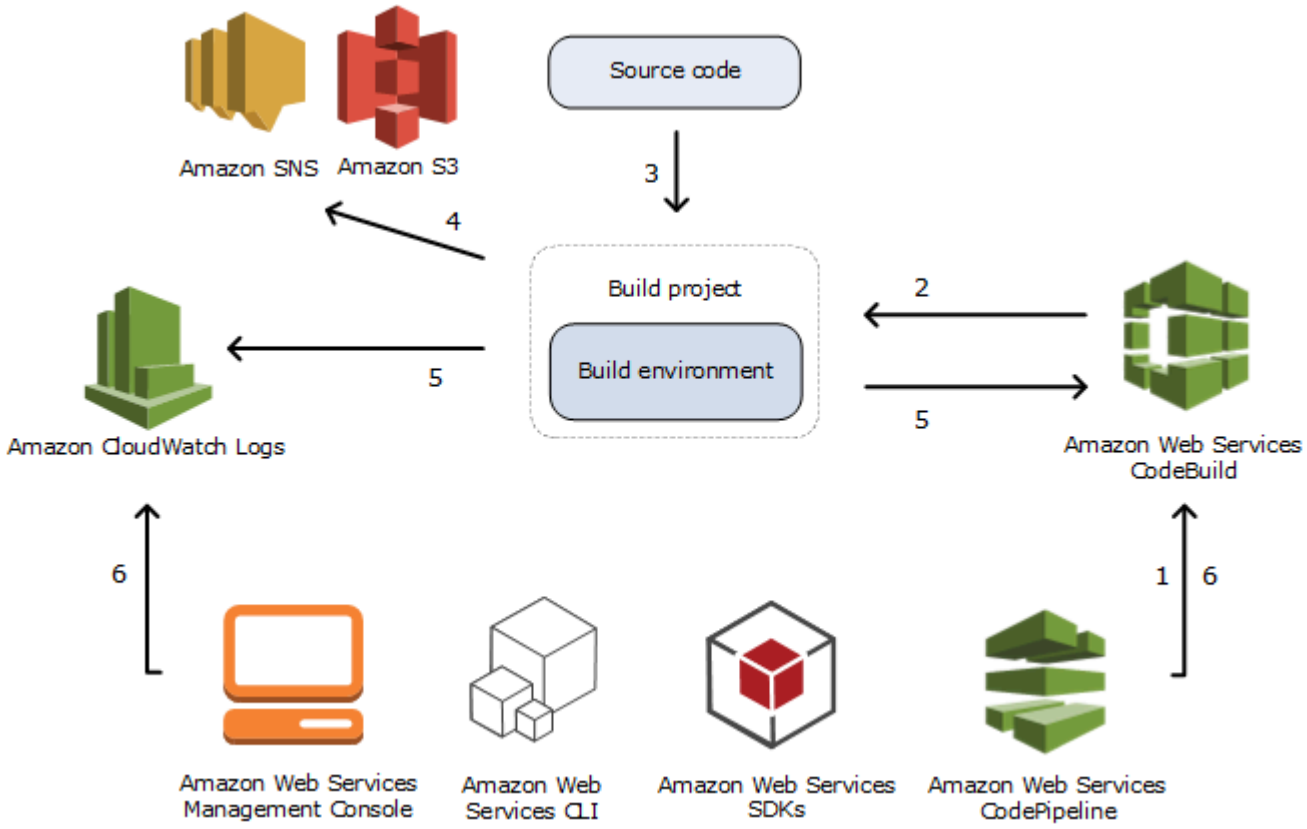
Konsep-konsep berikut ini penting untuk memahami cara CodeBuild kerja.

Topik

- Bagaimana cara CodeBuild kerja
- Langkah selanjutnya

Bagaimana cara CodeBuild kerja

Diagram berikut menunjukkan apa yang terjadi ketika Anda menjalankan build dengan CodeBuild:



1. Sebagai masukan, Anda harus CodeBuild menyediakan proyek build. Proyek build mencakup informasi tentang cara menjalankan build, termasuk tempat mendapatkan kode sumber, lingkungan build mana yang akan digunakan, perintah build mana yang akan dijalankan, dan tempat menyimpan output build. Lingkungan build mewakili kombinasi sistem operasi, runtime bahasa pemrograman, dan alat yang CodeBuild digunakan untuk menjalankan build. Untuk informasi lebih lanjut, lihat:

   - Membuat proyek build
   - Membangun referensi lingkungan

2. CodeBuild menggunakan proyek build untuk menciptakan lingkungan build.
3. CodeBuild mengunduh kode sumber ke lingkungan build dan kemudian menggunakan spesifikasi build (buildspec), seperti yang didefinisikan dalam proyek build atau disertakan langsung dalam kode sumber. Buildspec adalah kumpulan perintah build dan pengaturan terkait, dalam format YAMAL, yang CodeBuild digunakan untuk menjalankan build. Untuk informasi selengkapnya, lihat Referensi Buildspec.

4. Jika ada keluaran build, lingkungan build akan mengunggah outputnya ke bucket S3. Lingkungan build juga dapat menjalankan tugas yang Anda tentukan di buildspec (misalnya, mengirim notifikasi build ke topik Amazon SNS). Sebagai contoh, lihat Buat contoh notifikasi.
5. Saat build berjalan, lingkungan build mengirimkan informasi ke CodeBuild dan Amazon CloudWatch Logs.
6. Saat build berjalan, Anda dapat menggunakan AWS CodeBuild konsol, AWS CLI, atau AWS SDK untuk mendapatkan informasi build yang diringkas dari CodeBuild dan informasi build terperinci dari Amazon CloudWatch Logs. Jika Anda menggunakan AWS CodePipeline untuk menjalankan build, Anda bisa mendapatkan informasi build terbatas dari CodePipeline.

Langkah selanjutnya

Sekarang setelah Anda tahu lebih banyak tentang AWS CodeBuild, kami merekomendasikan langkah-langkah selanjutnya:

1. Bereksperimenlah dengan CodeBuild skenario contoh dengan mengikuti instruksi diMemulai menggunakan konsol.
2. Gunakan CodeBuild dalam skenario Anda sendiri dengan mengikuti instruksi diMerencanakan build.

Memulai dengan CodeBuild

Dalam tutorial berikut, Anda menggunakan AWS CodeBuild untuk membangun kumpulan file input kode sumber sampel ke dalam versi kode sumber yang dapat digunakan.

Kedua tutorial memiliki input dan hasil yang sama, tetapi satu menggunakan AWS CodeBuild konsol dan yang lainnya menggunakan AWS CLI.

> **Important**
>
> Kami tidak menyarankan Anda menggunakan akun AWS root Anda untuk menyelesaikan tutorial ini.

Topik

- Memulai dengan AWS CodeBuild menggunakan konsol
- Memulai dengan AWS CodeBuild menggunakan AWS CLI

Memulai dengan AWS CodeBuild menggunakan konsol

Dalam tutorial ini, Anda menggunakan AWS CodeBuild untuk membangun kumpulan file input kode sumber sampel (membangun artefak masukan atau membangun input) ke dalam versi kode sumber yang dapat digunakan (membangun artefak keluaran atau membangun output). Secara khusus, Anda menginstruksikan CodeBuild untuk menggunakan Apache Maven, alat build umum, untuk membangun satu set file kelas Java ke dalam file Java Archive (JAR). Anda tidak perlu terbiasa dengan Apache Maven atau Java untuk menyelesaikan tutorial ini.

Anda dapat bekerja dengan CodeBuild melalui CodeBuild konsol AWS CodePipeline,, AWS CLI, atau AWS SDK. Tutorial ini menunjukkan cara menggunakan CodeBuild konsol. Untuk informasi tentang penggunaan CodePipeline, lihatGunakan CodeBuild dengan CodePipeline.

> **Important**
>
> Langkah-langkah dalam tutorial ini mengharuskan Anda membuat sumber daya (misalnya, bucket S3) yang dapat mengakibatkan tagihan ke AWS akun Anda. Ini termasuk kemungkinan biaya untuk CodeBuild dan untuk AWS sumber daya dan tindakan yang terkait

dengan Amazon S3, AWS KMS, dan CloudWatch Log. Untuk informasi selengkapnya, lihat AWS CodeBuild harga, harga Amazon S3, AWS Key Management Service harga, dan CloudWatch harga Amazon.

Topik

- Langkah 1: Buat kode sumber
- Langkah 2: Buat file buildspec
- Langkah 3: Buat dua bucket S3
- Langkah 4: Unggah kode sumber dan file buildspec
- Langkah 5: Buat proyek pembangunan
- Langkah 6: Jalankan build
- Langkah 7: Lihat informasi build yang diringkas
- Langkah 8: Lihat informasi build terperinci
- Langkah 9: Dapatkan artefak keluaran build
- Langkah 10: Hapus bucket S3
- Membungkus

Langkah 1: Buat kode sumber

(Bagian dari:Memulai dengan AWS CodeBuild menggunakan konsol)

Pada langkah ini, Anda membuat kode sumber yang ingin Anda CodeBuild bangun ke bucket keluaran. Kode sumber ini terdiri dari dua file kelas Java dan file Apache Maven Project Object Model (POM).

1. Dalam direktori kosong di komputer atau instance lokal Anda, buat struktur direktori ini.

```
(root directory name)
    `-- src
        |-- main
        |     `-- java
        `-- test
              `-- java
```

2. Menggunakan editor teks pilihan Anda, buat file ini, beri nama`MessageUtil.java`, lalu simpan di `src/main/java` direktori.

```
public class MessageUtil {
  private String message;

  public MessageUtil(String message) {
    this.message = message;
  }

  public String printMessage() {
    System.out.println(message);
    return message;
  }

  public String salutationMessage() {
    message = "Hi!" + message;
    System.out.println(message);
    return message;
  }
}
```

File kelas ini membuat sebagai output string karakter yang diteruskan ke dalamnya. `MessageUtil`Konstruktor menetapkan string karakter. `printMessage`Metode ini menciptakan output. Out `salutationMessage` put metode `Hi!` diikuti oleh string karakter.

3. Buat file ini, beri nama`TestMessageUtil.java`, lalu simpan di `/src/test/java` direktori.

```
import org.junit.Test;
import org.junit.Ignore;
import static org.junit.Assert.assertEquals;

public class TestMessageUtil {

  String message = "Robert";
  MessageUtil messageUtil = new MessageUtil(message);

  @Test
  public void testPrintMessage() {
    System.out.println("Inside testPrintMessage()");
    assertEquals(message,messageUtil.printMessage());
  }
```

```
  @Test
  public void testSalutationMessage() {
    System.out.println("Inside testSalutationMessage()");
    message = "Hi!" + "Robert";
    assertEquals(message,messageUtil.salutationMessage());
  }
}
```

File kelas ini menetapkan `message` variabel di `MessageUtil` kelas ke`Robert`. Kemudian menguji untuk melihat apakah `message` variabel berhasil diatur dengan memeriksa apakah string `Robert` dan `Hi!Robert` muncul di output.

4. Buat file ini, beri nama`pom.xml`, lalu simpan di direktori root (tingkat atas).

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/
maven-v4_0_0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>org.example</groupId>
  <artifactId>messageUtil</artifactId>
  <version>1.0</version>
  <packaging>jar</packaging>
  <name>Message Utility Java Sample App</name>
  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>4.11</version>
      <scope>test</scope>
    </dependency>
  </dependencies>
  <build>
    <plugins>
      <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-compiler-plugin</artifactId>
        <version>3.8.0</version>
      </plugin>
    </plugins>
  </build>
</project>
```

Apache Maven menggunakan instruksi dalam file ini untuk mengonversi `TestMessageUtil.java` file `MessageUtil.java` dan menjadi file bernama `messageUtil-1.0.jar` dan kemudian menjalankan pengujian yang ditentukan.

Pada titik ini, struktur direktori Anda akan terlihat seperti ini.

```
(root directory name)
    |-- pom.xml
    `-- src
         |-- main
         |     `-- java
         |           `-- MessageUtil.java
         `-- test
               `-- java
                     `-- TestMessageUtil.java
```

Langkah 2: Buat file buildspec

(Langkah sebelumnya: Langkah 1: Buat kode sumber)

Pada langkah ini, Anda membuat file spesifikasi build (spesifikasi build). Buildspec adalah kumpulan perintah build dan pengaturan terkait, dalam format YAML, yang digunakan untuk menjalankan CodeBuild build. Tanpa spesifikasi build, CodeBuild tidak dapat berhasil mengonversi input build Anda menjadi output build atau menemukan artefak keluaran build di lingkungan build untuk diunggah ke bucket keluaran Anda.

Buat file ini, beri nama`buildspec.yml`, lalu simpan di direktori root (tingkat atas).

```
version: 0.2

phases:
  install:
    runtime-versions:
      java: corretto11
  pre_build:
    commands:
      - echo Nothing to do in the pre_build phase...
  build:
    commands:
      - echo Build started on `date`
```

```
      - mvn install
  post_build:
    commands:
      - echo Build completed on `date`
artifacts:
  files:
    - target/messageUtil-1.0.jar
```

Important

Karena deklarasi spesifikasi build harus YAML yang valid, spasi dalam deklarasi spesifikasi build penting. Jika jumlah spasi dalam deklarasi spesifikasi build Anda tidak cocok dengan yang ini, build mungkin segera gagal. Anda dapat menggunakan validator YAML untuk menguji apakah deklarasi spesifikasi build Anda adalah YAML yang valid.

Note

Alih-alih menyertakan file spesifikasi build dalam kode sumber Anda, Anda dapat mendeklarasikan perintah build secara terpisah saat Anda membuat proyek build. Ini sangat membantu jika Anda ingin membangun kode sumber Anda dengan perintah build yang berbeda tanpa memperbarui repositori kode sumber Anda setiap kali. Untuk informasi selengkapnya, lihat Sintaks Buildspec.

Dalam deklarasi spesifikasi build ini:

- `version`mewakili versi standar spesifikasi build yang digunakan. Deklarasi spesifikasi build ini menggunakan versi terbaru,`0.2`.
- `phases`mewakili fase build di mana Anda dapat menginstruksikan CodeBuild untuk menjalankan perintah. Fase build ini tercantum di sini sebagai `install`pre_build`,`build`,, dan`post_build`. Anda tidak dapat mengubah ejaan nama fase build ini, dan Anda tidak dapat membuat lebih banyak nama fase build.

  Dalam contoh ini, selama `build` fase, CodeBuild jalankan `mvn install` perintah. Perintah ini menginstruksikan Apache Maven untuk mengkompilasi, menguji, dan mengemas file kelas Java yang dikompilasi ke dalam artefak keluaran build. Untuk kelengkapan, beberapa `echo` perintah ditempatkan di setiap fase build dalam contoh ini. Ketika Anda melihat informasi build terperinci

nanti dalam tutorial ini, output dari `echo` perintah ini dapat membantu Anda lebih memahami bagaimana CodeBuild menjalankan perintah dan dalam urutan mana. (Meskipun semua fase build disertakan dalam contoh ini, Anda tidak diharuskan untuk menyertakan fase build jika Anda tidak berencana untuk menjalankan perintah apa pun selama fase itu.) Untuk setiap fase build, CodeBuild jalankan setiap perintah yang ditentukan, satu per satu, dalam urutan yang tercantum, dari awal hingga akhir.

- `artifacts`mewakili kumpulan artefak keluaran build yang di CodeBuild unggah ke bucket keluaran. `files`mewakili file untuk disertakan dalam output build. CodeBuild mengunggah `messageUtil-1.0.jar` file tunggal yang ditemukan di direktori `target` relatif di lingkungan build. Nama file `messageUtil-1.0.jar` dan nama direktori `target` didasarkan pada cara Apache Maven membuat dan menyimpan artefak keluaran build hanya untuk contoh ini. Dalam build Anda sendiri, nama file dan direktori ini berbeda.

Untuk informasi selengkapnya, lihat Referensi Buildspec.

Pada titik ini, struktur direktori Anda akan terlihat seperti ini.

```
(root directory name)
    |-- pom.xml
    |-- buildspec.yml
    `-- src
         |-- main
         |     `-- java
         |           `-- MessageUtil.java
         `-- test
               `-- java
                     `-- TestMessageUtil.java
```

Langkah 3: Buat dua bucket S3

(Langkah sebelumnya: Langkah 2: Buat file buildspec)

Meskipun Anda dapat menggunakan satu bucket untuk tutorial ini, dua bucket membuatnya lebih mudah untuk melihat dari mana input build berasal dan ke mana output build pergi.

- Salah satu bucket ini (bucket input) menyimpan input build. Dalam tutorial ini, nama bucket input ini adalah`codebuild-`*`region-ID`*`-`*`account-ID`*`-input-bucket`, di mana *`region-ID`* adalah AWS Region dari bucket dan *`account-ID`* merupakan ID AWS akun Anda.

- Bucket lainnya (bucket keluaran) menyimpan output build. Dalam tutorial ini, nama bucket keluaran ini adalah`codebuild-region-ID-account-ID-output-bucket`.

Jika Anda memilih nama yang berbeda untuk bucket ini, pastikan untuk menggunakannya di seluruh tutorial ini.

Kedua bucket ini harus berada di Wil AWS ayah yang sama dengan bangunan Anda. Misalnya, jika Anda menginstruksikan CodeBuild untuk menjalankan build di Wilayah AS Timur (Ohio), bucket ini juga harus berada di Wilayah AS Timur (Ohio).

Untuk informasi selengkapnya, lihat Membuat Bucket di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.

> **Note**
>
> Meskipun CodeBuild juga mendukung input build yang disimpan di CodeCommit GitHub,, dan repositori Bitbucket, tutorial ini tidak menunjukkan cara menggunakannya. Untuk informasi selengkapnya, lihat Merencanakan build.

Langkah 4: Unggah kode sumber dan file buildspec

(Langkah sebelumnya: Langkah 3: Buat dua bucket S3)

Pada langkah ini, Anda menambahkan kode sumber dan membangun file spesifikasi ke bucket input.

Menggunakan utilitas zip sistem operasi Anda, buat file bernama `MessageUtil.zip` yang mencakup`MessageUtil.java`,`TestMessageUtil.java`,`pom.xml`, dan`buildspec.yml`.

Struktur direktori `MessageUtil.zip` file harus terlihat seperti ini.

```
MessageUtil.zip
    |-- pom.xml
    |-- buildspec.yml
    `-- src
         |-- main
         |     `-- java
         |           `-- MessageUtil.java
         `-- test
```

```
            `-- java
                 `-- TestMessageUtil.java
```

> **Important**
>
> Jangan sertakan *(root directory name)* direktori, hanya direktori dan file di *(root directory name)* direktori.

Unggah `MessageUtil.zip` file ke bucket input bernama`codebuild-`*region-ID*`-`*account-ID*`-input-bucket`.

> **Important**
>
> Untuk repositori CodeCommit GitHub,, dan Bitbucket, menurut konvensi, Anda harus menyimpan file spesifikasi build yang dinamai `buildspec.yml` di root (tingkat atas) setiap repositori atau menyertakan deklarasi spesifikasi build sebagai bagian dari definisi proyek build. Jangan membuat file ZIP yang berisi kode sumber repositori dan membangun file spesifikasi.
>
> Untuk input build yang disimpan di bucket S3 saja, Anda harus membuat file ZIP yang berisi kode sumber dan, menurut konvensi, file spesifikasi build yang dinamai `buildspec.yml` di root (tingkat atas) atau menyertakan deklarasi spesifikasi build sebagai bagian dari definisi proyek build.
>
> Jika Anda ingin menggunakan nama yang berbeda untuk file spesifikasi build Anda, atau Anda ingin mereferensikan spesifikasi build di lokasi selain root, Anda dapat menentukan penggantian spesifikasi build sebagai bagian dari definisi proyek build. Untuk informasi selengkapnya, lihat Nama file buildspec dan lokasi penyimpanan.

Langkah 5: Buat proyek pembangunan

(Langkah sebelumnya: Langkah 4: Unggah kode sumber dan file buildspec)

Pada langkah ini, Anda membuat proyek build yang AWS CodeBuild digunakan untuk menjalankan build. Proyek build mencakup informasi tentang cara menjalankan build, termasuk di mana mendapatkan kode sumber, lingkungan build mana yang akan digunakan, perintah build mana yang akan dijalankan, dan tempat menyimpan output build. Lingkungan build mewakili kombinasi sistem operasi, runtime bahasa pemrograman, dan alat CodeBuild yang digunakan untuk menjalankan build.

Lingkungan build diekspresikan sebagai gambar Docker. Untuk informasi selengkapnya, lihat ik htisar Docker di situs Docker Docker.

Untuk lingkungan build ini, Anda menginstruksikan CodeBuild untuk menggunakan image Docker yang berisi versi Java Development Kit (JDK) dan Apache Maven.

Untuk membuat proyek pembangunan

1. Masuk ke Konsol Manajemen AWS dan buka AWS CodeBuild konsol di https://console.aws.amazon.com/codesuite/codebuild/home.
2. Gunakan pemilih AWS wilayah untuk memilih Wil AWS ayah yang CodeBuild didukung. Untuk informasi selengkapnya, lihat AWS CodeBuild titik akhir dan kuota di Referensi Umum Amazon Web Services.
3. Jika halaman CodeBuild informasi ditampilkan, pilih Buat proyek build. Jika tidak, pada panel navigasi, per luas Bang un, pilih Bang un proyek, lalu pilih Buat proyek build.
4. Pada halaman Buat proyek build, di konfigurasi proyek, untuk nama proyek, masukkan nama untuk proyek build ini (dalam contoh ini,`codebuild-demo-project`). Nama proyek build harus unik di setiap AWS akun. Jika Anda menggunakan nama yang berbeda, pastikan untuk menggunakannya di seluruh tutorial ini.

> **Note**
>
> Pada halaman Buat proyek build, Anda mungkin melihat pesan kesalahan yang mirip dengan berikut ini: Anda tidak berwenang untuk melakukan operasi ini. . Ini kemungkinan besar karena Anda masuk Konsol Manajemen AWS sebagai pengguna yang tidak memiliki izin untuk membuat proyek build. Untuk memperbaikinya, keluar dari Konsol Manajemen AWS, lalu masuk kembali dengan kredensibilitas milik salah satu entitas IAM berikut:
>
> - Pengguna administrator di AWS akun Anda. Untuk informasi selengkapnya, lihat Membuat pengguna dan grup Akun AWS root pertama Anda di Panduan pengguna.
> - Pengguna di AWS akun Anda dengan kebijakan`AWSCodeBuildAdminAccess`,`AmazonS3ReadOnlyAccess`, dan ter `IAMFullAccess` kelola yang dilampirkan ke pengguna tersebut atau grup IAM tempat pengguna tersebut berada. Jika Anda tidak memiliki pengguna atau grup di AWS akun Anda dengan izin ini, dan Anda tidak dapat menambahkan izin ini ke pengguna atau grup Anda, hubungi administrator AWS akun Anda untuk mendapatkan bantuan.

Untuk informasi selengkapnya, lihat AWS kebijakan terkelola (standar) untuk AWS CodeBuild.

Kedua opsi menyertakan izin administrator yang memungkinkan Anda membuat proyek build sehingga Anda dapat menyelesaikan tutorial ini. Sebaiknya Anda selalu menggunakan izin minimum yang diperlukan untuk menyelesaikan tugas Anda. Untuk informasi selengkapnya, lihat AWS CodeBuild referensi izin.

5. Di Sumber, untuk Penyedia Sumber, pilih Amazon S3.
6. Untuk Bucket, pilih codebuild- ***region-ID*** - ***account-ID*** -input- bucket.
7. Untuk kunci objek S3, masukkan**MessageUtil.zip**.
8. Di Lingkungan, untuk gambar Lingkungan, biarkan gambar terkelola dipilih.
9. Untuk Sistem operasi, pilih Amazon Linux.
10. Untuk Runtime (s), pilih Standar.
11. Untuk Gambar, pilih aws/codebuild /amaz onlinux-x86_64-standard:corretto11.
12. Di Peran Layanan, biarkan Peran layanan baru dipilih, dan biarkan nama Peran tidak berubah.
13. Untuk Buildspec, biarkan Gunakan file buildspec dipilih.
14. Di Artifacts, untuk Type, pilih Amazon S3.
15. Untuk nama Bucket, pilih codebuild- ***region-ID*** - ***account-ID*** -output- bucket.
16. Biarkan Nama dan Jalur kosong.
17. Pilih Buat proyek pembangunan.

Langkah 6: Jalankan build

(Langkah sebelumnya: Langkah 5: Buat proyek pembangunan)

Pada langkah ini, Anda menginstruksikan AWS CodeBuild untuk menjalankan build dengan pengaturan dalam proyek build.

Untuk menjalankan build

1. Buka AWS CodeBuild konsol di https://console.aws.amazon.com/codesuite/codebuild/home.
2. Di panel navigasi, pilih Bangun proyek.

3. Dalam daftar proyek build, pilih codebuild-demo-project, lalu pilih Mulai build. Pembangunan dimulai segera.

Langkah 7: Lihat informasi build yang diringkas

(Langkah sebelumnya: Langkah 6: Jalankan build)

Pada langkah ini, Anda melihat informasi ringkasan tentang status build Anda.

Untuk melihat informasi build yang diringkas

1. Jika *<build-ID>* halaman codebuild-demo-project: tidak ditampilkan, di bilah navigasi, pilih Build history. Selanjutnya, dalam daftar proyek build, untuk Project, pilih tautan Buil d run untuk codebuil d-demo-project. Seharusnya hanya ada satu tautan yang cocok. (Jika Anda telah menyelesaikan tutorial ini sebelumnya, pilih tautan dengan nilai terbaru di kolom Selesai.)
2. Pada halaman status Build, di Detail T ahap, fase build berikut harus ditampilkan, dengan Ber hasil di kolom Status:

   - DISERAHKAN
   - ANTRI
   - PENYEDIAAN
   - UNDUH_SUMBER
   - MENGINSTAL
   - PRE_BUILD
   - MEMBANGUN
   - POST_BUILD
   - UNGGAH_ARTEFAK
   - MENYELESAIKAN
   - SELESAI

   Di Status Bang un, Suk ses harus ditampilkan.

   Jika Anda melihat Sed ang Ber jalan sebagai gantinya, pilih tombol refresh.
3. Di samping setiap fase build, nilai Durasi menunjukkan berapa lama fase build berlangsung. Nilai Waktu akhir menunjukkan kapan fase build itu berakhir.

Langkah 8: Lihat informasi build terperinci

(Langkah sebelumnya: Langkah 7: Lihat informasi build yang diringkas)

Pada langkah ini, Anda melihat informasi terperinci tentang build Anda di CloudWatch Log.

> **Note**
>
> Untuk melindungi informasi sensitif, berikut ini disembunyikan di CodeBuild log:
>
> - AWS ID kunci akses. Untuk informasi selengkapnya, lihat Mengelola Kunci Akses untuk Pengguna IAM di Panduan AWS Identity and Access Management Pengguna.
> - String ditentukan menggunakan Parameter Store. Untuk informasi selengkapnya, lihat Panduan Konsol Penyimpanan Parameter Manajer Sistem dan Panduan Konsol Penyimpanan Parameter Manajer Sistem di Panduan Pengguna Amazon EC2 Systems Manager.
> - String ditentukan menggunakan AWS Secrets Manager. Untuk informasi selengkapnya, lihat Manajemen kunci.

Untuk melihat informasi build terperinci

1. Dengan halaman detail build yang masih ditampilkan dari langkah sebelumnya, 10.000 baris terakhir dari log build ditampilkan di log build. Untuk melihat seluruh log CloudWatch log build, pilih tautan Lihat seluruh log.
2. Di aliran CloudWatch log log, Anda dapat menelusuri peristiwa log. Secara default, hanya set terakhir dari peristiwa log yang ditampilkan. Untuk melihat peristiwa log sebelumnya, gulir ke awal daftar.
3. Dalam tutorial ini, sebagian besar peristiwa log berisi informasi verbose tentang CodeBuild mengunduh dan menginstal file ketergantungan build ke lingkungan build-nya, yang mungkin tidak Anda pedulikan. Anda dapat menggunakan kotak Filter peristiwa untuk mengurangi informasi yang ditampilkan. Misalnya, jika Anda memasukkan `"[INFO]"` peristiwa Filter, hanya peristiwa yang berisi yang `[INFO]` ditampilkan. Untuk informasi selengkapnya, lihat Filter dan sintaks pola di Panduan CloudWatch Pengguna Amazon.

Langkah 9: Dapatkan artefak keluaran build

(Langkah sebelumnya: Langkah 8: Lihat informasi build terperinci)

Pada langkah ini, Anda mendapatkan `messageUtil-1.0.jar` file yang CodeBuild dibangun dan diunggah ke bucket keluaran.

Anda dapat menggunakan CodeBuild konsol atau konsol Amazon S3 untuk menyelesaikan langkah ini.

Untuk mendapatkan artefak keluaran build (AWS CodeBuild konsol)

1. Dengan CodeBuild konsol masih terbuka dan halaman detail build masih ditampilkan dari langkah sebelumnya, pilih tab Detail build dan gulir ke bawah ke bagian Artifacts.

   > **Note**
   >
   > Jika halaman detail build tidak ditampilkan, di bilah navigasi, pilih Ri wayat build, lalu pilih tautan Build run.

2. Tautan ke folder Amazon S3 berada di bawah lokasi ungg ahan Artifacts. Tautan ini membuka folder di Amazon S3 tempat Anda menemukan file artefak keluaran `messageUtil-1.0.jar` build.

Untuk mendapatkan artefak keluaran build (konsol Amazon S3)

1. Buka konsol Amazon S3 di https://console.aws.amazon.com/s3/.
2. Buka `codebuild-`*`region-ID`*`-`*`account-ID`*`-output-bucket`.
3. Buka folder `codebuild-demo-project`.
4. Buka `target` folder, tempat Anda menemukan file artefak keluaran `messageUtil-1.0.jar` build.

Langkah 10: Hapus bucket S3

(Langkah sebelumnya: Langkah 9: Dapatkan artefak keluaran build)

Untuk mencegah tagihan berkelanjutan ke AWS akun Anda, Anda dapat menghapus bucket input dan output yang digunakan dalam tutorial ini. Untuk petunjuk, lihat Menghapus atau Mengosongkan Bucket di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.

Jika Anda menggunakan pengguna IAM atau pengguna IAM administrator untuk menghapus bucket ini, pengguna harus memiliki lebih banyak izin akses. Tambahkan pernyataan berikut antara penanda (*### BEGIN ADDING STATEMENT HERE ###*dan*### END ADDING STATEMENTS HERE ###*) ke kebijakan akses yang ada untuk pengguna.

Elips (...) dalam pernyataan ini digunakan untuk singkatnya. Jangan menghapus pernyataan apa pun dalam kebijakan akses yang ada. Jangan masukkan elips ini ke dalam kebijakan.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:DeleteBucket",
        "s3:DeleteObject"
      ],
      "Resource": "*"
    }
  ]
}
```

Membungkus

Dalam tutorial ini, Anda AWS CodeBuild biasa membangun satu set file kelas Java ke dalam file JAR. Anda kemudian melihat hasil build.

Anda sekarang dapat mencoba menggunakan CodeBuild dalam skenario Anda sendiri. Ikuti petunjuk dalam Merencanakan build. Jika Anda belum merasa siap, Anda mungkin ingin mencoba membangun beberapa sampel. Untuk informasi selengkapnya, lihat Gunakan sampel berbasis kasus untuk CodeBuild.

Memulai dengan AWS CodeBuild menggunakan AWS CLI

Dalam tutorial ini, Anda menggunakan AWS CodeBuild untuk membangun kumpulan file input kode sumber sampel (disebut membangun artefak input atau membangun input) ke dalam versi kode sumber yang dapat digunakan (disebut artefak keluaran build atau output build). Secara khusus, Anda menginstruksikan CodeBuild untuk menggunakan Apache Maven, alat build umum, untuk membangun satu set file kelas Java ke dalam file Java Archive (JAR). Anda tidak perlu terbiasa dengan Apache Maven atau Java untuk menyelesaikan tutorial ini.

Anda dapat bekerja dengan CodeBuild melalui CodeBuild konsol AWS CodePipeline,, AWS CLI, atau AWS SDK. Tutorial ini menunjukkan cara menggunakan CodeBuild dengan AWS CLI. Untuk informasi tentang penggunaan CodePipeline, lihatGunakan CodeBuild dengan CodePipeline.

> **Important**
>
> Langkah-langkah dalam tutorial ini mengharuskan Anda membuat sumber daya (misalnya, bucket S3) yang dapat mengakibatkan tagihan ke AWS akun Anda. Ini termasuk kemungkinan biaya untuk CodeBuild dan untuk AWS sumber daya dan tindakan yang terkait dengan Amazon S3, AWS KMS, dan CloudWatch Log. Untuk informasi selengkapnya, lihat CodeBuild harga, harga Amazon S3, AWS Key Management Service harga, dan CloudWatch harga Amazon.

Topik

- Langkah 1: Buat kode sumber
- Langkah 2: Buat file buildspec
- Langkah 3: Buat dua bucket S3
- Langkah 4: Unggah kode sumber dan file buildspec
- Langkah 5: Buat proyek pembangunan
- Langkah 6: Jalankan build
- Langkah 7: Lihat informasi build yang diringkas
- Langkah 8: Lihat informasi build terperinci
- Langkah 9: Dapatkan artefak keluaran build
- Langkah 10: Hapus bucket S3
- Membungkus

Langkah 1: Buat kode sumber

(Bagian dari:Memulai dengan AWS CodeBuild menggunakan AWS CLI)

Pada langkah ini, Anda membuat kode sumber yang ingin Anda CodeBuild bangun ke bucket keluaran. Kode sumber ini terdiri dari dua file kelas Java dan file Apache Maven Project Object Model (POM).

1. Dalam direktori kosong di komputer atau instance lokal Anda, buat struktur direktori ini.

```
(root directory name)
    `-- src
         |-- main
         |     `-- java
         `-- test
               `-- java
```

2. Menggunakan editor teks pilihan Anda, buat file ini, beri namaMessageUtil.java, lalu simpan di src/main/java direktori.

```
public class MessageUtil {
  private String message;

  public MessageUtil(String message) {
    this.message = message;
  }

  public String printMessage() {
    System.out.println(message);
    return message;
  }

  public String salutationMessage() {
    message = "Hi!" + message;
    System.out.println(message);
    return message;
  }
}
```

File kelas ini membuat sebagai output string karakter yang diteruskan ke dalamnya. MessageUtilKonstruktor menetapkan string karakter. printMessageMetode ini menciptakan output. Out salutationMessage put metode Hi! diikuti oleh string karakter.

3. Buat file ini, beri nama`TestMessageUtil.java`, lalu simpan di `/src/test/java` direktori.

```
import org.junit.Test;
import org.junit.Ignore;
import static org.junit.Assert.assertEquals;

public class TestMessageUtil {

  String message = "Robert";
  MessageUtil messageUtil = new MessageUtil(message);

  @Test
  public void testPrintMessage() {
    System.out.println("Inside testPrintMessage()");
    assertEquals(message,messageUtil.printMessage());
  }

  @Test
  public void testSalutationMessage() {
    System.out.println("Inside testSalutationMessage()");
    message = "Hi!" + "Robert";
    assertEquals(message,messageUtil.salutationMessage());
  }
}
```

File kelas ini menetapkan `message` variabel di `MessageUtil` kelas ke`Robert`. Kemudian menguji untuk melihat apakah `message` variabel berhasil diatur dengan memeriksa apakah string `Robert` dan `Hi!Robert` muncul di output.

4. Buat file ini, beri namapom`.xml`, lalu simpan di direktori root (tingkat atas).

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/
maven-v4_0_0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>org.example</groupId>
  <artifactId>messageUtil</artifactId>
  <version>1.0</version>
  <packaging>jar</packaging>
  <name>Message Utility Java Sample App</name>
  <dependencies>
    <dependency>
```

```
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>4.11</version>
      <scope>test</scope>
    </dependency>
  </dependencies>
  <build>
    <plugins>
      <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-compiler-plugin</artifactId>
        <version>3.8.0</version>
      </plugin>
    </plugins>
  </build>
</project>
```

Apache Maven menggunakan instruksi dalam file ini untuk mengonversi `TestMessageUtil.java` file `MessageUtil.java` dan menjadi file bernama `messageUtil-1.0.jar` dan kemudian menjalankan pengujian yang ditentukan.

Pada titik ini, struktur direktori Anda akan terlihat seperti ini.

```
(root directory name)
    |-- pom.xml
    `-- src
         |-- main
         |     `-- java
         |           `-- MessageUtil.java
         `-- test
               `-- java
                     `-- TestMessageUtil.java
```

Langkah 2: Buat file buildspec

(Langkah sebelumnya: Langkah 1: Buat kode sumber)

Pada langkah ini, Anda membuat file spesifikasi build (spesifikasi build). Buildspec adalah kumpulan perintah build dan pengaturan terkait, dalam format YAML, yang digunakan untuk menjalankan CodeBuild build. Tanpa spesifikasi build, CodeBuild tidak dapat berhasil mengonversi input build

Anda menjadi output build atau menemukan artefak keluaran build di lingkungan build untuk diunggah ke bucket keluaran Anda.

Buat file ini, beri nama`buildspec.yml`, lalu simpan di direktori root (tingkat atas).

```
version: 0.2

phases:
  install:
    runtime-versions:
      java: corretto11
  pre_build:
    commands:
      - echo Nothing to do in the pre_build phase...
  build:
    commands:
      - echo Build started on `date`
      - mvn install
  post_build:
    commands:
      - echo Build completed on `date`
artifacts:
  files:
    - target/messageUtil-1.0.jar
```

Important

Karena deklarasi spesifikasi build harus YAML yang valid, spasi dalam deklarasi spesifikasi build penting. Jika jumlah spasi dalam deklarasi spesifikasi build Anda tidak cocok dengan yang ini, build mungkin segera gagal. Anda dapat menggunakan validator YAML untuk menguji apakah deklarasi spesifikasi build Anda adalah YAML yang valid.

Note

Alih-alih menyertakan file spesifikasi build dalam kode sumber Anda, Anda dapat mendeklarasikan perintah build secara terpisah saat Anda membuat proyek build. Ini sangat membantu jika Anda ingin membangun kode sumber Anda dengan perintah build yang berbeda tanpa memperbarui repositori kode sumber Anda setiap kali. Untuk informasi selengkapnya, lihat Sintaks Buildspec.

Dalam deklarasi spesifikasi build ini:

- `version`mewakili versi standar spesifikasi build yang digunakan. Deklarasi spesifikasi build ini menggunakan versi terbaru,`0.2`.
- `phases`mewakili fase build di mana Anda dapat menginstruksikan CodeBuild untuk menjalankan perintah. Fase build ini tercantum di sini sebagai `install`,`pre_build`,`build`, dan`post_build`. Anda tidak dapat mengubah ejaan nama fase build ini, dan Anda tidak dapat membuat lebih banyak nama fase build.

  Dalam contoh ini, selama `build` fase, CodeBuild jalankan `mvn install` perintah. Perintah ini menginstruksikan Apache Maven untuk mengkompilasi, menguji, dan mengemas file kelas Java yang dikompilasi ke dalam artefak keluaran build. Untuk kelengkapan, beberapa `echo` perintah ditempatkan di setiap fase build dalam contoh ini. Ketika Anda melihat informasi build terperinci nanti dalam tutorial ini, output dari `echo` perintah ini dapat membantu Anda lebih memahami bagaimana CodeBuild menjalankan perintah dan dalam urutan mana. (Meskipun semua fase build disertakan dalam contoh ini, Anda tidak diharuskan untuk menyertakan fase build jika Anda tidak berencana untuk menjalankan perintah apa pun selama fase itu.) Untuk setiap fase build, CodeBuild jalankan setiap perintah yang ditentukan, satu per satu, dalam urutan yang tercantum, dari awal hingga akhir.
- `artifacts`mewakili kumpulan artefak keluaran build yang di CodeBuild unggah ke bucket keluaran. `files`mewakili file untuk disertakan dalam output build. CodeBuild mengunggah `messageUtil-1.0.jar` file tunggal yang ditemukan di direktori `target` relatif di lingkungan build. Nama file `messageUtil-1.0.jar` dan nama direktori `target` didasarkan pada cara Apache Maven membuat dan menyimpan artefak keluaran build hanya untuk contoh ini. Dalam build Anda sendiri, nama file dan direktori ini berbeda.

Untuk informasi selengkapnya, lihat Referensi Buildspec.

Pada titik ini, struktur direktori Anda akan terlihat seperti ini.

```
(root directory name)
    |-- pom.xml
    |-- buildspec.yml
    `-- src
         |-- main
         |     `-- java
         |           `-- MessageUtil.java
         `-- test
```

```
                `-- java
                      `-- TestMessageUtil.java
```

Langkah 3: Buat dua bucket S3

(Langkah sebelumnya: Langkah 2: Buat file buildspec)

Meskipun Anda dapat menggunakan satu bucket untuk tutorial ini, dua bucket membuatnya lebih mudah untuk melihat dari mana input build berasal dan ke mana output build pergi.

- Salah satu bucket ini (bucket input) menyimpan input build. Dalam tutorial ini, nama bucket input ini adalah`codebuild-`*`region-ID`*`-`*`account-ID`*`-input-bucket`, di mana *`region-ID`* adalah AWS Region dari bucket dan *`account-ID`* merupakan ID AWS akun Anda.
- Bucket lainnya (bucket keluaran) menyimpan output build. Dalam tutorial ini, nama bucket keluaran ini adalah`codebuild-`*`region-ID`*`-`*`account-ID`*`-output-bucket`.

Jika Anda memilih nama yang berbeda untuk bucket ini, pastikan untuk menggunakannya di seluruh tutorial ini.

Kedua bucket ini harus berada di Wil AWS ayah yang sama dengan bangunan Anda. Misalnya, jika Anda menginstruksikan CodeBuild untuk menjalankan build di Wilayah AS Timur (Ohio), bucket ini juga harus berada di Wilayah AS Timur (Ohio).

Untuk informasi selengkapnya, lihat Membuat Bucket di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.

> **Note**
>
> Meskipun CodeBuild juga mendukung input build yang disimpan di CodeCommit GitHub,, dan repositori Bitbucket, tutorial ini tidak menunjukkan cara menggunakannya. Untuk informasi selengkapnya, lihat Merencanakan build.

Langkah 4: Unggah kode sumber dan file buildspec

(Langkah sebelumnya: Langkah 3: Buat dua bucket S3)

Pada langkah ini, Anda menambahkan kode sumber dan membangun file spesifikasi ke bucket input.

Menggunakan utilitas zip sistem operasi Anda, buat file bernama `MessageUtil.zip` yang mencakup`MessageUtil.java`,`TestMessageUtil.java`,`pom.xml`, dan`buildspec.yml`.

Struktur direktori `MessageUtil.zip` file harus terlihat seperti ini.

```
MessageUtil.zip
    |-- pom.xml
    |-- buildspec.yml
    `-- src
         |-- main
         |     `-- java
         |           `-- MessageUtil.java
         `-- test
               `-- java
                     `-- TestMessageUtil.java
```

> **Important**
>
> Jangan sertakan *`(root directory name)`* direktori, hanya direktori dan file di *`(root directory name)`* direktori.

Unggah `MessageUtil.zip` file ke bucket input bernama`codebuild-`*`region-ID`*`-`*`account-ID`*`-input-bucket`.

> **Important**
>
> Untuk repositori CodeCommit GitHub,, dan Bitbucket, menurut konvensi, Anda harus menyimpan file spesifikasi build yang dinamai `buildspec.yml` di root (tingkat atas) setiap repositori atau menyertakan deklarasi spesifikasi build sebagai bagian dari definisi proyek build. Jangan membuat file ZIP yang berisi kode sumber repositori dan membangun file spesifikasi.
>
> Untuk input build yang disimpan di bucket S3 saja, Anda harus membuat file ZIP yang berisi kode sumber dan, menurut konvensi, file spesifikasi build yang dinamai `buildspec.yml` di root (tingkat atas) atau menyertakan deklarasi spesifikasi build sebagai bagian dari definisi proyek build.
>
> Jika Anda ingin menggunakan nama yang berbeda untuk file spesifikasi build Anda, atau Anda ingin mereferensikan spesifikasi build di lokasi selain root, Anda dapat menentukan

penggantian spesifikasi build sebagai bagian dari definisi proyek build. Untuk informasi selengkapnya, lihat Nama file buildspec dan lokasi penyimpanan.

Langkah 5: Buat proyek pembangunan

(Langkah sebelumnya: Langkah 4: Unggah kode sumber dan file buildspec)

Pada langkah ini, Anda membuat proyek build yang AWS CodeBuild digunakan untuk menjalankan build. Proyek build mencakup informasi tentang cara menjalankan build, termasuk di mana mendapatkan kode sumber, lingkungan build mana yang akan digunakan, perintah build mana yang akan dijalankan, dan tempat menyimpan output build. Lingkungan build mewakili kombinasi sistem operasi, runtime bahasa pemrograman, dan alat CodeBuild yang digunakan untuk menjalankan build. Lingkungan build diekspresikan sebagai gambar Docker. Untuk informasi selengkapnya, lihat ik htisar Docker di situs Docker Docker.

Untuk lingkungan build ini, Anda menginstruksikan CodeBuild untuk menggunakan image Docker yang berisi versi Java Development Kit (JDK) dan Apache Maven.

Untuk membuat proyek pembangunan

1. Gunakan AWS CLI untuk menjalankan create-project perintah:

```
aws codebuild create-project --generate-cli-skeleton
```

JSON-formatted data muncul di output. Salin data ke file bernama `create-project.json` di lokasi di komputer lokal atau instance tempat AWS CLI diinstal. Jika Anda memilih untuk menggunakan nama file yang berbeda, pastikan untuk menggunakannya di seluruh tutorial ini.

Ubah data yang disalin untuk mengikuti format ini, lalu simpan hasil Anda:

```
{
  "name": "codebuild-demo-project",
  "source": {
    "type": "S3",
    "location": "codebuild-region-ID-account-ID-input-bucket/MessageUtil.zip"
  },
  "artifacts": {
    "type": "S3",
    "location": "codebuild-region-ID-account-ID-output-bucket"
```

```
  },
  "environment": {
    "type": "LINUX_CONTAINER",
    "image": "aws/codebuild/standard:5.0",
    "computeType": "BUILD_GENERAL1_SMALL"
  },
  "serviceRole": "serviceIAMRole"
}
```

Ganti *serviceIAMRole* dengan Nama Sumber Daya Amazon (ARN) dari peran CodeBuild layanan (misalnya,`arn:aws:iam::`*account-ID*`:role/`*role-name*). Untuk membuatnya, lihat Memungkinkan CodeBuild untuk berinteraksi dengan orang lain AWS layanan.

Dalam data ini:

- `name`mewakili pengidentifikasi yang diperlukan untuk proyek build ini (dalam contoh ini,`codebuild-demo-project`). Nama proyek build harus unik di semua proyek build di akun Anda.
- For`source`, `type` adalah nilai wajib yang mewakili jenis repositori kode sumber (dalam contoh ini, `S3` untuk bucket Amazon S3).
- Untuk`source`, `location` mewakili jalur ke kode sumber (dalam contoh ini, nama bucket input diikuti oleh nama file ZIP).
- For`artifacts`, `type` adalah nilai wajib yang mewakili jenis repositori artefak keluaran build (dalam contoh ini, `S3` untuk bucket Amazon S3).
- For`artifacts`, `location` mewakili nama bucket keluaran yang Anda buat atau identifikasi sebelumnya (dalam contoh ini,`codebuild-`*region-ID*`-`*account-ID*`-output-bucket`).
- For`environment`, `type` adalah nilai yang diperlukan yang mewakili jenis lingkungan build (dalam contoh ini,`LINUX_CONTAINER`).
- For`environment`, `image` adalah nilai yang diperlukan yang mewakili nama gambar Docker dan kombinasi tag yang digunakan proyek build ini, seperti yang ditentukan oleh jenis repositori gambar Docker (dalam contoh ini, `aws/codebuild/standard:5.0` untuk gambar Docker di repositori gambar CodeBuild Docker). `aws/codebuild/standard`adalah nama gambar Docker. `5.0`adalah tag gambar Docker.

  Untuk menemukan lebih banyak gambar Docker yang dapat Anda gunakan dalam skenario Anda, lihat. Membangun referensi lingkungan

- Untuk`environment`, `computeType` adalah nilai yang diperlukan yang mewakili CodeBuild penggunaan sumber daya komputasi (dalam contoh ini,`BUILD_GENERAL1_SMALL`).

> **Note**
>
> Nilai lain yang tersedia dalam JSON-formatted data asli, seperti`description`,`buildspec`, `auth` (termasuk `type` dan`resource`),, `pathnamespaceType`, `name` (untuk`artifacts`),`packaging`, `environmentVariables` (termasuk `name` dan`value`),, `timeoutInMinutes`,`encryptionKey`, dan `tags` (termasuk `key` dan`value`) bersifat opsional. Mereka tidak digunakan dalam tutorial ini, jadi mereka tidak ditampilkan di sini. Untuk informasi selengkapnya, lihat Buat proyek build (AWS CLI).

2. Beralih ke direktori yang berisi file yang baru saja Anda simpan, lalu jalankan create-project perintah lagi.

```
aws codebuild create-project --cli-input-json file://create-project.json
```

Jika berhasil, data yang mirip dengan ini muncul di output.

```
{
  "project": {
    "name": "codebuild-demo-project",
    "serviceRole": "serviceIAMRole",
    "tags": [],
    "artifacts": {
      "packaging": "NONE",
      "type": "S3",
      "location": "codebuild-region-ID-account-ID-output-bucket",
      "name": "message-util.zip"
    },
    "lastModified": 1472661575.244,
    "timeoutInMinutes": 60,
    "created": 1472661575.244,
    "environment": {
      "computeType": "BUILD_GENERAL1_SMALL",
      "image": "aws/codebuild/standard:5.0",
      "type": "LINUX_CONTAINER",
      "environmentVariables": []
```

```
    },
    "source": {
      "type": "S3",
      "location": "codebuild-region-ID-account-ID-input-bucket/MessageUtil.zip"
    },
    "encryptionKey": "arn:aws:kms:region-ID:account-ID:alias/aws/s3",
    "arn": "arn:aws:codebuild:region-ID:account-ID:project/codebuild-demo-project"
  }
}
```

- `project`mewakili informasi tentang proyek pembangunan ini.
  - `tags`mewakili setiap tag yang dideklarasikan.
  - `packaging`mewakili bagaimana artefak keluaran build disimpan dalam bucket keluaran. `NONE`berarti bahwa folder dibuat di bucket keluaran. Artefak keluaran build disimpan di folder itu.
  - `lastModified`mewakili waktu, dalam format waktu Unix, ketika informasi tentang proyek build terakhir diubah.
  - `timeoutInMinutes`mewakili jumlah menit setelah CodeBuild itu menghentikan build jika build belum selesai. (Defaultnya adalah 60 menit.)
  - `created`mewakili waktu, dalam format waktu Unix, ketika proyek build dibuat.
  - `environmentVariables`mewakili variabel lingkungan apa pun yang dideklarasikan dan tersedia CodeBuild untuk digunakan selama pembuatan.
  - `encryptionKey`mewakili ARN dari kunci yang dikelola pelanggan yang CodeBuild digunakan untuk mengenkripsi artefak keluaran build.
  - `arn`mewakili ARN dari proyek pembangunan.

> **Note**
>
> Setelah Anda menjalankan create-project perintah, pesan kesalahan yang mirip dengan berikut mungkin keluar: Pengguna: tidak ***user-ARN*** berwenang untuk melakukan: codebuild: CreateProject. Ini kemungkinan besar karena Anda mengonfigurasi AWS CLI dengan kredenSIAL pengguna yang tidak memiliki izin yang cukup untuk digunakan CodeBuild untuk membuat proyek build. Untuk memperbaikinya, konfigurasikan AWS CLI dengan kredenSIAL milik salah satu entitas IAM berikut:

- Pengguna administrator di AWS akun Anda. Untuk informasi selengkapnya, lihat Membuat pengguna dan grup Akun AWS root pertama Anda di Panduan pengguna.
- Pengguna di AWS akun Anda dengan kebijakan`AWSCodeBuildAdminAccess`,`AmazonS3ReadOnlyAccess`, dan ter `IAMFullAccess` kelola yang dilampirkan ke pengguna tersebut atau grup IAM tempat pengguna tersebut berada. Jika Anda tidak memiliki pengguna atau grup di AWS akun Anda dengan izin ini, dan Anda tidak dapat menambahkan izin ini ke pengguna atau grup Anda, hubungi administrator AWS akun Anda untuk mendapatkan bantuan. Untuk informasi selengkapnya, lihat AWS kebijakan terkelola (standar) untuk AWS CodeBuild.

Langkah 6: Jalankan build

(Langkah sebelumnya: Langkah 5: Buat proyek pembangunan)

Pada langkah ini, Anda menginstruksikan AWS CodeBuild untuk menjalankan build dengan pengaturan dalam proyek build.

Untuk menjalankan build

1. Gunakan AWS CLI untuk menjalankan start-build perintah:

   ```
   aws codebuild start-build --project-name project-name
   ```

   Ganti *project-name* dengan nama proyek build Anda dari langkah sebelumnya (misalnya,`codebuild-demo-project`).

2. Jika berhasil, data yang mirip dengan berikut ini muncul di output:

   ```
   {
     "build": {
       "buildComplete": false,
       "initiator": "user-name",
       "artifacts": {
         "location": "arn:aws:s3:::codebuild-region-ID-account-ID-output-bucket/
   message-util.zip"
       },
       "projectName": "codebuild-demo-project",
       "timeoutInMinutes": 60,
   ```

```
    "buildStatus": "IN_PROGRESS",
    "environment": {
      "computeType": "BUILD_GENERAL1_SMALL",
      "image": "aws/codebuild/standard:5.0",
      "type": "LINUX_CONTAINER",
      "environmentVariables": []
    },
    "source": {
      "type": "S3",
      "location": "codebuild-region-ID-account-ID-input-bucket/MessageUtil.zip"
    },
    "currentPhase": "SUBMITTED",
    "startTime": 1472848787.882,
    "id": "codebuild-demo-project:0cfbb6ec-3db9-4e8c-992b-1ab28EXAMPLE",
    "arn": "arn:aws:codebuild:region-ID:account-ID:build/codebuild-demo-
project:0cfbb6ec-3db9-4e8c-992b-1ab28EXAMPLE"
  }
}
```

- `build`mewakili informasi tentang build ini.
  - `buildComplete`mewakili apakah build telah selesai (`true`). Atau, `false`.
  - `initiator`mewakili entitas yang memulai pembangunan.
  - `artifacts`mewakili informasi tentang output build, termasuk lokasinya.
  - `projectName`mewakili nama proyek pembangunan.
  - `buildStatus`mewakili status build saat ini saat start-build perintah dijalankan.
  - `currentPhase`mewakili fase build saat ini ketika start-build perintah dijalankan.
  - `startTime`mewakili waktu, dalam format waktu Unix, ketika proses pembuatan dimulai.
  - `id`mewakili ID build.
  - `arn`mewakili ARN bangunan.

Catat nil `id` ainya. Anda membutuhkannya di langkah berikutnya.

Langkah 7: Lihat informasi build yang diringkas

(Langkah sebelumnya: Langkah 6: Jalankan build)

Pada langkah ini, Anda melihat informasi ringkasan tentang status build Anda.

Untuk melihat informasi build yang diringkas

- Gunakan AWS CLI untuk menjalankan batch-get-builds perintah.

```
aws codebuild batch-get-builds --ids id
```

Ganti *id* dengan `id` nilai yang muncul di output langkah sebelumnya.

Jika berhasil, data yang mirip dengan ini muncul di output.

```
{
  "buildsNotFound": [],
  "builds": [
    {
      "buildComplete": true,
      "phases": [
        {
          "phaseStatus": "SUCCEEDED",
          "endTime": 1472848788.525,
          "phaseType": "SUBMITTED",
          "durationInSeconds": 0,
          "startTime": 1472848787.882
        },
        ... The full list of build phases has been omitted for brevity ...
        {
          "phaseType": "COMPLETED",
          "startTime": 1472848878.079
        }
      ],
      "logs": {
        "groupName": "/aws/codebuild/codebuild-demo-project",
        "deepLink": "https://console.aws.amazon.com/cloudwatch/home?region=region-
ID#logEvent:group=/aws/codebuild/codebuild-demo-project;stream=38ca1c4a-e9ca-4dbc-
bef1-d52bfEXAMPLE",
        "streamName": "38ca1c4a-e9ca-4dbc-bef1-d52bfEXAMPLE"
      },
      "artifacts": {
        "md5sum": "MD5-hash",
        "location": "arn:aws:s3:::codebuild-region-ID-account-ID-output-bucket/
message-util.zip",
        "sha256sum": "SHA-256-hash"
      },
      "projectName": "codebuild-demo-project",
```

```
      "timeoutInMinutes": 60,
      "initiator": "user-name",
      "buildStatus": "SUCCEEDED",
      "environment": {
        "computeType": "BUILD_GENERAL1_SMALL",
        "image": "aws/codebuild/standard:5.0",
        "type": "LINUX_CONTAINER",
        "environmentVariables": []
      },
      "source": {
        "type": "S3",
        "location": "codebuild-region-ID-account-ID-input-bucket/MessageUtil.zip"
      },
      "currentPhase": "COMPLETED",
      "startTime": 1472848787.882,
      "endTime": 1472848878.079,
      "id": "codebuild-demo-project:38ca1c4a-e9ca-4dbc-bef1-d52bfEXAMPLE",
      "arn": "arn:aws:codebuild:region-ID:account-ID:build/codebuild-demo-
project:38ca1c4a-e9ca-4dbc-bef1-d52bfEXAMPLE"
    }
  ]
}
```

- `buildsNotFound`mewakili ID build untuk setiap build di mana informasi tidak tersedia. Dalam contoh ini, itu harus kosong.
- `builds`mewakili informasi tentang setiap build di mana informasi tersedia. Dalam contoh ini, informasi tentang hanya satu build muncul di output.
  - `phases`mewakili serangkaian fase build yang CodeBuild berjalan selama proses build. Informasi tentang setiap fase build terdaftar secara terpisah sebagai`startTime`,`endTime`, dan `durationInSeconds` (ketika fase build dimulai dan berakhir, dinyatakan dalam format waktu Unix, dan berapa lama berlangsung, dalam detik), dan `phaseType` seperti (`SUBMITTED`,`PROVISIONING`,`DOWNLOAD_SOURCE`,`INSTALL`,`PRE_BUILD`,`BUILD`,`POST_BUILD`,`UPL` atau`COMPLETED`) dan `phaseStatus` (sepertiSUCCEEDED,,`FAILED`,`FAULT`, `TIMED_OUTIN_PROGRESS`, atau`STOPPED`). Pertama kali Anda menjalankan batch-get-builds perintah, mungkin tidak ada banyak (atau apa pun) fase. Setelah menjalankan batch-get-builds perintah berikutnya dengan ID build yang sama, lebih banyak fase build akan muncul di output.
  - `logs`mewakili informasi di Amazon CloudWatch Logs tentang log build.

- `md5sum`dan `sha256sum` mewakili MD5 dan SHA-256 hash dari artefak keluaran build. Ini muncul dalam output hanya jika `packaging` nilai proyek build disetel ke`ZIP`. (Anda tidak menetapkan nilai ini dalam tutorial ini.) Anda dapat menggunakan hash ini bersama dengan alat checksum untuk mengonfirmasi integritas dan keaslian file.

  > **Note**
  >
  > Anda juga dapat menggunakan konsol Amazon S3 untuk melihat hash ini. Pilih kotak di sebelah artefak keluaran build, pilih Ac tions, lalu pilih Proper ties. Di panel Properties, perluas Metadata, dan lihat nilai untuk x-amz-meta-codebuild-content-md5 dan x-amz-meta- codebuild-content-sha256. (Di konsol Amazon S3, nilai ETag artefak keluaran build tidak boleh ditafsirkan sebagai MD5 atau SHA-256 hash.)
  >
  > Jika Anda menggunakan AWS SDK untuk mendapatkan hash ini, nilainya diberi nama dan. `codebuild-content-md5` `codebuild-content-sha256`

- `endTime`memewakili waktu, dalam format waktu Unix, ketika proses pembuatan berakhir.

> **Note**
>
> Metadata Amazon S3 memiliki CodeBuild header bernama `x-amz-meta-codebuild-buildarn` yang berisi `buildArn` CodeBuild build yang menerbitkan artefak ke Amazon S3. Dit `buildArn` ambahkan untuk memungkinkan pelacakan sumber untuk pemberitahuan dan referensi dari mana bangunan artefak dihasilkan.

Langkah 8: Lihat informasi build terperinci

(Langkah sebelumnya: Langkah 7: Lihat informasi build yang diringkas)

Pada langkah ini, Anda melihat informasi terperinci tentang build Anda di CloudWatch Log.

> **Note**
>
> Untuk melindungi informasi sensitif, berikut ini disembunyikan di CodeBuild log:
>
> - AWS ID kunci akses. Untuk informasi selengkapnya, lihat Mengelola Kunci Akses untuk Pengguna IAM di Panduan AWS Identity and Access Management Pengguna.

- String ditentukan menggunakan Parameter Store. Untuk informasi selengkapnya, lihat Panduan Konsol Penyimpanan Parameter Manajer Sistem dan Panduan Konsol Penyimpanan Parameter Manajer Sistem di Panduan Pengguna Amazon EC2 Systems Manager.
- String ditentukan menggunakan AWS Secrets Manager. Untuk informasi selengkapnya, lihat Manajemen kunci.

Untuk melihat informasi build terperinci

1. Gunakan browser web Anda untuk pergi ke `deepLink` lokasi yang muncul di output pada langkah sebelumnya (misalnya,`https://console.aws.amazon.com/cloudwatch/home?region=`*`region-ID`*`#logEvent:group=/aws/codebuild/codebuild-demo-project;stream=38ca1c4a-e9ca-4dbc-bef1-d52bfEXAMPLE`).
2. Di aliran CloudWatch log log, Anda dapat menelusuri peristiwa log. Secara default, hanya set terakhir dari peristiwa log yang ditampilkan. Untuk melihat peristiwa log sebelumnya, gulir ke awal daftar.
3. Dalam tutorial ini, sebagian besar peristiwa log berisi informasi verbose tentang CodeBuild mengunduh dan menginstal file ketergantungan build ke lingkungan build-nya, yang mungkin tidak Anda pedulikan. Anda dapat menggunakan kotak Filter peristiwa untuk mengurangi informasi yang ditampilkan. Misalnya, jika Anda memasukkan `"[INFO]"` peristiwa Filter, hanya peristiwa yang berisi yang `[INFO]` ditampilkan. Untuk informasi selengkapnya, lihat Filter dan sintaks pola di Panduan CloudWatch Pengguna Amazon.

Bagian-bagian dari aliran CloudWatch log log ini berkaitan dengan tutorial ini.

```
...
[Container] 2016/04/15 17:49:42 Entering phase PRE_BUILD
[Container] 2016/04/15 17:49:42 Running command echo Entering pre_build phase...
[Container] 2016/04/15 17:49:42 Entering pre_build phase...
[Container] 2016/04/15 17:49:42 Phase complete: PRE_BUILD Success: true
[Container] 2016/04/15 17:49:42 Entering phase BUILD
[Container] 2016/04/15 17:49:42 Running command echo Entering build phase...
[Container] 2016/04/15 17:49:42 Entering build phase...
[Container] 2016/04/15 17:49:42 Running command mvn install
[Container] 2016/04/15 17:49:44 [INFO] Scanning for projects...
[Container] 2016/04/15 17:49:44 [INFO]
```

```
[Container] 2016/04/15 17:49:44 [INFO]
 ------------------------------------------------------------------------
[Container] 2016/04/15 17:49:44 [INFO] Building Message Utility Java Sample App 1.0
[Container] 2016/04/15 17:49:44 [INFO]
 ------------------------------------------------------------------------
...
[Container] 2016/04/15 17:49:55
 -------------------------------------------------------
[Container] 2016/04/15 17:49:55  T E S T S
[Container] 2016/04/15 17:49:55
 -------------------------------------------------------
[Container] 2016/04/15 17:49:55 Running TestMessageUtil
[Container] 2016/04/15 17:49:55 Inside testSalutationMessage()
[Container] 2016/04/15 17:49:55 Hi!Robert
[Container] 2016/04/15 17:49:55 Inside testPrintMessage()
[Container] 2016/04/15 17:49:55 Robert
[Container] 2016/04/15 17:49:55 Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time
 elapsed: 0.018 sec
[Container] 2016/04/15 17:49:55
[Container] 2016/04/15 17:49:55 Results :
[Container] 2016/04/15 17:49:55
[Container] 2016/04/15 17:49:55 Tests run: 2, Failures: 0, Errors: 0, Skipped: 0
...
[Container] 2016/04/15 17:49:56 [INFO]
 ------------------------------------------------------------------------
[Container] 2016/04/15 17:49:56 [INFO] BUILD SUCCESS
[Container] 2016/04/15 17:49:56 [INFO]
 ------------------------------------------------------------------------
[Container] 2016/04/15 17:49:56 [INFO] Total time: 11.845 s
[Container] 2016/04/15 17:49:56 [INFO] Finished at: 2016-04-15T17:49:56+00:00
[Container] 2016/04/15 17:49:56 [INFO] Final Memory: 18M/216M
[Container] 2016/04/15 17:49:56 [INFO]
 ------------------------------------------------------------------------
[Container] 2016/04/15 17:49:56 Phase complete: BUILD Success: true
[Container] 2016/04/15 17:49:56 Entering phase POST_BUILD
[Container] 2016/04/15 17:49:56 Running command echo Entering post_build phase...
[Container] 2016/04/15 17:49:56 Entering post_build phase...
[Container] 2016/04/15 17:49:56 Phase complete: POST_BUILD Success: true
[Container] 2016/04/15 17:49:57 Preparing to copy artifacts
[Container] 2016/04/15 17:49:57 Assembling file list
[Container] 2016/04/15 17:49:57 Expanding target/messageUtil-1.0.jar
[Container] 2016/04/15 17:49:57 Found target/messageUtil-1.0.jar
[Container] 2016/04/15 17:49:57 Creating zip artifact
```

Dalam contoh ini, CodeBuild berhasil menyelesaikan fase pra-build, build, dan post-build. Itu menjalankan pengujian unit dan berhasil membangun `messageUtil-1.0.jar` file.

Langkah 9: Dapatkan artefak keluaran build

(Langkah sebelumnya: Langkah 8: Lihat informasi build terperinci)

Pada langkah ini, Anda mendapatkan `messageUtil-1.0.jar` file yang CodeBuild dibangun dan diunggah ke bucket keluaran.

Anda dapat menggunakan CodeBuild konsol atau konsol Amazon S3 untuk menyelesaikan langkah ini.

Untuk mendapatkan artefak keluaran build (AWS CodeBuild konsol)

1. Dengan CodeBuild konsol masih terbuka dan halaman detail build masih ditampilkan dari langkah sebelumnya, pilih tab Detail build dan gulir ke bawah ke bagian Artifacts.

   > **Note**
   >
   > Jika halaman detail build tidak ditampilkan, di bilah navigasi, pilih Ri wayat build, lalu pilih tautan Build run.

2. Tautan ke folder Amazon S3 berada di bawah lokasi ungg ahan Artifacts. Tautan ini membuka folder di Amazon S3 tempat Anda menemukan file artefak keluaran `messageUtil-1.0.jar` build.

Untuk mendapatkan artefak keluaran build (konsol Amazon S3)

1. Buka konsol Amazon S3 di https://console.aws.amazon.com/s3/.
2. Buka `codebuild-`*`region-ID`*`-`*`account-ID`*`-output-bucket`.
3. Buka folder `codebuild-demo-project`.
4. Buka `target` folder, tempat Anda menemukan file artefak keluaran `messageUtil-1.0.jar` build.

Langkah 10: Hapus bucket S3

(Langkah sebelumnya: Langkah 9: Dapatkan artefak keluaran build)

Untuk mencegah tagihan berkelanjutan ke AWS akun Anda, Anda dapat menghapus bucket input dan output yang digunakan dalam tutorial ini. Untuk petunjuk, lihat Menghapus atau Mengosongkan Bucket di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.

Jika Anda menggunakan pengguna IAM atau pengguna IAM administrator untuk menghapus bucket ini, pengguna harus memiliki lebih banyak izin akses. Tambahkan pernyataan berikut antara penanda (*### BEGIN ADDING STATEMENT HERE ###*dan*### END ADDING STATEMENTS HERE ###*) ke kebijakan akses yang ada untuk pengguna.

Elips (...) dalam pernyataan ini digunakan untuk singkatnya. Jangan menghapus pernyataan apa pun dalam kebijakan akses yang ada. Jangan masukkan elips ini ke dalam kebijakan.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:DeleteBucket",
        "s3:DeleteObject"
      ],
      "Resource": "*"
    }
  ]
}
```

Membungkus

Dalam tutorial ini, Anda AWS CodeBuild biasa membangun satu set file kelas Java ke dalam file JAR. Anda kemudian melihat hasil build.

Anda sekarang dapat mencoba menggunakan CodeBuild dalam skenario Anda sendiri. Ikuti petunjuk dalam Merencanakan build. Jika Anda belum merasa siap, Anda mungkin ingin mencoba membangun beberapa sampel. Lihat informasi yang lebih lengkap di Gunakan sampel berbasis kasus untuk CodeBuild.

Gunakan sampel berbasis kasus untuk CodeBuild

Anda dapat menggunakan sampel berbasis kasus penggunaan ini untuk bereksperimen dengan: AWS CodeBuild

Cross-service sampel

Daftar sampel lintas layanan untuk bereksperimen AWS CodeBuild.

Bangun contoh lencana

Menunjukkan cara mengatur CodeBuild dengan lencana build.

Contoh laporan pengujian

Menggunakan AWS CLI untuk membuat, menjalankan, dan melihat hasil laporan pengujian.

Sampel Docker untuk CodeBuild

Menunjukkan cara menggunakan gambar Docker khusus, mempublikasikan gambar Docker ke repositori di Amazon ECR, dan menggunakan gambar Docker di registri pribadi.

Host build output dalam bucket S3

Menunjukkan cara membuat situs web statis di bucket S3 menggunakan artefak build yang tidak terenkripsi.

Beberapa sampel input dan output

Menunjukkan cara menggunakan beberapa sumber input dan beberapa artefak keluaran dalam proyek build.

Sampel eksekusi uji paralel

Menunjukkan cara menggunakan perintah `codebuild-tests-run` CLI untuk membagi dan menjalankan tes di seluruh lingkungan eksekusi paralel.

Versi runtime dalam contoh file buildspec

Menunjukkan cara menentukan runtime dan versinya di file buildspec.

Contoh versi sumber

Menunjukkan cara menggunakan versi tertentu dari sumber Anda dalam proyek CodeBuild build.

Third-party sampel repositori sumber untuk CodeBuild

Menunjukkan cara membuat BitBucket, Server GitHub Perusahaan, dan GitHub menarik permintaan dengan webhook menggunakan. CodeBuild

Tetapkan nama artefak pada waktu pembuatan menggunakan versi semantik

Menunjukkan cara menggunakan versi semantik untuk membuat nama artefak pada waktu pembuatan.

Cross-service sampel untuk CodeBuild

Anda dapat menggunakan sampel lintas layanan ini untuk bereksperimen dengan: AWS CodeBuild

Sampel Amazon ECR

Menggunakan gambar Docker di repositori Amazon ECR untuk menggunakan Apache Maven untuk menghasilkan satu file JAR. Instruksi sampel akan menunjukkan kepada Anda cara membuat dan mendorong gambar Docker ke Amazon ECR, membuat proyek Go, membangun proyek, menjalankan proyek, dan mengatur izin untuk memungkinkan terhubung CodeBuild ke Amazon ECR.

Sampel Amazon EFS

Menunjukkan cara mengonfigurasi file buildspec sehingga CodeBuild proyek dipasang dan dibangun di atas sistem file Amazon EFS. Instruksi sampel akan menunjukkan kepada Anda cara membuat Amazon VPC, membuat sistem file di Amazon VPC, membuat dan membangun proyek yang menggunakan Amazon VPC, dan kemudian meninjau file proyek dan variabel yang dihasilkan.

AWS CodePipeline sampel

Menunjukkan cara menggunakan AWS CodePipeline untuk membuat build dengan batch build serta beberapa sumber input dan beberapa artefak keluaran. Termasuk dalam bagian ini adalah contoh file JSON yang menunjukkan struktur pipeline yang membuat batch build dengan artefak terpisah, dan artefak gabungan. Sampel JSON tambahan disediakan yang menunjukkan struktur pipeline dengan beberapa sumber input dan beberapa artefak keluaran.

AWS Config sampel

Menunjukkan cara mengatur AWS Config. Daftar CodeBuild sumber daya mana yang dilacak dan menjelaskan cara mencari CodeBuild proyek di dalamnya AWS Config. Instruksi sampel akan

menunjukkan kepada Anda prasyarat untuk mengintegrasikan AWS Config, langkah-langkah untuk mengatur AWS Config, dan langkah-langkah untuk mencari CodeBuild proyek dan data di dalamnya AWS Config.

Buat contoh notifikasi

Menggunakan Apache Maven untuk menghasilkan satu file JAR. Mengirim pemberitahuan build ke pelanggan topik Amazon SNS. Petunjuk contoh menunjukkan kepada Anda cara mengatur izin sehingga CodeBuild dapat berkomunikasi dengan Amazon SNS dan CloudWatch, cara membuat dan mengidentifikasi CodeBuild topik di Amazon SNS, cara berlangganan penerima ke topik, dan cara mengatur aturan di. CloudWatch

Sampel Amazon ECR untuk CodeBuild

Contoh ini menggunakan image Docker di repositori image Amazon Elastic Container Registry (Amazon ECR) Registry ECR) untuk membuat proyek Go sampel.

> **Important**
>
> Menjalankan sampel ini dapat mengakibatkan biaya ke AWS akun Anda. Ini termasuk kemungkinan biaya untuk AWS CodeBuild dan untuk AWS sumber daya dan tindakan yang terkait dengan Amazon S3,, CloudWatch Log AWS KMS, dan Amazon ECR. Untuk informasi selengkapnya, lihat CodeBuild harga, harga Amazon S3, harga, AWS Key Management Service harga Amazon, dan CloudWatch harga Amazon Elastic Container Registry.

Topik

- Jalankan sampel Amazon ECR

Jalankan sampel Amazon ECR

Gunakan petunjuk berikut untuk menjalankan sampel Amazon ECR. CodeBuild

Untuk menjalankan sampel ini

1. Untuk membuat dan mendorong gambar Docker ke repositori gambar Anda di Amazon ECR, selesaikan langkah-langkah di bagianJalankan contoh 'Publikasikan gambar Docker ke Amazon ECR'. Contoh 'Publikasikan gambar Docker ke Amazon ECR'

2. Buat proyek Go:

   a. Buat file seperti yang dijelaskan di Pergi file proyek bagian Struktur proyek Go dan topik ini, lalu unggah ke bucket input S3 atau repositori AWS CodeCommit GitHub, atau Bitbucket.

   > **Important**
   >
   > Jangan mengunggah*(root directory name)*, hanya file di dalamnya*(root directory name)*.
   > Jika Anda menggunakan bucket input S3, pastikan untuk membuat file ZIP yang berisi file, lalu unggah ke bucket input. Jangan tambahkan *(root directory name)* ke file ZIP, hanya file di dalamnya*(root directory name)*.

   b. Buat proyek build, jalankan build, dan lihat informasi build terkait.

   Jika Anda menggunakan AWS CLI untuk membuat proyek build, JSON-formatted input ke `create-project` perintah mungkin terlihat mirip dengan ini. (Ganti placeholder dengan nilai Anda sendiri.)

```
{
  "name": "sample-go-project",
  "source": {
    "type": "S3",
    "location": "codebuild-region-ID-account-ID-input-bucket/GoSample.zip"
  },
  "artifacts": {
    "type": "S3",
    "location": "codebuild-region-ID-account-ID-output-bucket",
    "packaging": "ZIP",
    "name": "GoOutputArtifact.zip"
  },
  "environment": {
    "type": "LINUX_CONTAINER",
    "image": "aws/codebuild/standard:5.0",
    "computeType": "BUILD_GENERAL1_SMALL"
  },
  "serviceRole": "arn:aws:iam::account-ID:role/role-name",
  "encryptionKey": "arn:aws:kms:region-ID:account-ID:key/key-ID"
}
```

   c. Untuk mendapatkan artefak keluaran build, buka bucket keluaran S3 Anda.

d. Unduh *GoOutputArtifact*.zip file ke komputer atau instance lokal Anda, lalu ekstrak konten file tersebut. Dalam konten yang diekstraksi, dapatkan hello file.

3. Jika salah satu dari berikut ini benar, Anda harus menambahkan izin ke repositori gambar Anda di Amazon ECR sehingga AWS CodeBuild dapat menarik image Docker-nya ke lingkungan build.

   - Proyek Anda menggunakan CodeBuild kredensi untuk menarik gambar Amazon ECR. Ini dilambangkan dengan nilai CODEBUILD dalam imagePullCredentialsType atribut Anda. ProjectEnvironment
   - Proyek Anda menggunakan gambar ECR Amazon lintas akun. Dalam hal ini, proyek Anda harus menggunakan peran layanannya untuk menarik gambar Amazon ECR. Untuk mengaktifkan perilaku ini, setel imagePullCredentialsType atribut Anda ProjectEnvironment keSERVICE_ROLE.

   1. Buka konsol Amazon ECR di https://console.aws.amazon.com/ecr/.
   2. Dalam daftar nama repositori, pilih nama repositori yang Anda buat atau pilih.
   3. Dari panel navigasi, pilih Izin, pilih Edit, lalu pilih Tambah pernyataan.
   4. Untuk nama Pernyataan, masukkan pengenal (misalnya,**CodeBuildAccess**).
   5. Untuk Efek, biarkan Izinkan dipilih. Ini menunjukkan bahwa Anda ingin mengizinkan akses ke AWS akun lain.
   6. Untuk Principal, lakukan salah satu hal berikut:
      - Jika project Anda menggunakan CodeBuild kredensi untuk menarik image Amazon ECR, di Prinsipal layanan, masukkan. **codebuild.amazonaws.com**
      - Jika project Anda menggunakan image Amazon ECR lintas akun, untuk ID AWS akun, masukkan ID AWS akun yang ingin Anda akses.
   7. Lewati daftar Semua entitas IAM.
   8. Untuk Tindakan, pilih tindakan khusus tarik: ecr:, ecr: GetDownloadUrlForLayer, dan ecr: BatchGetImage. BatchCheckLayerAvailability
   9. Untuk Ketentuan, tambahkan yang berikut ini:

```
{
   "StringEquals":{
      "aws:SourceAccount":"<AWS-account-ID>",
      "aws:SourceArn":"arn:aws:codebuild:<region>:<AWS-account-
ID>:project/<project-name>"
   }
```

```
}
```

10. Pilih Simpan.

Kebijakan ini ditampilkan di Izin. Kepala sekolah adalah apa yang Anda masukkan untuk Principal pada langkah 3 dari prosedur ini:

- Jika project Anda menggunakan CodeBuild kredensi untuk menarik image Amazon ECR, **"codebuild.amazonaws.com"** muncul di bawah Prinsipal layanan.
- Jika project Anda menggunakan image Amazon ECR lintas akun, ID AWS akun yang ingin Anda berikan akses akan muncul di bawah ID AWS Akun.

  Kebijakan contoh berikut menggunakan CodeBuild kredensi dan gambar ECR Amazon lintas akun.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "CodeBuildAccessPrincipal",
            "Effect": "Allow",
            "Action": [
                "ecr:GetDownloadUrlForLayer",
                "ecr:BatchGetImage",
                "ecr:BatchCheckLayerAvailability"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:SourceArn": "arn:aws:codebuild:us-
east-1:111122223333:project/MyProject",
                    "aws:SourceAccount": "111122223333"
                }
            }
        },
        {
            "Sid": "CodeBuildAccessCrossAccount",
            "Effect": "Allow",
            "Action": [
                "ecr:GetDownloadUrlForLayer",
                "ecr:BatchGetImage",
```

```
                "ecr:BatchCheckLayerAvailability"
            ],
            "Resource": "*"
        }
    ]
}
```

- Jika proyek Anda menggunakan CodeBuild kredensi dan Anda ingin CodeBuild proyek Anda memiliki akses terbuka ke repositori Amazon ECR, Anda dapat menghilangkan `Condition` kunci dan menambahkan kebijakan sampel berikut.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "CodeBuildAccessPrincipal",
            "Effect": "Allow",
            "Resource": [
                "arn:aws:codecommit:us-
east-2:111122223333:MySharedDemoRepo"
            ],
            "Action": [
                "ecr:GetDownloadUrlForLayer",
                "ecr:BatchGetImage",
                "ecr:BatchCheckLayerAvailability"
            ]
        },
        {
            "Sid": "CodeBuildAccessCrossAccount",
            "Effect": "Allow",
            "Resource": [
                "arn:aws:codecommit:us-
east-2:111122223333:MySharedDemoRepo"
            ],
            "Action": [
                "ecr:GetDownloadUrlForLayer",
                "ecr:BatchGetImage",
                "ecr:BatchCheckLayerAvailability"
            ]
        }
    ]
```

```
}
```

4. Buat proyek build, jalankan build, dan lihat informasi build.

   Jika Anda menggunakan AWS CLI untuk membuat proyek build, JSON-formatted input ke `create-project` perintah mungkin terlihat mirip dengan ini. (Ganti placeholder dengan nilai Anda sendiri.)

```
{
  "name": "amazon-ecr-sample-project",
  "source": {
    "type": "S3",
    "location": "codebuild-region-ID-account-ID-input-bucket/GoSample.zip"
  },
  "artifacts": {
    "type": "S3",
    "location": "codebuild-region-ID-account-ID-output-bucket",
    "packaging": "ZIP",
    "name": "GoOutputArtifact.zip"
  },
  "environment": {
    "type": "LINUX_CONTAINER",
    "image": "account-ID.dkr.ecr.region-ID.amazonaws.com/your-Amazon-ECR-repo-
name:tag",
    "computeType": "BUILD_GENERAL1_SMALL"
  },
  "serviceRole": "arn:aws:iam::account-ID:role/role-name",
  "encryptionKey": "arn:aws:kms:region-ID:account-ID:key/key-ID"
}
```

5. Untuk mendapatkan artefak keluaran build, buka bucket keluaran S3 Anda.
6. Unduh *GoOutputArtifact*`.zip` file ke komputer atau instance lokal Anda, lalu ekstrak konten *GoOutputArtifact*`.zip` file tersebut. Dalam konten yang diekstraksi, dapatkan `hello` file.

Struktur proyek Go

Sampel ini mengasumsikan struktur direktori ini.

```
(root directory name)
### buildspec.yml
### hello.go
```

Pergi file proyek

Sampel ini menggunakan file-file ini.

`buildspec.yml`(dalam*(root directory name)*)

```
version: 0.2

phases:
  install:
   runtime-versions:
     golang: 1.13
  build:
    commands:
      - echo Build started on `date`
      - echo Compiling the Go code
      - go build hello.go
  post_build:
    commands:
      - echo Build completed on `date`
artifacts:
  files:
    - hello
```

`hello.go`(dalam*(root directory name)*)

```
package main
import "fmt"

func main() {
  fmt.Println("hello world")
  fmt.Println("1+1 =", 1+1)
  fmt.Println("7.0/3.0 =", 7.0/3.0)
  fmt.Println(true && false)
  fmt.Println(true || false)
  fmt.Println(!true)
}
```

Contoh Sistem File Amazon Elastic untuk AWS CodeBuild

Anda mungkin ingin membuat AWS CodeBuild build di Amazon Elastic File System, layanan file bersama yang dapat diskalakan untuk instans Amazon EC2. Kapasitas penyimpanan dengan

Amazon EFS elastis, sehingga tumbuh atau menyusut saat file ditambahkan dan dihapus. Ini memiliki antarmuka layanan web sederhana yang dapat Anda gunakan untuk membuat dan mengkonfigurasi sistem file. Ini juga mengelola semua infrastruktur penyimpanan file untuk Anda, sehingga Anda tidak perlu khawatir tentang penerapan, menambal, atau memelihara konfigurasi sistem file. Untuk informasi selengkapnya, lihat Apa itu Amazon Elastic File System? dalam Panduan Pengguna Sistem File Amazon Elastic.

Contoh ini menunjukkan cara mengkonfigurasi CodeBuild proyek sehingga dipasang dan kemudian membangun aplikasi Java ke sistem file Amazon EFS. Sebelum memulai, Anda harus memiliki aplikasi Java yang siap untuk dibangun yang diunggah ke bucket input S3 atau AWS CodeCommit,, GitHub Enterprise Server GitHub, atau repositori Bitbucket.

Data yang sedang transit untuk sistem file Anda dienkripsi. Untuk mengenkripsi data dalam transit menggunakan gambar yang berbeda, lihat Men genkripsi data dalam transit.

Topik

- Gunakan AWS CodeBuild dengan Amazon Elastic File System
- Memecahkan masalah integrasi Amazon EFS

Gunakan AWS CodeBuild dengan Amazon Elastic File System

Sampel mencakup empat langkah tingkat tinggi yang diperlukan untuk menggunakan Amazon EFS dengan AWS CodeBuild. File tersebut adalah:

1. Buat cloud pribadi virtual (VPC) di AWS akun Anda.
2. Buat sistem file yang menggunakan VPC ini.
3. Membuat dan membangun CodeBuild proyek yang menggunakan VPC. CodeBuild Proyek ini menggunakan yang berikut untuk mengidentifikasi sistem file:
   - Pengidentifikasi sistem file yang unik. Anda memilih pengidentifikasi ketika Anda menentukan sistem file dalam proyek build Anda.
   - ID sistem file. ID ditampilkan saat Anda melihat sistem file Anda di konsol Amazon EFS.
   - Titik pemasangan. Ini adalah direktori di wadah Docker Anda yang memasang sistem file.
   - Opsi pemasangan. Ini termasuk detail tentang cara memasang sistem file.
4. Tinjau proyek build untuk memastikan bahwa file dan variabel proyek yang benar telah dihasilkan.

> **Note**
>
> Sistem file yang dibuat di Amazon EFS hanya didukung pada platform Linux.

Topik

- Langkah 1: Buat VPC menggunakan CloudFormation
- Langkah 2: Buat sistem file Amazon Elastic File System dengan VPC Anda
- Langkah 3: Buat CodeBuild proyek untuk digunakan dengan Amazon EFS
- Langkah 4: Tinjau proyek pembangunan

Langkah 1: Buat VPC menggunakan CloudFormation

Buat VPC Anda dengan CloudFormation template.

1. Ikuti petunjuk CloudFormation Template VPC untuk digunakan CloudFormation untuk membuat VPC.

   > **Note**
   >
   > VPC yang dibuat oleh CloudFormation template ini memiliki dua subnet pribadi dan dua subnet publik. Anda hanya boleh menggunakan subnet pribadi saat Anda menggunakannya AWS CodeBuild untuk memasang sistem file yang Anda buat di Amazon EFS. Jika Anda menggunakan salah satu subnet publik, build gagal.

2. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon VPC di https://console.aws.amazon.com/vpc/.
3. Pilih VPC yang Anda buat. CloudFormation
4. Pada tab Deskripsi, catat nama VPC Anda dan ID-nya. Keduanya diperlukan saat Anda membuat AWS CodeBuild proyek Anda nanti dalam sampel ini.

Langkah 2: Buat sistem file Amazon Elastic File System dengan VPC Anda

Buat sistem file Amazon EFS sederhana untuk sampel ini menggunakan VPC yang Anda buat sebelumnya.

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon EFS di https://console.aws.amazon.com/efs/.
2. Pilih Buat sistem file.
3. Dari VPC, pilih nama VPC yang Anda catat sebelumnya dalam sampel ini.
4. Biarkan Zona Ketersediaan yang terkait dengan subnet Anda dipilih.
5. Pilih Langkah Selanjutnya.
6. Di Tambah tag, untuk kunci Nama default, di Nilai, masukkan nama sistem file Amazon EFS Anda.
7. Tet ap pilih Bursting dan Tujuan Umum sebagai mode kinerja dan throughput default Anda, lalu pilih Langkah Ber ikutnya.
8. Untuk Konfigurasikan akses klien, pilih Langkah Berikutnya.
9. Pilih Buat Sistem File.
10. (Opsional) Sebaiknya tambahkan kebijakan ke sistem file Amazon EFS Anda yang memberlakukan enkripsi data yang sedang transit. Di konsol Amazon EFS, pilih Kebijakan sistem file, pilih Edit, pilih kotak berlabel Ber laku enkripsi dalam transit untuk semua klien, lalu pilih S impan.

Langkah 3: Buat CodeBuild proyek untuk digunakan dengan Amazon EFS

Buat AWS CodeBuild proyek yang menggunakan VPC yang Anda buat sebelumnya dalam sampel ini. Saat build dijalankan, itu memasang sistem file Amazon EFS yang dibuat sebelumnya. Selanjutnya, ia menyimpan file.jar yang dibuat oleh aplikasi Java Anda di direktori titik pemasangan sistem file Anda.

1. Buka AWS CodeBuild konsol di https://console.aws.amazon.com/codesuite/codebuild/home.
2. Dari panel navigasi, pilih Bangun proyek, lalu pilih Buat proyek build.
3. Dalam Nama proyek, masukkan nama untuk proyek Anda.
4. Dari penyedia Sumber, pilih repositori yang berisi aplikasi Java yang ingin Anda buat.
5. Masukkan informasi, seperti URL repositori, yang CodeBuild digunakan untuk menemukan aplikasi Anda. Pilihannya berbeda untuk setiap penyedia sumber. Untuk informasi selengkapnya, lihat Choose source provider.
6. Dari gambar Lingkungan, pilih Gambar terkelola.
7. Dari Sistem operasi, pilih Amazon Linux 2.
8. Dari Runtime (s), pilih Standar.

9. Dari Gambar, pilih aws/codebuild /amaz onlinux-x86_64-standard:4.0.
10. Dari tipe Lingkungan, pilih Linux.
11. Di bawah Peran layanan, pilih Peran layanan baru. Di Nama peran, masukkan nama untuk peran yang CodeBuild dibuat untuk Anda.
12. Perluas Additional configuration (Konfigurasi tambahan).
13. Pilih Aktifkan tanda ini jika Anda ingin membuat gambar Docker atau ingin build Anda mendapatkan hak istimewa yang ditingkatkan.

> **Note**
>
> Secara default, daemon Docker diaktifkan untuk build non-VPC. Jika Anda ingin menggunakan wadah Docker untuk build VPC, lihat Runtime Privilege dan Kapabilitas Linux di situs Docker Docker dan aktifkan mode istimewa. Selain itu, Windows tidak mendukung mode istimewa.

14. Dari VPC, pilih ID VPC.
15. Dari Subnets, pilih satu atau beberapa subnet pribadi yang terkait dengan VPC Anda. Anda harus menggunakan subnet pribadi dalam build yang memasang sistem file Amazon EFS. Jika Anda menggunakan subnet publik, build gagal.
16. Dari Grup Keamanan, pilih grup keamanan default.
17. Dalam Sistem file, masukkan informasi berikut:

    - Untuk Identifier, masukkan pengenal sistem file yang unik. Itu harus kurang dari 129 karakter dan hanya berisi karakter alfanumerik dan garis bawah. CodeBuild menggunakan pengenal ini untuk membuat variabel lingkungan yang mengidentifikasi sistem file elastis. Format variabel lingkungan `CODEBUILD_<file_system_identifier>` dalam huruf kapital. Misalnya, jika Anda memasukkan`my_efs`, variabel lingkungan adalah`CODEBUILD_MY_EFS`.
    - Untuk ID, pilih ID sistem file.
    - (Opsional) Masukkan direktori dalam sistem file. CodeBuild memasang direktori ini. Jika Anda membiarkan jalur direktori kosong CodeBuild, pasang seluruh sistem file. Jalur ini relatif terhadap root sistem file.
    - Untuk Mount point, masukkan jalur absolut direktori di wadah build Anda tempat sistem file dipasang. Jika direktori ini tidak ada CodeBuild , buat selama build.
    - (Opsional) Masukkan opsi pemasangan. Jika Anda membiarkan opsi Mount kosong, gunakan CodeBuild opsi pemasangan defaultnya:

```
nfsvers=4.1
rsize=1048576
wsize=1048576
hard
timeo=600
retrans=2
```

Untuk informasi selengkapnya, lihat Opsi Pemasangan NFS yang Direkom endasikan di Panduan Pengguna Sistem File Amazon Elastic.

18. Untuk Spesifikasi Build, pilih Sisipkan perintah build, lalu pilih Beralih ke editor.
19. Masukkan perintah buildspec berikut ke dalam editor. Ganti *<file_system_identifier>* dengan pengenal yang Anda masukkan pada langkah 17. Gunakan huruf kapital (misalnya,CODEBUILD_MY_EFS).

```
version: 0.2
phases:
  install:
    runtime-versions:
      java: corretto11
  build:
    commands:
      - mvn compile -Dgpg.skip=true -Dmaven.repo.local=
$CODEBUILD_<file_system_identifier>
```

20. Gunakan nilai default untuk semua pengaturan lainnya, lalu pilih Buat proyek build. Ketika build Anda selesai, halaman konsol untuk proyek Anda ditampilkan.
21. Pilih Mulai membangun.

Langkah 4: Tinjau proyek pembangunan

Setelah AWS CodeBuild proyek Anda dibangun:

- Anda memiliki file.jar yang dibuat oleh aplikasi Java Anda yang dibuat untuk sistem file Amazon EFS Anda di bawah direktori titik pemasangan Anda.
- Variabel lingkungan yang mengidentifikasi sistem file Anda dibuat menggunakan pengidentifikasi sistem file yang Anda masukkan saat Anda membuat proyek.

Untuk informasi selengkapnya, lihat Memasang sistem file di Panduan Pengguna Sistem File Amazon Elastic.

Memecahkan masalah integrasi Amazon EFS

Berikut ini adalah kesalahan yang mungkin Anda temui saat menyiapkan Amazon EFS CodeBuild.

Topik

- CLIENT_ERROR: pemasangan '127.0.0.1: /' gagal. izin ditolak
- CLIENT_ERROR: pemasangan '127.0.0.1: /' gagal. koneksi diatur ulang oleh rekan
- VPC_CLIENT_ERROR: Kesalahan EC2 yang tidak terduga: UnauthorizedOperation

CLIENT_ERROR: pemasangan '127.0.0.1: /' gagal. izin ditolak

Otorisasi IAM tidak didukung untuk memasang Amazon EFS dengan CodeBuild. Jika Anda menggunakan kebijakan sistem file Amazon EFS khusus, Anda harus memberikan akses baca dan tulis ke semua prinsipal IAM. Contoh:

```
"Principal": {
  "AWS": "*"
}
```

CLIENT_ERROR: pemasangan '127.0.0.1: /' gagal. koneksi diatur ulang oleh rekan

Ada dua kemungkinan penyebab kesalahan ini:

- Subnet CodeBuild VPC berada di zona ketersediaan yang berbeda dari target pemasangan Amazon EFS. Anda dapat mengatasinya dengan menambahkan subnet VPC di zona ketersediaan yang sama dengan target pemasangan Amazon EFS.
- Grup keamanan tidak memiliki izin untuk berkomunikasi dengan Amazon EFS. Anda dapat mengatasinya dengan menambahkan aturan masuk untuk mengizinkan semua lalu lintas baik dari VPC (tambahkan blok CIDR utama untuk VPC Anda), atau grup keamanan itu sendiri.

VPC_CLIENT_ERROR: Kesalahan EC2 yang tidak terduga: UnauthorizedOperation

Kesalahan ini terjadi ketika semua subnet dalam konfigurasi VPC Anda untuk CodeBuild proyek adalah subnet publik. Anda harus memiliki setidaknya satu subnet pribadi di VPC untuk memastikan konektivitas jaringan.

AWS CodePipeline sampel untuk CodeBuild

Bagian ini menjelaskan integrasi sampel antara CodePipeline dan CodeBuild.

Sampel	Deskripsi
Contoh CodePipeline/CodeBuild integrasi dan build batch	Sampel ini menunjukkan cara menggunakan AWS CodePipeline untuk membuat proyek build yang menggunakan build batch.
Contoh CodePipeline/CodeBuild integrasi dengan beberapa sumber input dan artefak keluaran	Contoh ini menunjukkan cara menggunakan AWS CodePipeline untuk membuat proyek build yang menggunakan beberapa sumber input untuk membuat beberapa artefak keluaran.

Contoh CodePipeline/CodeBuild integrasi dan build batch

AWS CodeBuild mendukung pembuatan batch. Contoh berikut menunjukkan cara menggunakan AWS CodePipeline untuk membuat proyek build yang menggunakan build batch.

Anda dapat menggunakan JSON-formatted file yang mendefinisikan struktur pipeline Anda, dan kemudian menggunakannya dengan AWS CLI untuk membuat pipeline. Untuk informasi selengkapnya, lihat Referensi struktur AWS CodePipeline pipa di Panduan AWS CodePipeline Pengguna.

Bangun batch dengan artefak individu

Gunakan file JSON berikut sebagai contoh struktur pipeline yang membuat batch build dengan artefak terpisah. Untuk mengaktifkan build batch CodePipeline, atur `BatchEnabled` parameter `configuration` objek ke`true`.

```
{
  "pipeline": {
    "roleArn": "arn:aws:iam::account-id:role/my-AWS-CodePipeline-service-role-name",
    "stages": [
      {
        "name": "Source",
        "actions": [
```

```
        {
          "inputArtifacts": [],
          "name": "Source1",
          "actionTypeId": {
            "category": "Source",
            "owner": "AWS",
            "version": "1",
            "provider": "S3"
          },
          "outputArtifacts": [
            {
              "name": "source1"
            }
          ],
          "configuration": {
            "S3Bucket": "<my-input-bucket-name>",
            "S3ObjectKey": "my-source-code-file-name.zip"
          },
          "runOrder": 1
        },
        {
          "inputArtifacts": [],
          "name": "Source2",
          "actionTypeId": {
            "category": "Source",
            "owner": "AWS",
            "version": "1",
            "provider": "S3"
          },
          "outputArtifacts": [
            {
              "name": "source2"
            }
          ],
          "configuration": {
            "S3Bucket": "<my-other-input-bucket-name>",
            "S3ObjectKey": "my-other-source-code-file-name.zip"
          },
          "runOrder": 1
        }
      ]
    },
    {
      "name": "Build",
```

```
        "actions": [
          {
            "inputArtifacts": [
              {
                "name": "source1"
              },
              {
                "name": "source2"
              }
            ],
            "name": "Build",
            "actionTypeId": {
              "category": "Build",
              "owner": "AWS",
              "version": "1",
              "provider": "CodeBuild"
            },
            "outputArtifacts": [
              {
                "name": "build1"
              },
              {
                "name": "build1_artifact1"
              },
              {
                "name": "build1_artifact2"
              },
              {
                "name": "build2_artifact1"
              },
              {
                "name": "build2_artifact2"
              }
            ],
            "configuration": {
              "ProjectName": "my-build-project-name",
              "PrimarySource": "source1",
              "BatchEnabled": "true"
            },
            "runOrder": 1
          }
        ]
      }
    ],
```

```
    "artifactStore": {
      "type": "S3",
      "location": "<AWS-CodePipeline-internal-bucket-name>"
    },
    "name": "my-pipeline-name",
    "version": 1
  }
}
```

Berikut ini adalah contoh file CodeBuild buildspec yang akan bekerja dengan konfigurasi pipeline ini.

```
version: 0.2
batch:
  build-list:
    - identifier: build1
      env:
        compute-type: BUILD_GENERAL1_SMALL
    - identifier: build2
      env:
        compute-type: BUILD_GENERAL1_MEDIUM

phases:
  build:
    commands:
      - echo 'file' > output_file

artifacts:
  files:
    - output_file
  secondary-artifacts:
    artifact1:
      files:
        - output_file
    artifact2:
      files:
        - output_file
```

Nama-nama artefak keluaran yang ditentukan dalam file JSON pipeline harus sesuai dengan pengenal build dan artefak yang ditentukan dalam file buildspec Anda. Sintaks adalah *buildIdentifier* untuk artefak utama, dan *buildIdentifier* _ *artifactIdentifier* untuk artefak sekunder.

Misalnya, untuk nama artefak keluaran`build1`, CodeBuild akan mengunggah artefak utama `build1` ke lokasi. `build1` Untuk nama keluaran`build1_artifact1`, CodeBuild akan mengunggah artefak `artifact1` sekunder `build1` ke lokasi`build1_artifact1`, dan seterusnya. Jika hanya satu lokasi keluaran yang ditentukan, nama seharusnya *`buildIdentifier`* hanya.

Setelah Anda membuat file JSON, Anda dapat membuat pipeline Anda. Gunakan AWS CLI untuk menjalankan perintah create-pipeline dan meneruskan file ke parameter. `--cli-input-json` Untuk informasi selengkapnya, lihat Membuat pipeline (CLI) di Panduan AWS CodePipeline Pengguna.

Bangun batch dengan artefak gabungan

Gunakan file JSON berikut sebagai contoh struktur pipeline yang membuat batch build dengan artefak gabungan. Untuk mengaktifkan build batch CodePipeline, atur `BatchEnabled` parameter `configuration` objek ke`true`. Untuk menggabungkan artefak build ke lokasi yang sama, atur `CombineArtifacts` parameter `configuration` objek ke`true`.

```
{
 "pipeline": {
  "roleArn": "arn:aws:iam::account-id:role/my-AWS-CodePipeline-service-role-name",
  "stages": [
    {
      "name": "Source",
      "actions": [
        {
          "inputArtifacts": [],
          "name": "Source1",
          "actionTypeId": {
            "category": "Source",
            "owner": "AWS",
            "version": "1",
            "provider": "S3"
          },
          "outputArtifacts": [
            {
              "name": "source1"
            }
          ],
          "configuration": {
            "S3Bucket": "<my-input-bucket-name>",
            "S3ObjectKey": "my-source-code-file-name.zip"
          },
```

```
          "runOrder": 1
        },
        {
          "inputArtifacts": [],
          "name": "Source2",
          "actionTypeId": {
            "category": "Source",
            "owner": "AWS",
            "version": "1",
            "provider": "S3"
          },
          "outputArtifacts": [
            {
              "name": "source2"
            }
          ],
          "configuration": {
            "S3Bucket": "<my-other-input-bucket-name>",
            "S3ObjectKey": "my-other-source-code-file-name.zip"
          },
          "runOrder": 1
        }
      ]
    },
    {
      "name": "Build",
      "actions": [
        {
          "inputArtifacts": [
            {
              "name": "source1"
            },
            {
              "name": "source2"
            }
          ],
          "name": "Build",
          "actionTypeId": {
            "category": "Build",
            "owner": "AWS",
            "version": "1",
            "provider": "CodeBuild"
          },
          "outputArtifacts": [
```

```
            {
              "name": "output1 "
            }
          ],
          "configuration": {
            "ProjectName": "my-build-project-name",
            "PrimarySource": "source1",
             "BatchEnabled": "true",
             "CombineArtifacts": "true"
          },
          "runOrder": 1
        }
      ]
    }
  ],
  "artifactStore": {
    "type": "S3",
    "location": "<AWS-CodePipeline-internal-bucket-name>"
  },
  "name": "my-pipeline-name",
  "version": 1
 }
}
```

Berikut ini adalah contoh file CodeBuild buildspec yang akan bekerja dengan konfigurasi pipeline ini.

```
version: 0.2
batch:
  build-list:
    - identifier: build1
      env:
        compute-type: BUILD_GENERAL1_SMALL
    - identifier: build2
      env:
        compute-type: BUILD_GENERAL1_MEDIUM

phases:
  build:
    commands:
      - echo 'file' > output_file

artifacts:
  files:
```

```
    - output_file
```

Jika artefak gabungan diaktifkan untuk build batch, hanya ada satu output yang diizinkan. CodeBuild akan menggabungkan artefak utama dari semua build menjadi satu file ZIP tunggal.

Setelah Anda membuat file JSON, Anda dapat membuat pipeline Anda. Gunakan AWS CLI untuk menjalankan perintah create-pipeline dan meneruskan file ke parameter. `--cli-input-json` Untuk informasi selengkapnya, lihat Membuat pipeline (CLI) di Panduan AWS CodePipeline Pengguna.

Contoh CodePipeline/CodeBuild integrasi dengan beberapa sumber input dan artefak keluaran

Sebuah AWS CodeBuild proyek dapat mengambil lebih dari satu sumber input. Itu juga dapat membuat lebih dari satu artefak keluaran. Contoh ini menunjukkan cara menggunakan AWS CodePipeline untuk membuat proyek build yang menggunakan beberapa sumber input untuk membuat beberapa artefak keluaran. Untuk informasi selengkapnya, lihat Beberapa sumber input dan sampel artefak keluaran.

Anda dapat menggunakan JSON-formatted file yang mendefinisikan struktur pipeline Anda, dan kemudian menggunakannya dengan AWS CLI untuk membuat pipeline. Gunakan file JSON berikut sebagai contoh struktur pipeline yang membuat build dengan lebih dari satu sumber input dan lebih dari satu artefak keluaran. Kemudian dalam contoh ini Anda melihat bagaimana file ini menentukan beberapa input dan output. Untuk informasi selengkapnya, lihat referensi struktur CodePipeline pipa di Panduan AWS CodePipeline Pengguna.

```
{
 "pipeline": {
  "roleArn": "arn:aws:iam::account-id:role/my-AWS-CodePipeline-service-role-name",
  "stages": [
    {
      "name": "Source",
      "actions": [
        {
          "inputArtifacts": [],
          "name": "Source1",
          "actionTypeId": {
            "category": "Source",
            "owner": "AWS",
            "version": "1",
            "provider": "S3"
```

```
        },
        "outputArtifacts": [
          {
            "name": "source1"
          }
        ],
        "configuration": {
          "S3Bucket": "my-input-bucket-name",
          "S3ObjectKey": "my-source-code-file-name.zip"
        },
        "runOrder": 1
      },
      {
        "inputArtifacts": [],
        "name": "Source2",
        "actionTypeId": {
          "category": "Source",
          "owner": "AWS",
          "version": "1",
          "provider": "S3"
        },
        "outputArtifacts": [
          {
            "name": "source2"
          }
        ],
        "configuration": {
          "S3Bucket": "my-other-input-bucket-name",
          "S3ObjectKey": "my-other-source-code-file-name.zip"
        },
        "runOrder": 1
      }
    ]
  },
  {
    "name": "Build",
    "actions": [
      {
        "inputArtifacts": [
          {
            "name": "source1"
          },
          {
            "name": "source2"
```

```
            }
          ],
          "name": "Build",
          "actionTypeId": {
            "category": "Build",
            "owner": "AWS",
            "version": "1",
            "provider": "AWS CodeBuild"
          },
          "outputArtifacts": [
            {
              "name": "artifact1"
            },
            {
              "name": "artifact2"
            }
          ],
          "configuration": {
            "ProjectName": "my-build-project-name",
            "PrimarySource": "source1"
          },
          "runOrder": 1
        }
      ]
    }
  ],
  "artifactStore": {
    "type": "S3",
    "location": "AWS-CodePipeline-internal-bucket-name"
  },
  "name": "my-pipeline-name",
  "version": 1
 }
}
```

Dalam file JSON ini:

- Salah satu sumber input Anda harus ditunjuk sebagai`PrimarySource`. Sumber ini adalah direktori tempat mencari CodeBuild dan menjalankan file buildspec Anda. Kata kunci `PrimarySource` digunakan untuk menentukan sumber utama di `configuration` bagian CodeBuild tahap dalam file JSON.

- Setiap sumber input diinstal di direktorinya sendiri. Direktori ini disimpan dalam variabel lingkungan bawaan `$CODEBUILD_SRC_DIR` untuk sumber utama dan `$CODEBUILD_SRC_DIR_yourInputArtifactName` untuk semua sumber lainnya. Untuk pipeline dalam sampel ini, dua direktori sumber input adalah `$CODEBUILD_SRC_DIR` dan`$CODEBUILD_SRC_DIR_source2`. Untuk informasi selengkapnya, lihat Variabel lingkungan di lingkungan build.
- Nama-nama artefak keluaran yang ditentukan dalam file JSON pipeline harus cocok dengan nama artefak sekunder yang ditentukan dalam file buildspec Anda. Pipeline ini menggunakan file buildspec berikut. Untuk informasi selengkapnya, lihat Sintaks Buildspec.

```
version: 0.2

phases:
  build:
    commands:
      - touch source1_file
      - cd $CODEBUILD_SRC_DIR_source2
      - touch source2_file

artifacts:
  files:
    - '**/*'
  secondary-artifacts:
    artifact1:
      base-directory: $CODEBUILD_SRC_DIR
      files:
        - source1_file
    artifact2:
      base-directory: $CODEBUILD_SRC_DIR_source2
      files:
        - source2_file
```

Setelah Anda membuat file JSON, Anda dapat membuat pipeline Anda. Gunakan AWS CLI untuk menjalankan perintah create-pipeline dan meneruskan file ke parameter. `--cli-input-json` Untuk informasi selengkapnya, lihat Membuat pipeline (CLI) di Panduan AWS CodePipeline Pengguna.

AWS Config sampel dengan CodeBuild

AWS Config menyediakan inventaris sumber AWS daya Anda dan riwayat perubahan konfigurasi pada sumber daya ini. AWS Config sekarang mendukung AWS CodeBuild sebagai AWS sumber daya, yang berarti layanan dapat melacak CodeBuild proyek Anda. Untuk informasi lebih lanjut tentang AWS Config, lihat Apa itu AWS Config? di Panduan AWS Config Pengembang.

Anda dapat melihat informasi berikut tentang CodeBuild sumber daya di halaman Inventaris Sumber Daya di AWS Config konsol:

- Garis waktu CodeBuild konfigurasi Anda berubah.
- Detail konfigurasi untuk setiap CodeBuild proyek.
- Hubungan dengan AWS sumber daya lain.
- Daftar perubahan pada CodeBuild proyek Anda.

Topik

- Gunakan CodeBuild dengan AWS Config
- Langkah 3: Lihat AWS CodeBuild data di AWS Config konsol

Gunakan CodeBuild dengan AWS Config

Prosedur dalam topik ini menunjukkan kepada Anda cara mengatur AWS Config dan mencari CodeBuild proyek.

Topik

- Prasyarat
- Langkah 1: Mengatur AWS Config
- Langkah 2: Cari AWS CodeBuild proyek

Prasyarat

Buat AWS CodeBuild proyek Anda. Untuk petunjuk, lihat Membuat proyek build.

Langkah 1: Mengatur AWS Config

- Menyiapkan AWS Config (konsol)
- Menyiapkan AWS Config (AWS CLI)

> **Note**
>
> Setelah Anda menyelesaikan penyiapan, mungkin diperlukan waktu hingga 10 menit sebelum Anda dapat melihat AWS CodeBuild proyek di AWS Config konsol.

Langkah 2: Cari AWS CodeBuild proyek

1. Masuk ke Konsol AWS Manajemen dan buka AWS Config konsol di https://console.aws.amazon.com/config.
2. Pada halaman Inventaris sumber daya, pilih AWS CodeBuild Proyek di bawah Jenis sumber daya. Gulir ke bawah dan pilih kotak centang CodeBuildproyek.
3. Pilih Lihat.
4. Setelah daftar CodeBuild proyek ditambahkan, pilih tautan nama CodeBuild proyek di kolom timeline Config.

Langkah 3: Lihat AWS CodeBuild data di AWS Config konsol

Saat Anda mencari sumber daya di halaman inventaris Sumber Daya, Anda dapat memilih AWS Config garis waktu untuk melihat detail tentang CodeBuild proyek Anda. Halaman detail untuk sumber daya menyediakan informasi tentang konfigurasi, hubungan, dan jumlah perubahan yang dilakukan pada sumber daya tersebut.

Blok di bagian atas halaman secara kolektif disebut timeline. Garis waktu menunjukkan tanggal dan waktu rekaman itu dibuat.

Untuk informasi selengkapnya, lihat Melihat detail konfigurasi di AWS Config konsol di Panduan AWS Config Pengembang.

Buat contoh notifikasi untuk CodeBuild

Amazon CloudWatch Events memiliki dukungan bawaan untuk AWS CodeBuild. CloudWatch Peristiwa adalah aliran peristiwa sistem yang menggambarkan perubahan dalam AWS sumber daya Anda. Dengan CloudWatch Acara, Anda menulis aturan deklaratif untuk mengaitkan peristiwa yang menarik dengan tindakan otomatis yang akan diambil. Sampel ini menggunakan Amazon CloudWatch Events dan Amazon Simple Notification Service (Amazon SNS) untuk mengirim pemberitahuan build kepada pelanggan setiap kali build berhasil, gagal, beralih dari satu fase build ke fase build lainnya, atau kombinasi apa pun dari peristiwa ini.

> **Important**
>
> Menjalankan sampel ini dapat mengakibatkan tagihan ke AWS akun Anda. Ini termasuk kemungkinan biaya untuk CodeBuild dan untuk AWS sumber daya dan tindakan yang terkait dengan Amazon CloudWatch dan Amazon SNS. Untuk informasi selengkapnya, lihat CodeBuild harga, CloudWatch harga Amazon, dan harga https://aws.amazon.com/sns/pricing Amazon SNS.

Topik

- Jalankan contoh pemberitahuan build
- Buat referensi format input notifikasi

Jalankan contoh pemberitahuan build

Gunakan prosedur berikut untuk menjalankan contoh pemberitahuan build.

Untuk menjalankan sampel ini

1. Jika Anda sudah menyiapkan topik dan berlangganan di Amazon SNS yang ingin Anda gunakan untuk sampel ini, lanjutkan ke langkah 4. Jika tidak, jika Anda menggunakan pengguna IAM alih-alih akun AWS root atau pengguna administrator untuk bekerja dengan Amazon SNS, tambahkan pernyataan berikut (antara *### BEGIN ADDING STATEMENT HERE ###* dan*### END ADDING STATEMENT HERE ###*) ke pengguna (atau grup IAM yang dikaitkan dengan pengguna). Menggunakan akun AWS root tidak disarankan. Pernyataan ini memungkinkan melihat, membuat, berlangganan, dan menguji pengiriman notifikasi ke topik di Amazon SNS. Ellipses (`...`) digunakan untuk singkatnya dan untuk membantu Anda menemukan tempat menambahkan pernyataan. Jangan menghapus pernyataan apa pun, dan jangan mengetikkan elips ini ke dalam kebijakan yang ada.

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
```

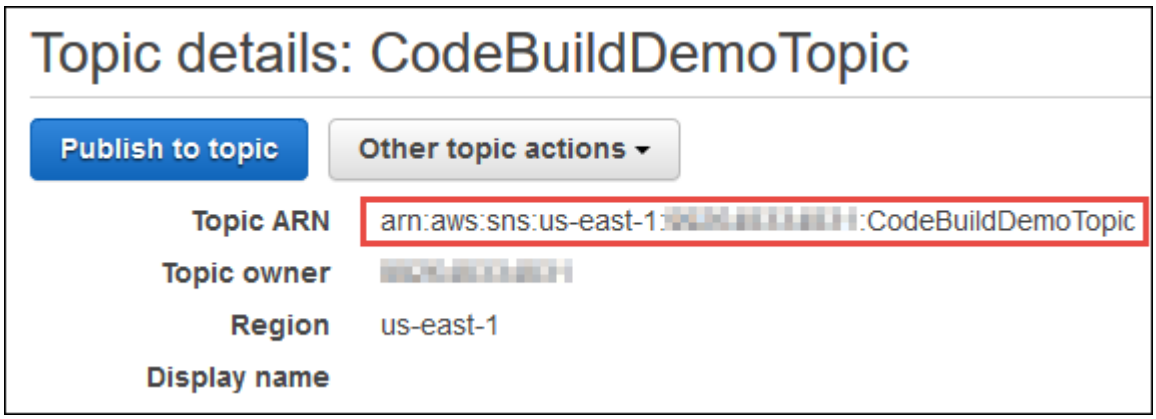
```
            "Action": [
                "sns:CreateTopic",
                "sns:GetTopicAttributes",
                "sns:List*",
                "sns:Publish",
                "sns:SetTopicAttributes",
                "sns:Subscribe"
            ],
            "Resource": "*"
        }
    ]
}
```

> **Note**
>
> Entitas IAM yang mengubah kebijakan ini harus memiliki izin di IAM untuk mengubah kebijakan.
> Untuk informasi selengkapnya, lihat Meng edit kebijakan yang dikelola pelanggan atau bagian “Mengedit atau menghapus kebijakan sebaris untuk grup, pengguna, atau peran” di Bek erja dengan kebijakan sebaris (konsol) di Panduan Pengguna IAM.

2. Buat atau identifikasi topik di Amazon SNS. AWS CodeBuild menggunakan CloudWatch Events untuk mengirim pemberitahuan build ke topik ini melalui Amazon SNS.

   Untuk membuat topik:

   1. Buka konsol Amazon SNS di https://console.aws.amazon.com/sns.
   2. Pilih Buat topik.
   3. Di Buat topik baru, untuk Nama Topik, masukkan nama untuk topik (misalnya,**`CodeBuildDemoTopic`**). (Jika Anda memilih nama yang berbeda, gantilah di seluruh sampel ini.)
   4. Pilih Buat topik.
   5. Pada CodeBuildDemoTopic halaman Topik detail:, salin nilai Topik ARN. Anda memerlukan nilai ini untuk langkah berikutnya.

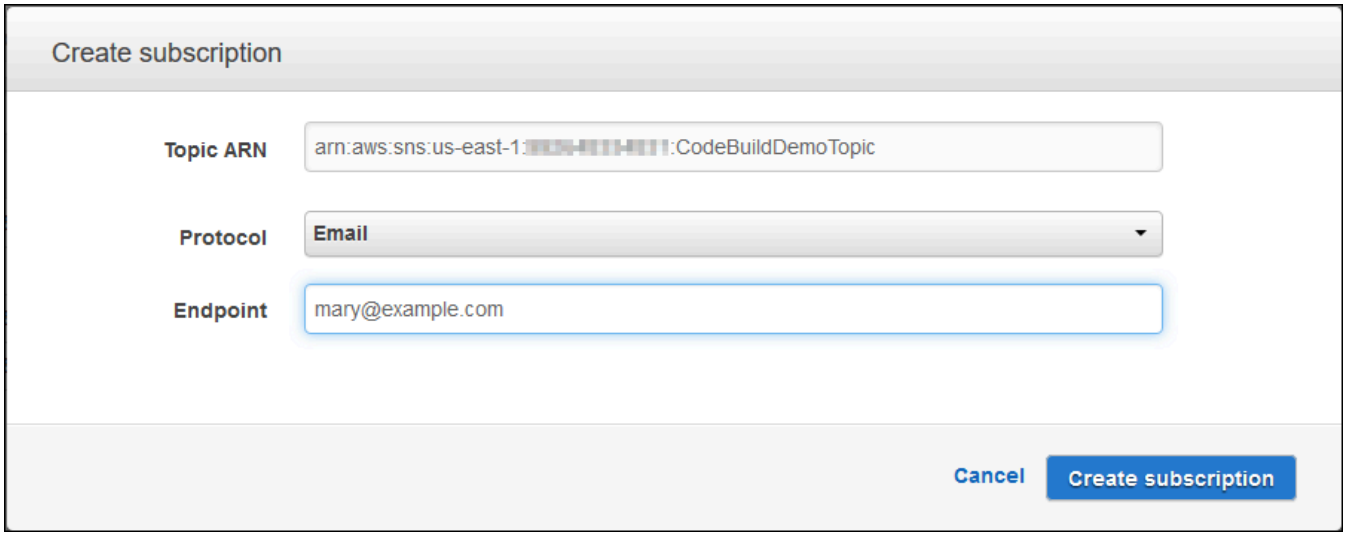


Untuk informasi selengkapnya, lihat Membuat topik di Panduan Pengembang Amazon SNS.

3. Berlangganan satu atau lebih penerima ke topik untuk menerima pemberitahuan email.

   Untuk berlangganan penerima ke topik:

   1. Dengan konsol Amazon SNS terbuka dari langkah sebelumnya, di panel navigasi, pilih Langganan, lalu pilih Buat langganan.
   2. Di Buat langganan, untuk Topik ARN, tempel topik ARN yang Anda salin dari langkah sebelumnya.
   3. Untuk Protokol, pilih Email.
   4. Untuk Endpoint, masukkan alamat email lengkap penerima.



   5. Pilih Buat Langganan.

6. Amazon SNS mengirimkan email konfirmasi langganan ke penerima. Untuk mulai menerima pemberitahuan email, penerima harus memilih tautan Kon firmasi langganan di email konfirmasi langganan. Setelah penerima mengklik tautan, jika berhasil berlangganan, Amazon SNS menampilkan pesan konfirmasi di browser web penerima.

Untuk informasi selengkapnya, lihat Ber langganan topik di Panduan Pengembang Amazon SNS.

4. Jika Anda menggunakan pengguna alih-alih akun AWS root atau pengguna administrator untuk bekerja dengan A CloudWatch cara, tambahkan pernyataan berikut (antara *### BEGIN ADDING STATEMENT HERE ###* dan*### END ADDING STATEMENT HERE ###*) ke pengguna (atau grup IAM yang dikaitkan dengan pengguna). Menggunakan akun AWS root tidak disarankan. Pernyataan ini digunakan untuk memungkinkan pengguna bekerja dengan A CloudWatch cara. Ellipses (`...`) digunakan untuk singkatnya dan untuk membantu Anda menemukan tempat menambahkan pernyataan. Jangan menghapus pernyataan apa pun, dan jangan mengetikkan elips ini ke dalam kebijakan yang ada.

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "events:*",
                "iam:PassRole"
            ],
            "Resource": "arn:aws:iam::*:role/Service*"
        }
    ]
}
```

> **Note**
>
> Entitas IAM yang mengubah kebijakan ini harus memiliki izin di IAM untuk mengubah kebijakan.

Untuk informasi selengkapnya, lihat Meng edit kebijakan yang dikelola pelanggan atau bagian “Mengedit atau menghapus kebijakan sebaris untuk grup, pengguna, atau peran” di Bek erja dengan kebijakan sebaris (konsol) di Panduan Pengguna IAM.

5. Buat aturan di A CloudWatch cara. Untuk melakukan ini, buka CloudWatch konsol, di https://console.aws.amazon.com/cloudwatch.
6. Di panel navigasi, di bawah A cara, pilih A turan, lalu pilih Buat aturan.
7. Pada Langkah 1: Buat halaman aturan, Pola Per istiwa dan Mem bangun pola acara untuk mencocokkan peristiwa berdasarkan layanan seharusnya sudah dipilih.
8. Untuk Nama Layanan, pilih CodeBuild. Untuk Jenis A cara, Semua A cara seharusnya sudah dipilih.
9. Kode berikut harus ditampilkan di Pratinjau P ola Peristiwa:

```
{
  "source": [
    "aws.codebuild"
  ]
}
```

10. Pilih Edit dan ganti kode di Prat injau Pola Peristiwa dengan salah satu dari dua pola aturan berikut.

    Pola aturan pertama ini memicu peristiwa ketika build dimulai atau selesai untuk proyek build yang ditentukan di AWS CodeBuild.

```
{
  "source": [
    "aws.codebuild"
  ],
  "detail-type": [
    "CodeBuild Build State Change"
  ],
  "detail": {
    "build-status": [
      "IN_PROGRESS",
      "SUCCEEDED",
      "FAILED",
      "STOPPED"
    ],
    "project-name": [
```

```
      "my-demo-project-1",
      "my-demo-project-2"
    ]
  }
}
```

Dalam aturan sebelumnya, buat perubahan kode berikut sesuai kebutuhan.

- Untuk memicu peristiwa ketika build dimulai atau selesai, tinggalkan semua nilai seperti yang ditunjukkan dalam `build-status` array, atau hapus `build-status` array sama sekali.
- Untuk memicu peristiwa hanya ketika build selesai, hapus `IN_PROGRESS` dari `build-status` array.
- Untuk memicu peristiwa hanya ketika build dimulai, hapus semua nilai kecuali `IN_PROGRESS` dari `build-status` array.
- Untuk memicu peristiwa untuk semua proyek build, hapus `project-name` array sama sekali.
- Untuk memicu peristiwa hanya untuk proyek build individual, tentukan nama setiap proyek build dalam `project-name` array.

Pola aturan kedua ini memicu peristiwa setiap kali build berpindah dari satu fase build ke fase build lainnya untuk proyek build yang ditentukan di AWS CodeBuild.

```
{
  "source": [
    "aws.codebuild"
  ],
  "detail-type": [
    "CodeBuild Build Phase Change"
  ],
  "detail": {
    "completed-phase": [
      "SUBMITTED",
      "PROVISIONING",
      "DOWNLOAD_SOURCE",
      "INSTALL",
      "PRE_BUILD",
      "BUILD",
      "POST_BUILD",
      "UPLOAD_ARTIFACTS",
      "FINALIZING"
```

```
    ],
    "completed-phase-status": [
      "TIMED_OUT",
      "STOPPED",
      "FAILED",
      "SUCCEEDED",
      "FAULT",
      "CLIENT_ERROR"
    ],
    "project-name": [
      "my-demo-project-1",
      "my-demo-project-2"
    ]
  }
}
```

Dalam aturan sebelumnya, buat perubahan kode berikut sesuai kebutuhan.

- Untuk memicu peristiwa untuk setiap perubahan fase build (yang mungkin mengirim hingga sembilan pemberitahuan untuk setiap build), tinggalkan semua nilai seperti yang ditunjukkan dalam `completed-phase` array, atau hapus `completed-phase` array sama sekali.
- Untuk memicu peristiwa hanya untuk perubahan fase build individual, hapus nama setiap fase build dalam `completed-phase` array yang tidak ingin Anda picu peristiwa.
- Untuk memicu peristiwa untuk setiap perubahan status fase build, tinggalkan semua nilai seperti yang ditunjukkan dalam `completed-phase-status` array, atau hapus `completed-phase-status` array sama sekali.
- Untuk memicu peristiwa hanya untuk perubahan status fase build individual, hapus nama setiap status fase build dalam `completed-phase-status` array yang tidak ingin Anda picu peristiwa.
- Untuk memicu peristiwa untuk semua proyek build, hapus `project-name` array.
- Untuk memicu peristiwa untuk proyek build individual, tentukan nama setiap proyek build dalam `project-name` array.

Untuk informasi selengkapnya tentang pola peristiwa, lihat Pola Peristiwa di Panduan EventBridge Pengguna Amazon.

Untuk informasi selengkapnya tentang pemfilteran dengan pola peristiwa, lihat Content-based Memfilter dengan Pola Peristiwa di Panduan EventBridge Pengguna Amazon.

> **Note**
>
> Jika Anda ingin memicu peristiwa untuk perubahan status build dan perubahan fase build, Anda harus membuat dua aturan terpisah: satu untuk perubahan status build dan satu lagi untuk perubahan fase build. Jika Anda mencoba menggabungkan kedua aturan menjadi satu aturan, aturan gabungan mungkin menghasilkan hasil yang tidak terduga atau berhenti bekerja sama sekali.

Setelah selesai mengganti kode, pilih S impan.

11. Untuk Targets (Target), pilih Add target (Tambahkan target).
12. Dalam daftar target, pilih topik SNS.
13. Untuk Top ik, pilih topik yang Anda identifikasi atau buat sebelumnya.
14. Perluas Konfigurasi input, lalu pilih Input Transformer.
15. Di kotak Path Input, masukkan salah satu jalur input berikut.

    Untuk aturan dengan `detail-type` nilai`CodeBuild Build State Change`, masukkan yang berikut ini.

    ```
    {"build-id":"$.detail.build-id","project-name":"$.detail.project-name","build-status":"$.detail.build-status"}
    ```

    Untuk aturan dengan `detail-type` nilai`CodeBuild Build Phase Change`, masukkan yang berikut ini.

    ```
    {"build-id":"$.detail.build-id","project-name":"$.detail.project-name","completed-phase":"$.detail.completed-phase","completed-phase-status":"$.detail.completed-phase-status"}
    ```

    Untuk mendapatkan jenis informasi lainnya, lihatBuat referensi format input notifikasi.
16. Di kotak Template Input, masukkan salah satu templat input berikut.

    Untuk aturan dengan `detail-type` nilai`CodeBuild Build State Change`, masukkan yang berikut ini.

```
"Build '<build-id>' for build project '<project-name>' has reached the build status
 of '<build-status>'."
```

Untuk aturan dengan `detail-type` nilai`CodeBuild Build Phase Change`, masukkan yang berikut ini.

```
"Build '<build-id>' for build project '<project-name>' has completed the build
 phase of '<completed-phase>' with a status of '<completed-phase-status>'."
```

17. Pilih Konfigurasikan detail.
18. Pada halaman Langkah 2: Konfigurasikan detail aturan, masukkan nama dan deskripsi opsional. Untuk Status, biarkan Di aktifkan dipilih.
19. Pilih Buat aturan.
20. Buat proyek build, jalankan build, dan lihat informasi build.
21. Konfirmas CodeBuild ikan bahwa sekarang berhasil mengirim pemberitahuan build. Misalnya, periksa untuk melihat apakah email pemberitahuan build sekarang ada di kotak masuk Anda.

Untuk mengubah perilaku aturan, di CloudWatch konsol, pilih aturan yang ingin Anda ubah, pilih Tind akan, lalu pilih Edit. Buat perubahan pada aturan, pilih Konfigurasikan detail, lalu pilih Per barui aturan.

Untuk berhenti menggunakan aturan untuk mengirim pemberitahuan build, di CloudWatch konsol, pilih aturan yang ingin Anda berhenti gunakan, pilih Tindakan , lalu pilih Non aktifkan.

Untuk menghapus aturan sama sekali, di CloudWatch konsol, pilih aturan yang ingin Anda hapus, pilih Tind akan, lalu pilih H apus.

Buat referensi format input notifikasi

CloudWatch mengirimkan notifikasi dalam format JSON.

Pemberitahuan perubahan status build menggunakan format berikut:

```
{
  "version": "0",
  "id": "c030038d-8c4d-6141-9545-00ff7b7153EX",
  "detail-type": "CodeBuild Build State Change",
  "source": "aws.codebuild",
```

```
  "account": "123456789012",
  "time": "2017-09-01T16:14:28Z",
  "region": "us-west-2",
  "resources":[
    "arn:aws:codebuild:us-west-2:123456789012:build/my-sample-project:8745a7a9-
c340-456a-9166-edf953571bEX"
  ],
  "detail":{
    "build-status": "SUCCEEDED",
    "project-name": "my-sample-project",
    "build-id": "arn:aws:codebuild:us-west-2:123456789012:build/my-sample-
project:8745a7a9-c340-456a-9166-edf953571bEX",
    "additional-information": {
      "artifact": {
        "md5sum": "da9c44c8a9a3cd4b443126e823168fEX",
        "sha256sum":
 "6ccc2ae1df9d155ba83c597051611c42d60e09c6329dcb14a312cecc0a8e39EX",
        "location": "arn:aws:s3:::codebuild-123456789012-output-bucket/my-output-
artifact.zip"
      },
      "environment": {
        "image": "aws/codebuild/standard:5.0",
        "privileged-mode": false,
        "compute-type": "BUILD_GENERAL1_SMALL",
        "type": "LINUX_CONTAINER",
        "environment-variables": []
      },
      "timeout-in-minutes": 60,
      "build-complete": true,
      "initiator": "MyCodeBuildDemoUser",
      "build-start-time": "Sep 1, 2017 4:12:29 PM",
      "source": {
        "location": "codebuild-123456789012-input-bucket/my-input-artifact.zip",
        "type": "S3"
      },
      "logs": {
        "group-name": "/aws/codebuild/my-sample-project",
        "stream-name": "8745a7a9-c340-456a-9166-edf953571bEX",
        "deep-link": "https://console.aws.amazon.com/cloudwatch/home?region=us-
west-2#logEvent:group=/aws/codebuild/my-sample-project;stream=8745a7a9-c340-456a-9166-
edf953571bEX"
      },
      "phases": [
        {
```

```
      "phase-context": [],
      "start-time": "Sep 1, 2017 4:12:29 PM",
      "end-time": "Sep 1, 2017 4:12:29 PM",
      "duration-in-seconds": 0,
      "phase-type": "SUBMITTED",
      "phase-status": "SUCCEEDED"
    },
    {
      "phase-context": [],
      "start-time": "Sep 1, 2017 4:12:29 PM",
      "end-time": "Sep 1, 2017 4:13:05 PM",
      "duration-in-seconds": 36,
      "phase-type": "PROVISIONING",
      "phase-status": "SUCCEEDED"
    },
    {
      "phase-context": [],
      "start-time": "Sep 1, 2017 4:13:05 PM",
      "end-time": "Sep 1, 2017 4:13:10 PM",
      "duration-in-seconds": 4,
      "phase-type": "DOWNLOAD_SOURCE",
      "phase-status": "SUCCEEDED"
    },
    {
      "phase-context": [],
      "start-time": "Sep 1, 2017 4:13:10 PM",
      "end-time": "Sep 1, 2017 4:13:10 PM",
      "duration-in-seconds": 0,
      "phase-type": "INSTALL",
      "phase-status": "SUCCEEDED"
    },
    {
      "phase-context": [],
      "start-time": "Sep 1, 2017 4:13:10 PM",
      "end-time": "Sep 1, 2017 4:13:10 PM",
      "duration-in-seconds": 0,
      "phase-type": "PRE_BUILD",
      "phase-status": "SUCCEEDED"
    },
    {
      "phase-context": [],
      "start-time": "Sep 1, 2017 4:13:10 PM",
      "end-time": "Sep 1, 2017 4:14:21 PM",
      "duration-in-seconds": 70,
```

```
          "phase-type": "BUILD",
          "phase-status": "SUCCEEDED"
        },
        {
          "phase-context": [],
          "start-time": "Sep 1, 2017 4:14:21 PM",
          "end-time": "Sep 1, 2017 4:14:21 PM",
          "duration-in-seconds": 0,
          "phase-type": "POST_BUILD",
          "phase-status": "SUCCEEDED"
        },
        {
          "phase-context": [],
          "start-time": "Sep 1, 2017 4:14:21 PM",
          "end-time": "Sep 1, 2017 4:14:21 PM",
          "duration-in-seconds": 0,
          "phase-type": "UPLOAD_ARTIFACTS",
          "phase-status": "SUCCEEDED"
        },
         {
          "phase-context": [],
          "start-time": "Sep 1, 2017 4:14:21 PM",
          "end-time": "Sep 1, 2017 4:14:26 PM",
          "duration-in-seconds": 4,
          "phase-type": "FINALIZING",
          "phase-status": "SUCCEEDED"
        },
        {
          "start-time": "Sep 1, 2017 4:14:26 PM",
          "phase-type": "COMPLETED"
        }
      ]
    },
    "current-phase": "COMPLETED",
    "current-phase-context": "[]",
    "version": "1"
  }
}
```

Pemberitahuan perubahan fase build menggunakan format berikut:

```
{
  "version": "0",
```

```
  "id": "43ddc2bd-af76-9ca5-2dc7-b695e15adeEX",
  "detail-type": "CodeBuild Build Phase Change",
  "source": "aws.codebuild",
  "account": "123456789012",
  "time": "2017-09-01T16:14:21Z",
  "region": "us-west-2",
  "resources":[
    "arn:aws:codebuild:us-west-2:123456789012:build/my-sample-project:8745a7a9-
c340-456a-9166-edf953571bEX"
  ],
  "detail":{
    "completed-phase": "COMPLETED",
    "project-name": "my-sample-project",
    "build-id": "arn:aws:codebuild:us-west-2:123456789012:build/my-sample-
project:8745a7a9-c340-456a-9166-edf953571bEX",
    "completed-phase-context": "[]",
    "additional-information": {
      "artifact": {
        "md5sum": "da9c44c8a9a3cd4b443126e823168fEX",
        "sha256sum":
 "6ccc2ae1df9d155ba83c597051611c42d60e09c6329dcb14a312cecc0a8e39EX",
        "location": "arn:aws:s3:::codebuild-123456789012-output-bucket/my-output-
artifact.zip"
      },
      "environment": {
        "image": "aws/codebuild/standard:5.0",
        "privileged-mode": false,
        "compute-type": "BUILD_GENERAL1_SMALL",
        "type": "LINUX_CONTAINER",
        "environment-variables": []
      },
      "timeout-in-minutes": 60,
      "build-complete": true,
      "initiator": "MyCodeBuildDemoUser",
      "build-start-time": "Sep 1, 2017 4:12:29 PM",
      "source": {
        "location": "codebuild-123456789012-input-bucket/my-input-artifact.zip",
        "type": "S3"
      },
      "logs": {
        "group-name": "/aws/codebuild/my-sample-project",
        "stream-name": "8745a7a9-c340-456a-9166-edf953571bEX",
```

```
        "deep-link": "https://console.aws.amazon.com/cloudwatch/home?region=us-
west-2#logEvent:group=/aws/codebuild/my-sample-project;stream=8745a7a9-c340-456a-9166-
edf953571bEX"
      },
      "phases": [
        {
          "phase-context": [],
          "start-time": "Sep 1, 2017 4:12:29 PM",
          "end-time": "Sep 1, 2017 4:12:29 PM",
          "duration-in-seconds": 0,
          "phase-type": "SUBMITTED",
          "phase-status": "SUCCEEDED"
        },
        {
          "phase-context": [],
          "start-time": "Sep 1, 2017 4:12:29 PM",
          "end-time": "Sep 1, 2017 4:13:05 PM",
          "duration-in-seconds": 36,
          "phase-type": "PROVISIONING",
          "phase-status": "SUCCEEDED"
        },
        {
          "phase-context": [],
          "start-time": "Sep 1, 2017 4:13:05 PM",
          "end-time": "Sep 1, 2017 4:13:10 PM",
          "duration-in-seconds": 4,
          "phase-type": "DOWNLOAD_SOURCE",
          "phase-status": "SUCCEEDED"
        },
        {
          "phase-context": [],
          "start-time": "Sep 1, 2017 4:13:10 PM",
          "end-time": "Sep 1, 2017 4:13:10 PM",
          "duration-in-seconds": 0,
          "phase-type": "INSTALL",
          "phase-status": "SUCCEEDED"
        },
        {
          "phase-context": [],
          "start-time": "Sep 1, 2017 4:13:10 PM",
          "end-time": "Sep 1, 2017 4:13:10 PM",
          "duration-in-seconds": 0,
          "phase-type": "PRE_BUILD",
          "phase-status": "SUCCEEDED"
```

```
        },
        {
          "phase-context": [],
          "start-time": "Sep 1, 2017 4:13:10 PM",
          "end-time": "Sep 1, 2017 4:14:21 PM",
          "duration-in-seconds": 70,
          "phase-type": "BUILD",
          "phase-status": "SUCCEEDED"
        },
        {
          "phase-context": [],
          "start-time": "Sep 1, 2017 4:14:21 PM",
          "end-time": "Sep 1, 2017 4:14:21 PM",
          "duration-in-seconds": 0,
          "phase-type": "POST_BUILD",
          "phase-status": "SUCCEEDED"
        },
        {
          "phase-context": [],
          "start-time": "Sep 1, 2017 4:14:21 PM",
          "end-time": "Sep 1, 2017 4:14:21 PM",
          "duration-in-seconds": 0,
          "phase-type": "UPLOAD_ARTIFACTS",
          "phase-status": "SUCCEEDED"
        },
        {
          "phase-context": [],
          "start-time": "Sep 1, 2017 4:14:21 PM",
          "end-time": "Sep 1, 2017 4:14:26 PM",
          "duration-in-seconds": 4,
          "phase-type": "FINALIZING",
          "phase-status": "SUCCEEDED"
        },
        {
          "start-time": "Sep 1, 2017 4:14:26 PM",
          "phase-type": "COMPLETED"
        }
      ]
    },
    "completed-phase-status": "SUCCEEDED",
    "completed-phase-duration-seconds": 4,
    "version": "1",
    "completed-phase-start": "Sep 1, 2017 4:14:21 PM",
    "completed-phase-end": "Sep 1, 2017 4:14:26 PM"
```

```
  }
}
```

Buat sampel lencana dengan CodeBuild

AWS CodeBuild sekarang mendukung penggunaan lencana build, yang menyediakan gambar (badge) yang dapat disematkan dan dihasilkan secara dinamis yang menampilkan status build terbaru untuk sebuah proyek. Gambar ini dapat diakses melalui URL yang tersedia untuk umum yang dihasilkan untuk CodeBuild proyek Anda. Ini memungkinkan siapa saja untuk melihat status CodeBuild proyek. Lencana build tidak berisi informasi keamanan apa pun, sehingga tidak memerlukan otentikasi.

Topik

- Buat proyek build dengan lencana build
- Akses AWS CodeBuild membangun lencana
- Publikasikan CodeBuild lencana build
- CodeBuild status lencana

Buat proyek build dengan lencana build

Gunakan salah satu prosedur berikut untuk membuat proyek build dengan lencana build diaktifkan. Anda dapat menggunakan AWS CLI atau Konsol Manajemen AWS.

Untuk membuat proyek build dengan lencana build diaktifkan (AWS CLI)

- Untuk informasi tentang membuat proyek build, lihatBuat proyek build (AWS CLI). Untuk menyertakan lencana build dengan AWS CodeBuild proyek Anda, Anda harus menentukan *badgeEnabled* dengan nilai`true`.

Untuk membuat proyek build dengan lencana build diaktifkan (konsol)

1. Buka AWS CodeBuild konsol di https://console.aws.amazon.com/codesuite/codebuild/home.
2. Jika halaman CodeBuild informasi ditampilkan, pilih Buat proyek build. Jika tidak, pada panel navigasi, per luas Bang un, pilih Bang un proyek, lalu pilih Buat proyek build.

3. Di Nama proyek, masukkan nama untuk proyek pembangunan ini. Nama proyek build harus unik di setiap AWS akun. Anda juga dapat menyertakan deskripsi opsional proyek build untuk membantu pengguna lain memahami untuk apa proyek ini digunakan.
4. Di Sumber, untuk Penyedia Sumber, pilih jenis penyedia kode sumber, lalu lakukan salah satu hal berikut:

> **Note**
>
> CodeBuild tidak mendukung lencana build dengan penyedia sumber Amazon S3. Karena AWS CodePipeline menggunakan Amazon S3 untuk transfer artefak, lencana build tidak didukung untuk proyek build yang merupakan bagian dari pipeline yang dibuat di. CodePipeline

- Jika Anda memilih CodeCommit, maka untuk Repositori, pilih nama repositori. Pilih Aktifkan lencana build untuk membuat status build proyek Anda terlihat dan dapat disematkan.
- Jika Anda memilih GitHub, ikuti instruksi untuk menghubungkan (atau menyambung kembali). GitHub Pada halaman GitHub Otorisasi aplikasi, untuk Akses organisasi, pilih Minta akses di samping setiap repositori AWS CodeBuild yang ingin Anda akses. Setelah Anda memilih Otorisasi aplikasi, kembali di AWS CodeBuild konsol, untuk Repositori, pilih nama repositori yang berisi kode sumber. Pilih Aktifkan lencana build untuk membuat status build proyek Anda terlihat dan dapat disematkan.
- Jika Anda memilih Bitbucket, ikuti instruksi untuk menghubungkan (atau menyambung kembali) dengan Bitbucket. Pada halaman Bitbucket Kon firmasikan akses ke akun Anda, untuk Ak ses Organisasi, pilih Berikan akses. Setelah Anda memilih Berikan akses, kembali di AWS CodeBuild konsol, untuk Repositori, pilih nama repositori yang berisi kode sumber. Pilih Aktifkan lencana build untuk membuat status build proyek Anda terlihat dan dapat disematkan.

> **Important**
>
> Memperbarui sumber proyek Anda dapat memengaruhi keakuratan lencana build proyek.

5. Di Lingkungan:

   Untuk gambar Lingkungan, lakukan salah satu hal berikut:

- Untuk menggunakan image Docker yang dikelola oleh AWS CodeBuild, pilih Gambar terkel ola, lalu buat pilihan dari Sistem operasi, Runtime (s), Gambar, dan versi Gambar. Buat pilihan dari jenis Lingkungan jika tersedia.
- Untuk menggunakan gambar Docker lain, pilih Custom image. Untuk tipe Lingkungan, pilih ARM, Linux, Linux GPU, atau Windows. Jika Anda memilih Registry lain, untuk URL registri eksternal, masukkan nama dan tag gambar Docker di Docker Hub, menggunakan format. *docker repository*/*docker image name* Jika Anda memilih Amazon ECR, gunakan repositori Amazon ECR dan gambar Amazon ECR untuk memilih gambar Docker di AWS akun Anda.
- Untuk menggunakan image Docker pribadi, pilih Gambar kustom. Untuk tipe Lingkungan, pilih ARM, Linux, Linux GPU, atau Windows. Untuk Registri gambar, pilih Registry lain, lalu masukkan ARN kredensional untuk image Docker pribadi Anda. KredenSIAL harus dibuat oleh Manajer Rahasia. Untuk informasi lebih lanjut, lihat Apa itu AWS Secrets Manager? di Panduan Pengguna AWS Secrets Manager .

6. Di Peran Layanan, lakukan salah satu hal berikut:

   - Jika Anda tidak memiliki peran CodeBuild layanan, pilih Per an layanan baru. Di Nama peran, masukkan nama untuk peran baru.
   - Jika Anda memiliki peran CodeBuild layanan, pilih Peran layanan yang ada. Di Role ARN, pilih peran layanan.

   > **Note**
   >
   > Saat Anda menggunakan konsol untuk membuat atau memperbarui proyek build, Anda dapat membuat peran CodeBuild layanan secara bersamaan. Secara default, peran hanya berfungsi dengan proyek build itu. Jika Anda menggunakan konsol untuk mengaitkan peran layanan ini dengan proyek build lain, peran tersebut diperbarui untuk bekerja dengan proyek build lainnya. Peran layanan dapat bekerja dengan hingga 10 proyek pembangunan.

7. Di Buildspec, lakukan salah satu hal berikut:

   - Pilih Gunakan file buildspec untuk menggunakan file buildspec.yml di direktori root kode sumber.
   - Pilih S isipkan perintah build untuk menggunakan konsol untuk menyisipkan perintah build.

Untuk informasi selengkapnya, lihat Referensi Buildspec.

8. Di Artifacts, untuk Type, lakukan salah satu hal berikut:

   - Jika Anda tidak ingin membuat artefak keluaran build, pilih Tidak ada artefak.
   - Untuk menyimpan output build dalam bucket S3, pilih Amazon S3, lalu lakukan hal berikut:
     - Jika Anda ingin menggunakan nama proyek Anda untuk file atau folder ZIP keluaran build, biarkan Nama kosong. Jika tidak, masukkan nama. Secara default, nama artefak adalah nama proyek. Jika Anda ingin menggunakan nama yang berbeda, masukkan di kotak nama artefak. Jika Anda ingin menampilkan file ZIP, sertakan ekstensi zip.
     - Untuk nama Bucket, pilih nama bucket keluaran.
     - Jika Anda memilih Sisip kan perintah build sebelumnya dalam prosedur ini, untuk file Output, masukkan lokasi file dari build yang ingin Anda masukkan ke dalam file atau folder ZIP keluaran build. Untuk beberapa lokasi, pisahkan setiap lokasi dengan koma (misalnya,`appspec.yml, target/my-app.jar`). Untuk informasi lebih lanjut, lihat deskripsi `files` diSintaks Buildspec.
9. Per luas Konfigurasi tambahan dan pilih opsi yang sesuai.
10. Pilih Buat proyek pembangunan. Pada halaman Review, pilih Mulai build untuk menjalankan build.

Akses AWS CodeBuild membangun lencana

Anda dapat menggunakan AWS CodeBuild konsol atau AWS CLI untuk mengakses lencana build.

- Di CodeBuild konsol, dalam daftar proyek build, di kolom Nama, pilih tautan yang sesuai dengan proyek build. Pada *`project-name`* halaman Build project:, di Konfigurasi, pilih Sal in URL lencana. Untuk informasi selengkapnya, lihat Melihat detail proyek build (konsol).
- Di dalam AWS CLI, jalankan `batch-get-projects` perintah. URL lencana build disertakan dalam bagian detail lingkungan proyek dari output. Untuk informasi selengkapnya, lihat Lihat detail proyek build (AWS CLI).

URL permintaan lencana build dibuat dengan cabang default umum, tetapi Anda dapat menentukan cabang apa pun di repositori sumber yang telah Anda gunakan untuk menjalankan build. Contoh:

```
https://codebuild.us-east-1.amazon.com/badges?uuid=...&branch=<branch>
```

Anda juga dapat menentukan tag dari repositori sumber Anda dengan mengganti `branch` parameter dengan `tag` parameter di URL lencana. Contoh:

```
https://codebuild.us-east-1.amazon.com/badges?uuid=...&tag=<tag>
```

Publikasikan CodeBuild lencana build

Anda dapat menampilkan status build terbaru dalam file penurunan harga menggunakan URL lencana build Anda dalam gambar penurunan harga. Ini berguna untuk menampilkan status build terbaru dalam file readme.md di repositori sumber Anda (misalnya, GitHub atau). CodeCommit Contoh:

```
![](<build badge URL>)
```

CodeBuild status lencana

Lencana CodeBuild build dapat memiliki salah satu status berikut.

- LUL US Build terbaru pada cabang tertentu dilewati.
- GAG AL Build terbaru pada cabang tertentu habis waktunya, gagal, rusak, atau dihentikan.
- IN_PROGR ESS Pembuatan terbaru pada cabang yang diberikan sedang berlangsung.
- TIDAK DIKETAHUI Proyek belum menjalankan build untuk cabang tertentu atau sama sekali. Selain itu, fitur build lencana mungkin telah dinonaktifkan.

'Laporan uji menggunakan AWS CLI'sampel

Pengujian yang Anda tentukan dalam file buildspec dijalankan selama build Anda. Contoh ini menunjukkan kepada Anda cara menggunakan AWS CLI untuk memasukkan tes ke dalam build in CodeBuild. Anda dapat menggunakan JUnit untuk membuat pengujian unit, atau Anda dapat menggunakan alat lain untuk membuat pengujian konfigurasi. Anda kemudian dapat mengevaluasi hasil pengujian untuk memperbaiki masalah atau mengoptimalkan aplikasi Anda.

Anda dapat menggunakan CodeBuild API atau AWS CodeBuild konsol untuk mengakses hasil pengujian. Contoh ini menunjukkan cara mengonfigurasi laporan Anda sehingga hasil pengujiannya diekspor ke bucket S3.

Topik

- Jalankan sampel laporan pengujian

Jalankan sampel laporan pengujian

Gunakan langkah-langkah berikut untuk menjalankan sampel laporan pengujian.

Topik

- Prasyarat
- Langkah 1: Buat grup laporan
- Langkah 2: Konfigurasikan proyek dengan grup laporan
- Langkah 3: Jalankan dan melihat hasil laporan

Prasyarat

- Buat kasus uji Anda. Sampel ini ditulis dengan asumsi bahwa Anda memiliki kasus uji untuk disertakan dalam laporan pengujian sampel Anda. Anda menentukan lokasi file pengujian Anda di file buildspec.

  Format file laporan pengujian berikut didukung:

  - Mentimun JSON (.json)
  - JUnit XML (.xml)
  - nUnit XML (.xml)
  - XML NuNiT3 (.xml)
  - XML TestNG (.xml)
  - Visual Studio TRX (.trx)
  - Visual Studio TRX XML (.xml)

  Buat kasus uji Anda dengan kerangka pengujian apa pun yang dapat membuat file laporan dalam salah satu format ini (misalnya, plugin Surefire JUnit, TestNG, atau Cucumber).
- Buat bucket S3 dan catat namanya. Untuk informasi selengkapnya, lihat Bagaimana cara membuat bucket S3? dalam Panduan Pengguna Amazon S3.
- Buat peran IAM dan catat ARN-nya. Anda membutuhkan ARN saat membuat proyek pembangunan Anda.

- Jika peran Anda tidak memiliki izin berikut, tambahkan izin tersebut.

```
{
    "Effect": "Allow",
    "Resource": [
        "*"
    ],
    "Action": [
        "codebuild:CreateReportGroup",
        "codebuild:CreateReport",
        "codebuild:UpdateReport",
        "codebuild:BatchPutTestCases"
    ]
}
```

Untuk informasi selengkapnya, lihat Izin untuk operasi pelaporan pengujian.

Langkah 1: Buat grup laporan

1. Buat file bernama `CreateReportGroupInput.json`.
2. Buat folder di bucket S3 tempat hasil pengujian Anda diekspor.
3. Salin yang berikut ini ke`CreateReportGroupInput.json`. Untuk*<bucket-name>*, gunakan nama bucket S3. Untuk*<path-to-folder>*, masukkan jalur ke folder di bucket S3 Anda.

```
{
  "name": "<report-name>",
  "type": "TEST",
  "exportConfig": {
    "exportConfigType": "S3",
    "s3Destination": {
      "bucket": "<bucket-name>",
      "path": "<path-to-folder>",
      "packaging": "NONE"
    }
  }
}
```

4. Jalankan perintah berikut di direktori yang berisi`CreateReportGroupInput.json`.

```
aws codebuild create-report-group --cli-input-json file://
CreateReportGroupInput.json
```

Output-nya akan terlihat seperti berikut. Catat ARN untuk`reportGroup`. Anda menggunakannya saat membuat proyek yang menggunakan grup laporan ini.

```
{
  "reportGroup": {
    "arn": "arn:aws:codebuild:us-west-2:123456789012:report-group/<report-name>",
    "name": "<report-name>",
    "type": "TEST",
    "exportConfig": {
      "exportConfigType": "S3",
      "s3Destination": {
        "bucket": "<s3-bucket-name>",
        "path": "<folder-path>",
        "packaging": "NONE",
        "encryptionKey": "arn:aws:kms:us-west-2:123456789012:alias/aws/s3"
      }
    },
    "created": 1570837165.885,
    "lastModified": 1570837165.885
  }
}
```

Langkah 2: Konfigurasikan proyek dengan grup laporan

Untuk menjalankan laporan, pertama-tama Anda membuat proyek CodeBuild build yang dikonfigurasi dengan grup laporan Anda. Kasus uji yang ditentukan untuk grup laporan Anda dijalankan saat Anda menjalankan build.

1. Buat file buildspec bernama. `buildspec.yml`
2. Gunakan YAML berikut sebagai template untuk `buildspec.yml` file Anda. Pastikan untuk menyertakan perintah yang menjalankan pengujian Anda. Di `reports` bagian ini, tentukan file yang berisi hasil kasus uji Anda. File-file ini menyimpan hasil tes yang dapat Anda akses CodeBuild. Mereka kedaluwarsa 30 hari setelah mereka dibuat. File-file ini berbeda dari file hasil kasus uji mentah yang Anda ekspor ke bucket S3.

```
version: 0.2
    phases:
    install:
        runtime-versions:
            java: openjdk8
    build:
      commands:
        - echo Running tests
        - <enter commands to run your tests>

    reports:
      <report-name-or-arn>: #test file information
      files:
        - '<test-result-files>'
      base-directory: '<optional-base-directory>'
      discard-paths: false #do not remove file paths from test result files
```

> **Note**
>
> Alih-alih ARN dari grup laporan yang ada, Anda juga dapat menentukan nama untuk grup laporan yang belum dibuat. Jika Anda menentukan nama alih-alih AR CodeBuild N, membuat grup laporan saat menjalankan build. Namanya berisi nama proyek Anda dan nama yang Anda tentukan dalam file buildspec, dalam format ini:. `project-name-report-group-name` Untuk informasi selengkapnya, lihat Buat laporan pengujian dan Penamaan grup laporan.

3. Buat file bernama `project.json`. File ini berisi input untuk create-project perintah.
4. Salin JSON berikut ke`project.json`. Untuk`source`, masukkan jenis dan lokasi repositori yang berisi file sumber Anda. Untuk`serviceRole`, tentukan ARN peran yang Anda gunakan.

```
{
  "name": "test-report-project",
  "description": "sample-test-report-project",
  "source": {
    "type": "CODECOMMIT|CODEPIPELINE|GITHUB|S3|BITBUCKET|GITHUB_ENTERPRISE|
NO_SOURCE",
    "location": "<your-source-url>"
  },
  "artifacts": {
    "type": "NO_ARTIFACTS"
```

```
  },
  "cache": {
    "type": "NO_CACHE"
  },
  "environment": {
    "type": "LINUX_CONTAINER",
    "image": "aws/codebuild/standard:5.0",
    "computeType": "small"
  },
  "serviceRole": "arn:aws:iam::<your-aws-account-id>:role/service-role/<your-role-name>"
}
```

5. Jalankan perintah berikut di direktori yang berisi`project.json`. Ini menciptakan proyek bernama`test-project`.

```
aws codebuild create-project --cli-input-json file://project.json
```

Langkah 3: Jalankan dan melihat hasil laporan

Di bagian ini, Anda menjalankan build proyek yang Anda buat sebelumnya. Selama proses pembuatan CodeBuild , membuat laporan dengan hasil kasus uji. Laporan tersebut terkandung dalam grup laporan yang Anda tentukan.

1. Untuk memulai build, jalankan perintah berikut. `test-report-project`adalah nama proyek build yang dibuat di atas. Catat ID build yang muncul di output.

```
aws codebuild start-build --project-name test-report-project
```

2. Jalankan perintah berikut untuk mendapatkan informasi tentang build Anda, termasuk ARN laporan Anda. Untuk*<build-id>*, tentukan ID build Anda. Buat catatan laporan ARN di `reportArns` properti output.

```
aws codebuild batch-get-builds --ids <build-id>
```

3. Jalankan perintah berikut untuk mendapatkan detail tentang laporan Anda. Untuk*<report-arn>*, tentukan laporan ARN Anda.

```
aws codebuild batch-get-reports --report-arns <report-arn>
```

Output-nya akan terlihat seperti berikut. Output sampel ini menunjukkan berapa banyak pengujian yang berhasil, gagal, dilewati, mengakibatkan kesalahan, atau mengembalikan status yang tidak diketahui.

```
{
  "reports": [
    {
      "status": "FAILED",
      "reportGroupArn": "<report-group-arn>",
      "name": "<report-group-name>",
      "created": 1573324770.154,
      "exportConfig": {
        "exportConfigType": "S3",
        "s3Destination": {
          "bucket": "<amzn-s3-demo-bucket>",
          "path": "<path-to-your-report-results>",
          "packaging": "NONE",
          "encryptionKey": "<encryption-key>"
        }
      },
      "expired": 1575916770.0,
      "truncated": false,
      "executionId": "arn:aws:codebuild:us-west-2:123456789012:build/<name-of-
build-project>:2c254862-ddf6-4831-a53f-6839a73829c1",
      "type": "TEST",
      "arn": "<report-arn>",
      "testSummary": {
        "durationInNanoSeconds": 6657770,
        "total": 11,
        "statusCounts": {
          "FAILED": 3,
          "SKIPPED": 7,
          "ERROR": 0,
          "SUCCEEDED": 1,
          "UNKNOWN": 0
        }
      }
    }
  ],
  "reportsNotFound": []
}
```

4. Jalankan perintah berikut untuk mencantumkan informasi tentang kasus uji untuk laporan Anda. Untuk*<report-arn>*, tentukan ARN laporan Anda. Untuk `--filter` parameter opsional, Anda dapat menentukan satu hasil status (`SUCCEEDED`,`FAILED`,`SKIPPED`,`ERROR`, atau`UNKNOWN`).

```
aws codebuild describe-test-cases \
    --report-arn <report-arn> \
    --filter status=SUCCEEDED|FAILED|SKIPPED|ERROR|UNKNOWN
```

Output-nya akan terlihat seperti berikut.

```
{
  "testCases": [
    {
      "status": "FAILED",
      "name": "Test case 1",
      "expired": 1575916770.0,
      "reportArn": "<report-arn>",
      "prefix": "Cucumber tests for agent",
      "message": "A test message",
      "durationInNanoSeconds": 1540540,
      "testRawDataPath": "<path-to-output-report-files>"
    },
    {
      "status": "SUCCEEDED",
      "name": "Test case 2",
      "expired": 1575916770.0,
      "reportArn": "<report-arn>",
      "prefix": "Cucumber tests for agent",
      "message": "A test message",
      "durationInNanoSeconds": 1540540,
      "testRawDataPath": "<path-to-output-report-files>"
    }
  ]
}
```

Sampel Docker untuk CodeBuild

Bagian ini menjelaskan contoh integrasi antara Docker dan. AWS CodeBuild

Sampel	Deskripsi
Docker dalam sampel gambar khusus untuk CodeBuild	Sampel ini membangun dan menjalank an image Docker dengan menggunakan CodeBuild dan image build Docker kustom (`docker:dind` di Docker Hub).
Contoh server pembuatan gambar Docker untuk CodeBuild	Contoh ini membongkar build Docker Anda ke server build image terkelola.
Windows Docker membuat sampel untuk CodeBuild	Contoh ini membangun dan menjalankan image Windows Docker dengan menggunakan. CodeBuild
Contoh 'Publikasikan gambar Docker ke repositori gambar Amazon ECR' untuk CodeBuild	Sampel ini menghasilkan image Docker sebagai output build dan kemudian mendorong image Docker ke repositori image Amazon Elastic Container Registry (Amazon ECR).
Registri pribadi dengan AWS Secrets Manager sampel untuk CodeBuild	Contoh ini menunjukkan cara menggunakan image Docker yang disimpan dalam registri pribadi sebagai lingkungan CodeBuild runtime Anda.

Docker dalam sampel gambar khusus untuk CodeBuild

Contoh berikut membangun dan menjalankan image Docker dengan menggunakan AWS CodeBuild dan image build Docker kustom (`docker:dind`di Docker Hub).

Untuk mempelajari cara membuat image Docker dengan menggunakan image build yang disediakan oleh CodeBuild dukungan Docker, lihat. Contoh 'Publikasikan gambar Docker ke Amazon ECR'

> **Important**
>
> Menjalankan sampel ini dapat mengakibatkan biaya ke AWS akun Anda. Ini termasuk kemungkinan biaya untuk CodeBuild dan untuk AWS sumber daya dan tindakan yang terkait dengan Amazon S3, AWS KMS, dan CloudWatch Log. Untuk informasi selengkapnya, lihat

CodeBuild harga, harga Amazon S3, AWS Key Management Service harga, dan harga Amazon CloudWatch.

Topik

- Jalankan Docker dalam sampel gambar khusus

Jalankan Docker dalam sampel gambar khusus

Gunakan prosedur berikut untuk menjalankan Docker dalam sampel gambar khusus. Untuk informasi lebih lanjut tentang sampel ini, lihatDocker dalam sampel gambar khusus untuk CodeBuild.

Untuk menjalankan Docker dalam sampel gambar khusus

1. Buat file seperti yang dijelaskan di Berkas bagian Struktur direktori dan topik ini, lalu unggah ke bucket input S3 atau repositori AWS CodeCommit GitHub, atau Bitbucket.

   > **Important**
   >
   > Jangan mengunggah*(root directory name)*, hanya file di dalamnya*(root directory name)*.
   > Jika Anda menggunakan bucket input S3, pastikan untuk membuat file ZIP yang berisi file, lalu unggah ke bucket input. Jangan tambahkan *(root directory name)* ke file ZIP, hanya file di dalamnya*(root directory name)*.

2. Buat proyek build, jalankan build, dan lihat informasi build terkait.

   Jika Anda menggunakan AWS CLI untuk membuat proyek build, JSON-formatted input ke `create-project` perintah mungkin terlihat mirip dengan ini. (Ganti placeholder dengan nilai Anda sendiri.)

   ```
   {
     "name": "sample-docker-custom-image-project",
     "source": {
       "type": "S3",
       "location": "codebuild-region-ID-account-ID-input-
   bucket/DockerCustomImageSample.zip"
     },
     "artifacts": {
   ```

```
    "type": "NO_ARTIFACTS"
  },
  "environment": {
    "type": "LINUX_CONTAINER",
    "image": "docker:dind",
    "computeType": "BUILD_GENERAL1_SMALL",
    "privilegedMode": false
  },
  "serviceRole": "arn:aws:iam::account-ID:role/role-name",
  "encryptionKey": "arn:aws:kms:region-ID:account-ID:key/key-ID"
}
```

> **Note**
>
> Secara default, daemon Docker diaktifkan untuk build non-VPC. Jika Anda ingin menggunakan kontainer Docker untuk build VPC, lihat Runtime Privilege dan Kemampuan Linux di situs web Docker Docs dan aktifkan mode istimewa. Juga, Windows tidak mendukung mode istimewa.

3. Untuk melihat hasil build, lihat di log build untuk string`Hello, World!`. Untuk informasi selengkapnya, lihat Lihat detail build.

Struktur direktori

Sampel ini mengasumsikan struktur direktori ini.

```
(root directory name)
### buildspec.yml
### Dockerfile
```

Berkas

Gambar dasar dari sistem operasi yang digunakan dalam sampel ini adalah Ubuntu. Sampel menggunakan file-file ini.

`buildspec.yml`(dalam*(root directory name)*)

```
version: 0.2

phases:
  pre_build:
```

```
    commands:
      - docker build -t helloworld .
  build:
    commands:
      - docker images
      - docker run helloworld echo "Hello, World!"
```

`Dockerfile`(dalam*(root directory name)*)

```
FROM maven:3.3.9-jdk-8

RUN echo "Hello World"
```

Contoh server pembuatan gambar Docker untuk CodeBuild

Contoh berikut menurunkan build Docker Anda ke server build image terkelola. Anda dapat menyesuaikan sampel ini untuk menyediakan server build image Docker khusus dan terkelola dalam konfigurasi CodeBuild proyek Anda. Perhatikan bahwa instance yang disediakan aktif saat build dijalankan secara aktif untuk proyek, dan instance dihentikan saat build tidak berjalan. Instans yang disediakan disimpan hingga satu bulan sebelum didaur ulang. Untuk informasi selengkapnya, lihat CodeBuild Kapabilitas Server Docker.

> **Important**
>
> Menjalankan sampel ini dapat mengakibatkan tagihan ke AWS akun Anda. Ini termasuk kemungkinan biaya untuk CodeBuild dan untuk AWS sumber daya dan tindakan yang terkait dengan Amazon S3, AWS KMS, dan CloudWatch Log. Untuk informasi selengkapnya, lihat CodeBuild harga, harga Amazon S3, AWS Key Management Service harga, dan CloudWatch harga Amazon.

Topik

- Konfigurasikan server Docker

Konfigurasikan server Docker

Gunakan prosedur berikut untuk menyediakan lingkungan komputasi khusus untuk CodeBuild proyek yang mengelola beban kerja Docker dan menyimpan lapisan image Docker.

Untuk mengkonfigurasi server Docker

1. Buat file seperti yang dijelaskan di Berkas bagian Struktur direktori dan topik ini, lalu unggah ke bucket input S3 atau repositori AWS CodeCommit, GitHub, atau Bitbucket.

   > **Important**
   >
   > Jangan mengunggah*(root directory name)*, hanya file di dalamnya*(root directory name)*.
   > Jika Anda menggunakan bucket input S3, pastikan untuk membuat file ZIP yang berisi file tersebut, lalu unggah ke bucket input. Jangan tambahkan *(root directory name)* ke file ZIP, hanya file di dalamnya*(root directory name)*.

2. Buat proyek build, jalankan build, dan lihat informasi build terkait:

   a. Di bagian Lingkungan konsol, pilih Konfigurasi tambahan, navigasikan ke konfigurasi server Docker, lalu pilih Aktifkan server docker untuk proyek ini. Anda kemudian dapat memilih jenis komputasi server Docker dan menyediakan kre densi Registry.

   b. Jika Anda menggunakan AWS CLI untuk membuat proyek build, JSON-formatted input ke `create-project` perintah mungkin terlihat mirip dengan ini. (Ganti placeholder dengan nilai Anda sendiri.)

```
{
  "name": "sample-docker-custom-image-project",
  "source": {
    "type": "S3",
    "location": "codebuild-region-ID-account-ID-input-
bucket/DockerServerSample.zip"
  },
  "artifacts": {
    "type": "NO_ARTIFACTS"
  },
  "environment": {
    "type": "LINUX_CONTAINER",
    "image": "aws/codebuild/amazonlinux-x86_64-standard:5.0",
    "computeType": "BUILD_GENERAL1_LARGE",
    "dockerServer": [
         {
            "computeType": "BUILD_GENERAL1_LARGE",
            "securityGroupIds": [ "security-groups-ID" ]
         }
```

```
      ]
  },
  "serviceRole": "arn:aws:iam::account-ID:role/role-name"
}
```

> **Note**
>
> Grup keamanan yang dikonfigurasi untuk server Docker harus mengizinkan lalu lintas jaringan masuk dari VPC yang dikonfigurasi dalam proyek. Mereka harus mengizinkan masuknya di port 9876.

3. Untuk melihat hasil build, lihat string di log build`Hello, World!`. Untuk informasi selengkapnya, lihat Lihat detail build.

Struktur direktori

Contoh ini mengasumsikan struktur direktori ini.

```
(root directory name)
### buildspec.yml
### Dockerfile
```

Berkas

Gambar dasar dari sistem operasi yang digunakan dalam sampel ini adalah Ubuntu. Sampel menggunakan file-file ini.

`buildspec.yml`(dalam*(root directory name)*)

```
version: 0.2

phases:
  build:
    commands:
      - docker buildx build -t helloworld --load .
      - docker run helloworld echo "Hello, World!"
```

`Dockerfile`(dalam*(root directory name)*)

```
FROM public.ecr.aws/amazonlinux/amazonlinux:latest
```

```
RUN echo "Hello World"
```

Windows Docker membuat sampel untuk CodeBuild

Contoh berikut membangun dan menjalankan image Windows Docker dengan menggunakan. CodeBuild

Topik

- Jalankan sampel build Windows Docker

Jalankan sampel build Windows Docker

Gunakan prosedur berikut untuk menjalankan build Windows Docker.

Untuk menjalankan Windows Docker membangun sampel

1. Buat file seperti yang dijelaskan di Berkas bagian Struktur direktori dan topik ini, lalu unggah ke bucket input S3 atau repositori AWS CodeCommit GitHub, atau Bitbucket.

   > **Important**
   >
   > Jangan mengunggah*(root directory name)*, hanya file di dalamnya*(root directory name)*.
   > Jika Anda menggunakan bucket input S3, pastikan untuk membuat file ZIP yang berisi file, lalu unggah ke bucket input. Jangan tambahkan *(root directory name)* ke file ZIP, hanya file di dalamnya*(root directory name)*.

2. Buat `WINDOWS_EC2` armada.

   Jika Anda menggunakan AWS CLI untuk membuat armada, JSON-formatted input ke `create-fleet` perintah mungkin terlihat mirip dengan ini. (Ganti placeholder dengan nilai Anda sendiri.)

   ```
   {
     "name": "fleet-name",
     "baseCapacity": 1,
     "environmentType": "WINDOWS_EC2",
     "computeType": "BUILD_GENERAL1_MEDIUM"
   }
   ```

3. Buat proyek build, jalankan build, dan lihat informasi build terkait.

   Jika Anda menggunakan AWS CLI untuk membuat proyek build, JSON-formatted input ke `create-project` perintah mungkin terlihat mirip dengan ini. (Ganti placeholder dengan nilai Anda sendiri.)

   ```
   {
     "name": "project-name",
     "source": {
       "type": "S3",
       "location": "bucket-name/DockerImageSample.zip"
     },
     "artifacts": {
       "type": "NO_ARTIFACTS"
     },
     "environment": {
       "type": "WINDOWS_EC2",
       "image": "Windows",
       "computeType": "BUILD_GENERAL1_MEDIUM",
       "fleet": {
          "fleetArn": "fleet-arn"
       }
     },
     "serviceRole": "arn:aws:iam::account-ID:role/role-name"
   }
   ```

4. Untuk melihat hasil build, lihat di log build untuk string`Hello, World!`. Untuk informasi selengkapnya, lihat Lihat detail build.

Struktur direktori

Sampel ini mengasumsikan struktur direktori ini.

```
(root directory name)
### buildspec.yml
### Dockerfile
```

Berkas

Gambar dasar dari sistem operasi yang digunakan dalam sampel ini adalah`mcr.microsoft.com/windows/servercore:ltsc2022`. Sampel menggunakan file-file ini.

`buildspec.yml`(dalam*(root directory name)*)

```
version: 0.2

phases:
  pre_build:
    commands:
      - docker build -t helloworld .
  build:
    commands:
      - docker images
      - docker run helloworld powershell -Command "Write-Host 'Hello World!'"
```

`Dockerfile`(dalam*(root directory name)*)

```
FROM mcr.microsoft.com/windows/servercore:ltsc2022

RUN powershell -Command "Write-Host 'Hello World'"
```

Contoh 'Publikasikan gambar Docker ke repositori gambar Amazon ECR' untuk CodeBuild

Sampel ini menghasilkan gambar Docker sebagai output build dan kemudian mendorong gambar Docker ke repositori gambar Amazon Elastic Container Registry (Amazon ECR). Anda dapat menyesuaikan sampel ini untuk mendorong gambar Docker ke Docker Hub. Untuk informasi selengkapnya, lihat Sesuaikan sampel 'Publikasikan gambar Docker ke Amazon ECR' untuk didorong ke Docker Hub.

Untuk mempelajari cara membuat image Docker dengan menggunakan image build Docker kustom (`docker:dind`di Docker Hub), lihat kami. Docker dalam sampel gambar khusus

Sampel ini diuji dengan referensi. `golang:1.12`

Sampel ini menggunakan fitur build Docker multi-tahap baru, yang menghasilkan gambar Docker sebagai output build. Kemudian mendorong gambar Docker ke repositori gambar Amazon ECR. Multi-stage Pembuatan gambar Docker membantu mengurangi ukuran gambar Docker akhir. Untuk informasi selengkapnya, lihat Menggunakan build multi-tahap dengan Doc ker.

> **Important**
>
> Menjalankan sampel ini dapat mengakibatkan tagihan ke AWS akun Anda. Ini termasuk kemungkinan biaya untuk AWS CodeBuild dan untuk AWS sumber daya dan tindakan yang terkait dengan Amazon S3, AWS KMS, CloudWatch Logs, dan Amazon ECR. Untuk informasi selengkapnya, lihat CodeBuild harga, harga Amazon S3, harga, AWS Key Management Service harga Amazon, dan CloudWatch harga Amazon Elastic Container Registry.

Topik

- Jalankan contoh 'Publikasikan gambar Docker ke Amazon ECR'
- Sesuaikan sampel 'Publikasikan gambar Docker ke Amazon ECR' untuk didorong ke Docker Hub

Jalankan contoh 'Publikasikan gambar Docker ke Amazon ECR'

Gunakan prosedur berikut untuk menjalankan sampel yang menerbitkan gambar Docker ke Amazon ECR. Untuk informasi lebih lanjut tentang sampel ini, lihatContoh 'Publikasikan gambar Docker ke repositori gambar Amazon ECR' untuk CodeBuild.

Untuk menjalankan sampel ini

1. Jika Anda sudah memiliki repositori gambar di Amazon ECR yang ingin Anda gunakan, lewati ke langkah 3. Jika tidak, jika Anda menggunakan pengguna alih-alih akun AWS root atau pengguna administrator untuk bekerja dengan Amazon ECR, tambahkan pernyataan ini (antara *### BEGIN ADDING STATEMENT HERE ###* dan*### END ADDING STATEMENT HERE ###*) ke pengguna (atau grup IAM yang dikaitkan dengan pengguna). Menggunakan akun AWS root tidak disarankan. Pernyataan ini memungkinkan pembuatan repositori Amazon ECR untuk menyimpan gambar Docker. Ellipses (`...`) digunakan untuk singkatnya dan untuk membantu Anda menemukan tempat menambahkan pernyataan. Jangan menghapus pernyataan apa pun, dan jangan mengetikkan elips ini ke dalam kebijakan. Untuk informasi selengkapnya, lihat Bek erja dengan kebijakan sebaris menggunakan Konsol Manajemen AWS dalam Panduan pengguna.

   JSON

   ```
   {
   ```

```
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ecr:CreateRepository"
            ],
            "Resource": "*"
        }
    ]
}
```

> **Note**
>
> Entitas IAM yang mengubah kebijakan ini harus memiliki izin di IAM untuk mengubah kebijakan.

2. Buat repositori gambar di Amazon ECR. Pastikan untuk membuat repositori di Wil AWS ayah yang sama tempat Anda membuat lingkungan build dan menjalankan build Anda. Untuk informasi selengkapnya, lihat Membuat repositori di Panduan Pengguna Amazon ECR. Nama repositori ini harus sesuai dengan nama repositori yang Anda tentukan nanti dalam prosedur ini, diwakili oleh variabel `IMAGE_REPO_NAME` lingkungan. Pastikan kebijakan repositori Amazon ECR memberikan akses push gambar untuk peran IAM CodeBuild layanan Anda.
3. Tambahkan pernyataan ini (antara *### BEGIN ADDING STATEMENT HERE ###* dan*### END ADDING STATEMENT HERE ###*) ke kebijakan yang Anda lampirkan ke peran AWS CodeBuild layanan Anda. Pernyataan ini memungkinkan CodeBuild untuk mengunggah gambar Docker ke repositori Amazon ECR. Ellipses (`...`) digunakan untuk singkatnya dan untuk membantu Anda menemukan tempat menambahkan pernyataan. Jangan menghapus pernyataan apa pun, dan jangan mengetikkan elips ini ke dalam kebijakan.

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
```

```
                "ecr:BatchCheckLayerAvailability",
                "ecr:CompleteLayerUpload",
                "ecr:GetAuthorizationToken",
                "ecr:InitiateLayerUpload",
                "ecr:PutImage",
                "ecr:UploadLayerPart"
            ],
            "Resource": "*"
        }
    ]
}
```

> **Note**
>
> Entitas IAM yang mengubah kebijakan ini harus memiliki izin di IAM untuk mengubah kebijakan.

4. Buat file seperti yang dijelaskan di Berkas bagian Struktur direktori dan topik ini, lalu unggah ke bucket input S3 atau repositori AWS CodeCommit, GitHub, atau Bitbucket. Untuk informasi selengkapnya, lihat Referensi file definisi gambar di Panduan AWS CodePipeline Pengguna.

> **Important**
>
> Jangan mengunggah*(root directory name)*, hanya file di dalamnya*(root directory name)*.
> Jika Anda menggunakan bucket input S3, pastikan untuk membuat file ZIP yang berisi file tersebut, lalu unggah ke bucket input. Jangan tambahkan *(root directory name)* ke file ZIP, hanya file di dalamnya*(root directory name)*.

5. Buat proyek build, jalankan build, dan lihat informasi build.

   Jika Anda menggunakan konsol untuk membuat proyek Anda:

   a. Untuk Sistem operasi, pilih Ubuntu.

   b. Untuk Waktu aktif, pilih Standar.

   c. Untuk Gambar, pilih aws/codebuild /standard:5.0.

   d. Tambahkan variabel lingkungan berikut:

- AWS_DEFAULT_REGION dengan nilai *region-ID*
- AWS_ACCOUNT_ID dengan nilai *account-ID*
- IMAGE_TAG dengan nilai Terbaru
- IMAGE_REPO_NAME dengan nilai *Amazon-ECR-repo-name*

Jika Anda menggunakan AWS CLI untuk membuat proyek build, JSON-formatted input ke `create-project` perintah mungkin terlihat mirip dengan ini. (Ganti placeholder dengan nilai Anda sendiri.)

```
{
  "name": "sample-docker-project",
  "source": {
    "type": "S3",
    "location": "codebuild-region-ID-account-ID-input-bucket/DockerSample.zip"
  },
  "artifacts": {
    "type": "NO_ARTIFACTS"
  },
  "environment": {
    "type": "LINUX_CONTAINER",
    "image": "aws/codebuild/standard:5.0",
    "computeType": "BUILD_GENERAL1_SMALL",
    "environmentVariables": [
      {
        "name": "AWS_DEFAULT_REGION",
        "value": "region-ID"
      },
      {
        "name": "AWS_ACCOUNT_ID",
        "value": "account-ID"
      },
      {
        "name": "IMAGE_REPO_NAME",
        "value": "Amazon-ECR-repo-name"
      },
      {
        "name": "IMAGE_TAG",
        "value": "latest"
      }
    ],
```

```
  },
  "serviceRole": "arn:aws:iam::account-ID:role/role-name",
  "encryptionKey": "arn:aws:kms:region-ID:account-ID:key/key-ID"
}
```

6. Konfirmas CodeBuild ikan bahwa berhasil mendorong gambar Docker ke repositori:

   1. Buka konsol Amazon ECR di https://console.aws.amazon.com/ecr/.
   2. Pilih nama repositori. Gambar harus dicantumkan di kolom Tag gambar.

Struktur direktori

Contoh ini mengasumsikan struktur direktori ini.

```
(root directory name)
### buildspec.yml
### Dockerfile
```

Berkas

Sampel ini menggunakan file-file ini.

`buildspec.yml`(dalam*`(root directory name)`*)

```
version: 0.2

phases:
  pre_build:
    commands:
      - echo Logging in to Amazon ECR...
      - aws ecr get-login-password --region $AWS_DEFAULT_REGION | docker login --
username AWS --password-stdin $AWS_ACCOUNT_ID.dkr.ecr.$AWS_DEFAULT_REGION.amazonaws.com
  build:
    commands:
      - echo Build started on `date`
      - echo Building the Docker image...
      - docker build -t $IMAGE_REPO_NAME:$IMAGE_TAG .
      - docker tag $IMAGE_REPO_NAME:$IMAGE_TAG $AWS_ACCOUNT_ID.dkr.ecr.
$AWS_DEFAULT_REGION.amazonaws.com/$IMAGE_REPO_NAME:$IMAGE_TAG
  post_build:
    commands:
      - echo Build completed on `date`
```

```
      - echo Pushing the Docker image...
      - docker push $AWS_ACCOUNT_ID.dkr.ecr.$AWS_DEFAULT_REGION.amazonaws.com/
$IMAGE_REPO_NAME:$IMAGE_TAG
```

`Dockerfile`(dalam*`(root directory name)`*)

```
FROM golang:1.12-alpine AS build
#Install git
RUN apk add --no-cache git
#Get the hello world package from a GitHub repository
RUN go get github.com/golang/example/hello
WORKDIR /go/src/github.com/golang/example/hello
# Build the project and send the output to /bin/HelloWorld
RUN go build -o /bin/HelloWorld

FROM golang:1.12-alpine
#Copy the build's output binary from the previous build container
COPY --from=build /bin/HelloWorld /bin/HelloWorld
ENTRYPOINT ["/bin/HelloWorld"]
```

> **Note**
>
> CodeBuild mengesampingkan `ENTRYPOINT` untuk gambar Docker kustom.

Sesuaikan sampel 'Publikasikan gambar Docker ke Amazon ECR' untuk didorong ke Docker Hub

Untuk mengadaptasi sampel 'Publikasikan gambar Docker ke Amazon ECR' sehingga gambar Docker didorong ke Docker Hub alih-alih Amazon ECR, edit kode sampel. Untuk informasi lebih lanjut tentang sampel, lihat Contoh 'Publikasikan gambar Docker ke repositori gambar Amazon ECR' untuk CodeBuild danJalankan contoh 'Publikasikan gambar Docker ke Amazon ECR'.

> **Note**
>
> Jika Anda menggunakan versi Docker lebih awal dari 17.06, hapus opsi. `--no-include-email`

1. Ganti ECR-specific baris kode Amazon ini dalam `buildspec.yml` file:

```
...
  pre_build:
    commands:
      - echo Logging in to Amazon ECR...
      - aws ecr get-login-password --region $AWS_DEFAULT_REGION |
 docker login --username AWS --password-stdin $AWS_ACCOUNT_ID.dkr.ecr.
$AWS_DEFAULT_REGION.amazonaws.com
  build:
    commands:
      - echo Build started on `date`
      - echo Building the Docker image...
      - docker build -t $IMAGE_REPO_NAME:$IMAGE_TAG .
      - docker tag $IMAGE_REPO_NAME:$IMAGE_TAG $AWS_ACCOUNT_ID.dkr.ecr.
$AWS_DEFAULT_REGION.amazonaws.com/$IMAGE_REPO_NAME:$IMAGE_TAG
  post_build:
    commands:
      - echo Build completed on `date`
      - echo Pushing the Docker image...
      - docker push $AWS_ACCOUNT_ID.dkr.ecr.$AWS_DEFAULT_REGION.amazonaws.com/
$IMAGE_REPO_NAME:$IMAGE_TAG
...
```

Dengan baris kode Docker Hub-specific ini:

```
...
  pre_build:
    commands:
      - echo Logging in to Docker Hub...
      # Type the command to log in to your Docker Hub account here.
  build:
    commands:
      - echo Build started on `date`
      - echo Building the Docker image...
      - docker build -t $IMAGE_REPO_NAME:$IMAGE_TAG .
      - docker tag $IMAGE_REPO_NAME:$IMAGE_TAG $IMAGE_REPO_NAME:$IMAGE_TAG
  post_build:
    commands:
      - echo Build completed on `date`
      - echo Pushing the Docker image...
      - docker push $IMAGE_REPO_NAME:$IMAGE_TAG
...
```

2. Unggah kode yang diedit ke bucket input S3 atau repositori AWS CodeCommit GitHub,, atau Bitbucket.

> **Important**
>
> Jangan mengunggah*(root directory name)*, hanya file di dalamnya*(root directory name)*.
> Jika Anda menggunakan bucket input S3, pastikan untuk membuat file ZIP yang berisi file tersebut, lalu unggah ke bucket input. Jangan tambahkan *(root directory name)* ke file ZIP, hanya file di dalamnya*(root directory name)*.

3. Ganti baris kode ini dari JSON-formatted input ke `create-project` perintah:

```
...
    "environmentVariables": [
      {
        "name": "AWS_DEFAULT_REGION",
        "value": "region-ID"
      },
      {
        "name": "AWS_ACCOUNT_ID",
        "value": "account-ID"
      },
      {
        "name": "IMAGE_REPO_NAME",
        "value": "Amazon-ECR-repo-name"
      },
      {
        "name": "IMAGE_TAG",
        "value": "latest"
      }
    ]
...
```

Dengan baris kode ini:

```
...
    "environmentVariables": [
      {
        "name": "IMAGE_REPO_NAME",
        "value": "your-Docker-Hub-repo-name"
```

```
      },
      {
        "name": "IMAGE_TAG",
        "value": "latest"
      }
    ]
...
```

4. Buat lingkungan build, jalankan build, dan lihat informasi build terkait.
5. Konfirmas AWS CodeBuild ikan bahwa berhasil mendorong gambar Docker ke repositori. Masuk ke Docker Hub, buka repositori, dan pilih tab Tag. `latest`Tag harus berisi nilai Ter akhir Diperbarui yang sangat baru.

Registri pribadi dengan AWS Secrets Manager sampel untuk CodeBuild

Contoh ini menunjukkan cara menggunakan image Docker yang disimpan dalam registri pribadi sebagai lingkungan AWS CodeBuild runtime Anda. Kredensi untuk registri pribadi disimpan di. AWS Secrets Manager Registri pribadi apa pun berfungsi dengan CodeBuild. Contoh ini menggunakan Docker Hub.

> **Note**
>
> Rahasia terlihat oleh tindakan dan tidak disamarkan saat ditulis ke file.

Topik

- Persyaratan sampel registri pribadi
- Buat CodeBuild proyek dengan registri pribadi
- Konfigurasikan kredenal registri pribadi untuk pelari yang dihosting sendiri

Persyaratan sampel registri pribadi

Untuk menggunakan registri pribadi dengan AWS CodeBuild, Anda harus memiliki yang berikut:

- Rahasia Secrets Manager yang menyimpan kredensi Docker Hub Anda. Kredensialnya digunakan untuk mengakses repositori pribadi Anda.

> **Note**
>
> Anda akan dikenakan biaya untuk rahasia yang Anda buat.

- Repositori atau akun pribadi.
- Kebijakan IAM peran CodeBuild layanan yang memberikan akses ke rahasia Secrets Manager Anda.

Ikuti langkah-langkah ini untuk membuat sumber daya ini dan kemudian membuat proyek CodeBuild build menggunakan gambar Docker yang disimpan di registri pribadi Anda.

Buat CodeBuild proyek dengan registri pribadi

1. Untuk informasi tentang cara membuat repositori pribadi gratis, lihat Repositori di Docker Hub. Anda juga dapat menjalankan perintah berikut di terminal untuk menarik gambar, mendapatkan ID-nya, dan mendorongnya ke repositori baru.

   ```
   docker pull amazonlinux
   docker images amazonlinux --format {{.ID}}
   docker tag image-id your-username/repository-name:tag
   docker login
   docker push your-username/repository-name
   ```

2. Ikuti langkah-langkah di Buat AWS Secrets Manager rahasia di Panduan AWS Secrets Manager Pengguna.

   a. Pada langkah 3, di Pilih tipe rahasia, pilih Jenis rahasia lainnya.

   b. Secara Key/value berpasangan, buat satu pasangan nilai kunci untuk nama pengguna Docker Hub Anda dan satu pasangan nilai kunci untuk kata sandi Docker Hub Anda.

   c. Lanjutkan mengikuti langkah-langkah di Buat AWS Secrets Manager rahasia.

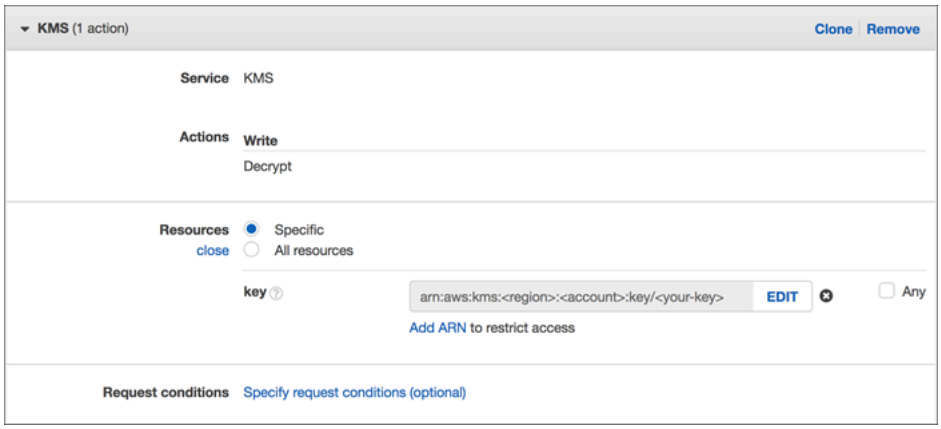
   d. Pada langkah 5, pada halaman Konfigurasi rotasi otomatis, matikan karena kunci sesuai dengan kredenal Docker Hub Anda.

   e. Selesai mengikuti langkah-langkah di Buat AWS Secrets Manager rahasia.

   Untuk informasi selengkapnya, lihat Apa itu AWS Secrets Manager?

3. Saat Anda membuat AWS CodeBuild proyek di konsol, CodeBuild lampirkan izin yang diperlukan untuk Anda. Jika Anda menggunakan AWS KMS kunci selain`DefaultEncryptionKey`, Anda harus menambahkannya ke peran layanan. Untuk informasi selengkapnya, lihat Memodifikasi peran (konsol) di Panduan Pengguna IAM.

   Agar peran layanan Anda berfungsi dengan Secrets Manager, setidaknya harus memiliki `secretsmanager:GetSecretValue` izin.



4. Untuk menggunakan konsol untuk membuat proyek dengan lingkungan yang disimpan dalam registri pribadi, lakukan hal berikut saat Anda membuat proyek. Untuk informasi, lihat Buat proyek build (konsol).

   > **Note**
   >
   > Jika registri pribadi Anda ada di VPC Anda, itu harus memiliki akses internet publik. CodeBuild tidak dapat menarik gambar dari alamat IP pribadi di VPC.

   a. Dalam gambar Lingkungan, pilih Gambar kustom.

   b. Untuk jenis Lingkungan, pilih Linux atau Windows.

   c. Untuk registri Gambar, pilih Registri lain.

   d. Di URL registri Eksternal, masukkan lokasi gambar dan kredensi Registri - opsional masukkan ARN atau nama kredenal Secrets Manager Anda.

> **Note**
>
> Jika kredensional Anda tidak ada di Wilayah Anda saat ini, maka Anda harus menggunakan ARN. Anda tidak dapat menggunakan nama kredensi jika kredensialnya ada di Wilayah yang berbeda.

Konfigurasikan kredenal registri pribadi untuk pelari yang dihosting sendiri

Gunakan petunjuk berikut untuk mengonfigurasi kredensi registri untuk pelari yang dihosting sendiri.

> **Note**
>
> Perhatikan bahwa kredensi ini hanya akan digunakan jika gambar diganti dengan yang berasal dari pendaftar pribadi.

AWS Management Console

1. Buka AWS CodeBuild konsol di https://console.aws.amazon.com/codesuite/codebuild/home.
2. Buat proyek build atau pilih proyek yang sudah ada. Untuk informasi selengkapnya, lihat Buat proyek build (konsol) dan Mengubah setelan proyek build (konsol).
3. Di Lingkungan, pilih Konfigurasi tambahan.
4. Dalam Konfigurasi tambahan, masukkan nama atau ARN rahasia dari AWS Secrets Manager untuk kredensi registri - opsional.



AWS CLI

1. Jika Anda ingin membuat proyek baru, jalankan perintah create-project.

```
aws codebuild create-project \
    --name project-name \
    --source type=source-type,location=source-location \
```

```
    --environment "type=environment-type,image=image,computeType=compute-
type,registryCredential={credentialProvider=SECRETS_MANAGER,credential=secret-
name-or-arn},imagePullCredentialsType=CODEBUILD|SERVICE_ROLE" \
    --artifacts type=artifacts-type \
    --service-role arn:aws:iam::account-ID:role/service-role/service-role-name
```

2. Jika Anda ingin memperbarui proyek yang sudah ada, jalankan perintah update-project.

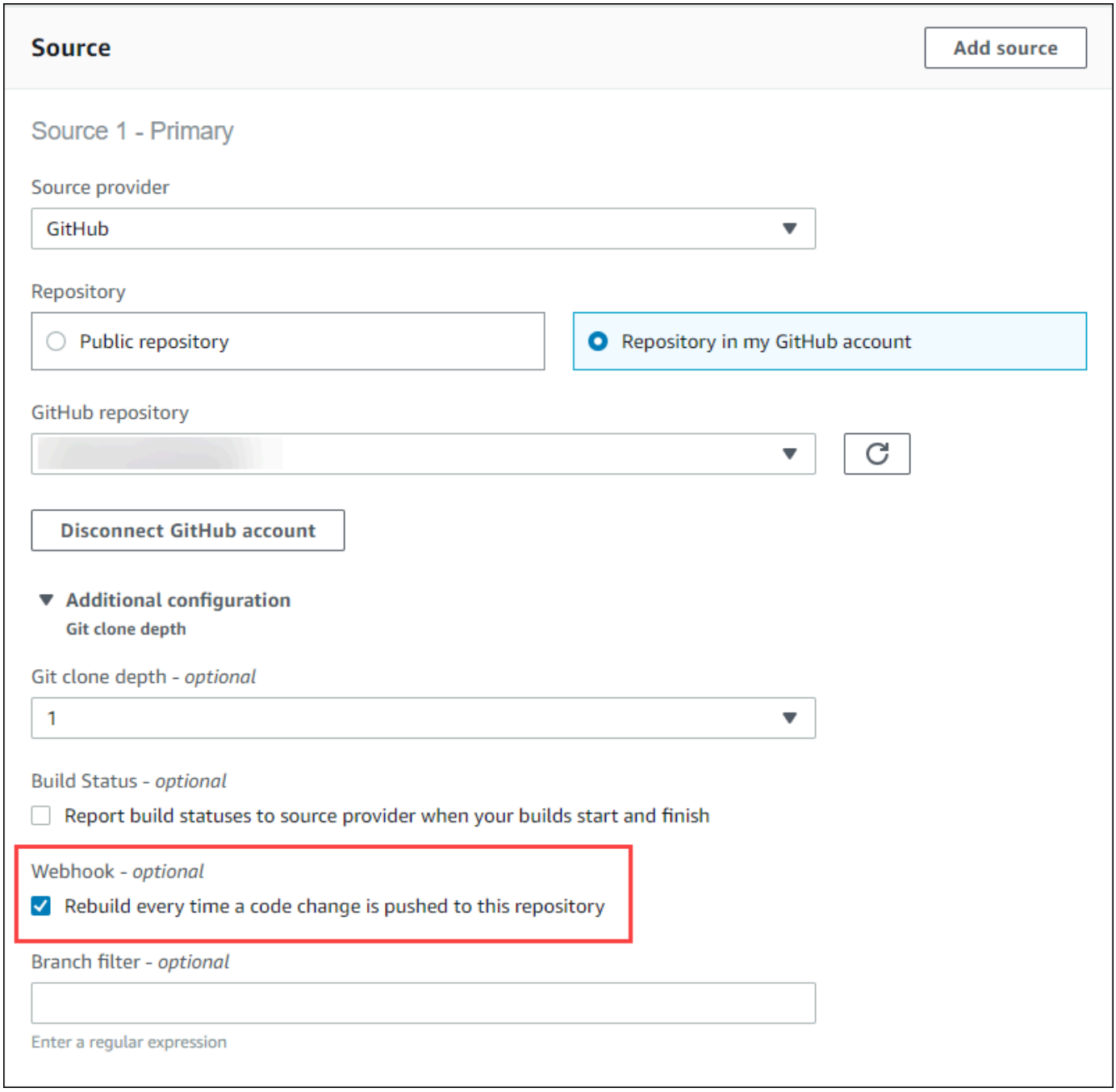
```
aws codebuild update-project \
    --name project-name \
    --environment "type=environment-type,image=image,computeType=compute-
type,registryCredential={credentialProvider=SECRETS_MANAGER,credential=secret-
name-or-arn}"
```

Buat situs web statis dengan keluaran build yang dihosting di bucket S3

Anda dapat menonaktifkan enkripsi artefak dalam build. Anda mungkin ingin melakukan ini sehingga Anda dapat mempublikasikan artefak ke lokasi yang dikonfigurasi untuk meng-host situs web. (Anda tidak dapat mempublikasikan artefak terenkripsi.) Contoh ini menunjukkan bagaimana Anda dapat menggunakan webhook untuk memicu build dan mempublikasikan artefaknya ke bucket S3 yang dikonfigurasi untuk menjadi situs web.

1. Ikuti petunjuk dalam Menyiapkan situs web statis untuk mengonfigurasi bucket S3 agar berfungsi seperti situs web.
2. Buka AWS CodeBuild konsol di https://console.aws.amazon.com/codesuite/codebuild/home.
3. Jika halaman CodeBuild informasi ditampilkan, pilih Buat proyek build. Jika tidak, pada panel navigasi, perluas Build, pilih Build projects, lalu pilih Create build project.
4. Di Nama proyek, masukkan nama untuk proyek pembangunan ini. Membangun nama proyek harus unik di setiap AWS akun. Anda juga dapat menyertakan deskripsi opsional proyek build untuk membantu pengguna lain memahami tujuan proyek ini.
5. Di Sumber, untuk penyedia Sumber, pilih GitHub. Ikuti petunjuk untuk menghubungkan (atau menyambung kembali) dengan GitHub, dan kemudian pilih Otorisasi.

   Untuk Webhook, pilih Rebuild setiap kali perubahan kode didorong ke repositori ini. Anda dapat memilih kotak centang ini hanya jika Anda memilih Gunakan repositori di akun saya.



6. Di Lingkungan:

   Untuk citra Lingkungan, lakukan salah satu hal berikut:

   - Untuk menggunakan image Docker yang dikelola oleh AWS CodeBuild, pilih Managed image, lalu buat pilihan dari Sistem operasi, Runtime, Image, dan versi Image. Buat pilihan dari jenis Lingkungan jika tersedia.

- Untuk menggunakan gambar Docker lain, pilih Custom image. Untuk jenis Lingkungan, pilih ARM, Linux, GPU Linux, atau Windows. Jika Anda memilih Registri lain, untuk URL registri eksternal, masukkan nama dan tag gambar Docker di Docker Hub, menggunakan format. *docker repository*/*docker image name* Jika Anda memilih Amazon ECR, gunakan repositori Amazon ECR dan gambar Amazon ECR untuk memilih gambar Docker di akun Anda. AWS
- Untuk menggunakan image Docker pribadi, pilih Custom image. Untuk jenis Lingkungan, pilih ARM, Linux, GPU Linux, atau Windows. Untuk registri Gambar, pilih Registri lain, lalu masukkan ARN kredensil untuk gambar Docker pribadi Anda. Kredensialnya harus dibuat oleh Secrets Manager. Untuk informasi lebih lanjut, lihat Apa itu AWS Secrets Manager? di Panduan Pengguna AWS Secrets Manager .

7. Dalam peran Layanan, lakukan salah satu hal berikut:

   - Jika Anda tidak memiliki peran CodeBuild layanan, pilih Peran layanan baru. Di Nama peran, masukkan nama untuk peran baru.
   - Jika Anda memiliki peran CodeBuild layanan, pilih Peran layanan yang ada. Di Peran ARN, pilih peran layanan.

   > **Note**
   >
   > Saat menggunakan konsol untuk membuat atau memperbarui proyek build, Anda dapat membuat peran CodeBuild layanan secara bersamaan. Secara default, peran hanya berfungsi dengan proyek build tersebut. Jika Anda menggunakan konsol untuk mengaitkan peran layanan ini dengan proyek build lain, peran akan diperbarui agar berfungsi dengan proyek build lainnya. Peran layanan dapat bekerja dengan hingga 10 proyek pembangunan.

8. Di Buildspec, lakukan salah satu hal berikut:

   - Pilih Gunakan file buildspec untuk menggunakan file buildspec.ymldi direktori root kode sumber.
   - Pilih Sisipkan perintah build untuk menggunakan konsol untuk menyisipkan perintah build.

   Untuk informasi selengkapnya, lihat Referensi Buildspec.

9. Di Artefak, untuk Jenis, pilih Amazon S3 untuk menyimpan output build dalam bucket S3.

10. Untuk nama Bucket, pilih nama bucket S3 yang Anda konfigurasikan agar berfungsi sebagai situs web pada langkah 1.
11. Jika Anda memilih Insert perintah build di Environment, lalu untuk file Output, masukkan lokasi file dari build yang ingin Anda masukkan ke dalam bucket keluaran. Jika Anda memiliki lebih dari satu lokasi, gunakan koma untuk memisahkan setiap lokasi (misalnya,**`appspec.yml, target/my-app.jar`**). Untuk informasi selengkapnya, lihat Artifacts reference-key in the buildspec file.
12. Pilih Nonaktifkan enkripsi artefak.
13. Perluas Konfigurasi tambahan dan pilih opsi yang sesuai.
14. Pilih Buat proyek build. Pada halaman proyek build, dalam riwayat Build, pilih Start build untuk menjalankan build.
15. (Opsional) Ikuti petunjuk di Contoh: Percepat situs web Anda dengan Amazon CloudFront di Panduan Pengembang Amazon S3.

Beberapa sumber input dan sampel artefak keluaran

Anda dapat membuat proyek AWS CodeBuild build dengan lebih dari satu sumber input dan lebih dari satu set artefak keluaran. Contoh ini menunjukkan kepada Anda cara mengatur proyek build yang:

- Menggunakan beberapa sumber dan repositori dari berbagai jenis.
- Menerbitkan artefak build ke beberapa bucket S3 dalam satu build.

Dalam contoh berikut, Anda membuat proyek build dan menggunakannya untuk menjalankan build. Sampel menggunakan file buildspec proyek build untuk menunjukkan cara menggabungkan lebih dari satu sumber dan membuat lebih dari satu set artefak.

Untuk mempelajari cara membuat pipeline yang menggunakan beberapa input sumber CodeBuild untuk membuat beberapa artefak keluaran, lihatContoh CodePipeline/CodeBuild integrasi dengan beberapa sumber input dan artefak keluaran.

Topik

- Buat proyek build dengan beberapa input dan output
- Membuat proyek build tanpa sumber

Buat proyek build dengan beberapa input dan output

Gunakan prosedur berikut untuk membuat proyek build dengan beberapa input dan output.

Untuk membuat proyek build dengan beberapa input dan output

1. Unggah sumber Anda ke satu atau beberapa bucket S3, CodeCommit GitHub, GitHub Enterprise Server, atau repositori Bitbucket.
2. Pilih sumber mana yang merupakan sumber utama. Ini adalah sumber yang mencari dan CodeBuild menjalankan file buildspec Anda.
3. Buat proyek pembangunan. Untuk informasi selengkapnya, lihat Membuat proyek pembangunan di AWS CodeBuild.
4. Buat proyek build Anda, jalankan build, dan dapatkan informasi tentang build.
5. Jika Anda menggunakan AWS CLI untuk membuat proyek build, JSON-formatted input ke `create-project` perintah mungkin terlihat mirip dengan berikut ini:

```
{
  "name": "sample-project",
  "source": {
    "type": "S3",
    "location": "<bucket/sample.zip>"
  },
  "secondarySources": [
    {
      "type": "CODECOMMIT",
      "location": "https://git-codecommit.us-west-2.amazonaws.com/v1/repos/repo",
      "sourceIdentifier": "source1"
    },
    {
      "type": "GITHUB",
      "location": "https://github.com/awslabs/aws-codebuild-jenkins-plugin",
      "sourceIdentifier": "source2"
    }
  ],
  "secondaryArtifacts": [ss
    {
      "type": "S3",
      "location": "<output-bucket>",
      "artifactIdentifier": "artifact1"
    },
    {
```

```
      "type": "S3",
      "location": "<other-output-bucket>",
      "artifactIdentifier": "artifact2"
    }
  ],
  "environment": {
    "type": "LINUX_CONTAINER",
    "image": "aws/codebuild/standard:5.0",
    "computeType": "BUILD_GENERAL1_SMALL"
  },
  "serviceRole": "arn:aws:iam::account-ID:role/role-name",
  "encryptionKey": "arn:aws:kms:region-ID:account-ID:key/key-ID"
}
```

Sumber utama Anda ditentukan di bawah `source` atribut. Semua sumber lain disebut sumber sekunder dan muncul di bawah`secondarySources`. Semua sumber sekunder diinstal di direktori mereka sendiri. Direktori ini disimpan dalam variabel lingkungan bawaan`CODEBUILD_SRC_DIR_`*`sourceIdentifer`*. Untuk informasi selengkapnya, lihat Variabel lingkungan di lingkungan build.

A `secondaryArtifacts` tribut berisi daftar definisi artefak. Artefak ini menggunakan `secondary-artifacts` blok file buildspec yang bersarang di dalam blok. `artifacts`

Artefak sekunder dalam file buildspec memiliki struktur yang sama dengan artefak dan dipisahkan oleh pengidentifikasi artefaknya.

> **Note**
>
> Di CodeBuild API, artefak sekunder adalah atribut yang diperlukan di `CreateProject` dan`UpdateProject`. `artifactIdentifier` Ini harus digunakan untuk merujuk artefak sekunder.

Menggunakan JSON-formatted input sebelumnya, file buildspec untuk proyek mungkin terlihat seperti:

```
version: 0.2

phases:
  install:
```

```
    runtime-versions:
      java: openjdk11
  build:
    commands:
      - cd $CODEBUILD_SRC_DIR_source1
      - touch file1
      - cd $CODEBUILD_SRC_DIR_source2
      - touch file2

artifacts:
  files:
    - '**.*'
  secondary-artifacts:
    artifact1:
      base-directory: $CODEBUILD_SRC_DIR_source1
      files:
        - file1
    artifact2:
      base-directory: $CODEBUILD_SRC_DIR_source2
      files:
        - file2
```

Anda dapat mengganti versi sumber utama menggunakan API dengan `sourceVersion` atribut di`StartBuild`. Untuk mengganti satu atau lebih versi sumber sekunder, gunakan `secondarySourceVersionOverride` atribut.

In JSON-formatted put ke `start-build` perintah di AWS CLI mungkin terlihat seperti:

```
{
   "projectName": "sample-project",
   "secondarySourcesVersionOverride": [
      {
        "sourceIdentifier": "source1",
        "sourceVersion": "codecommit-branch"
      },
      {
        "sourceIdentifier": "source2",
        "sourceVersion": "github-branch"
      },
   ]
}
```

Membuat proyek build tanpa sumber

Anda dapat mengonfigurasi CodeBuild proyek dengan memilih jenis **`NO_SOURCE`** sumber saat Anda mengonfigurasi sumber Anda. Ketika jenis sumber Anda adalah**`NO_SOURCE`**, Anda tidak dapat menentukan file buildspec karena proyek Anda tidak memiliki sumber. Sebagai gantinya, Anda harus menentukan string YAML-formatted buildspec dalam `buildspec` atribut JSON-formatted input ke perintah CLI. `create-project` Manajer tersebut mungkin akan terlihat seperti ini:

```
{
  "name": "project-name",
  "source": {
    "type": "NO_SOURCE",
    "buildspec": "version: 0.2\n\nphases:\n  build:\n    commands:\n      - command"
   },
  "environment": {
    "type": "LINUX_CONTAINER",
    "image": "aws/codebuild/standard:5.0",
    "computeType": "BUILD_GENERAL1_SMALL",
  },
  "serviceRole": "arn:aws:iam::account-ID:role/role-name",
  "encryptionKey": "arn:aws:kms:region-ID:account-ID:key/key-ID"
}
```

Lihat informasi yang lebih lengkap di Buat proyek build (AWS CLI).

Versi runtime dalam contoh file buildspec untuk CodeBuild

Jika Anda menggunakan gambar standar Amazon Linux 2 (AL2) versi 1.0 atau yang lebih baru, atau gambar standar Ubuntu versi 2.0 atau yang lebih baru, Anda dapat menentukan satu atau beberapa runtime di `runtime-versions` bagian file buildspec Anda. Contoh berikut menunjukkan bagaimana Anda dapat mengubah runtime project, menentukan lebih dari satu runtime, dan menentukan runtime yang bergantung pada runtime lain. Untuk informasi tentang runtime yang didukung, lihatGambar Docker disediakan oleh CodeBuild.

> **Note**
>
> Jika Anda menggunakan Docker di container build, build Anda harus berjalan dalam mode istimewa. Untuk informasi selengkapnya, lihat Jalankan AWS CodeBuild build secara manual dan Membuat proyek pembangunan di AWS CodeBuild.

Topik

- Perbarui versi runtime dalam file buildspec
- Tentukan dua runtime

Perbarui versi runtime dalam file buildspec

Anda dapat mengubah runtime yang digunakan oleh proyek Anda ke versi baru dengan memperbarui `runtime-versions` bagian file buildspec Anda. Contoh berikut menunjukkan cara menentukan java versi 8 dan 11.

- `runtime-versions`Bagian yang menentukan versi 8 dari Java:

```
phases:
  install:
    runtime-versions:
      java: corretto8
```

- `runtime-versions`Bagian yang menentukan versi 11 dari Java:

```
phases:
  install:
    runtime-versions:
      java: corretto11
```

Contoh berikut menunjukkan cara menentukan versi Python yang berbeda menggunakan gambar standar Ubuntu 5.0 atau gambar standar Amazon Linux 2 3.0:

- `runtime-versions`Bagian yang menentukan Python versi 3.7:

```
phases:
  install:
    runtime-versions:
      python: 3.7
```

- `runtime-versions`Bagian yang menentukan Python versi 3.8:

```
phases:
  install:
    runtime-versions:
```

```
      python: 3.8
```

Contoh ini menunjukkan proyek yang dimulai dengan runtime Java versi 8, dan kemudian diperbarui ke runtime Java versi 10.

1. Unduh dan instal Maven. Untuk informasi, lihat Mengunduh Apache Maven dan Menginstal Apache Maven di situs web Apache Maven.
2. Beralih ke direktori kosong di komputer atau instance lokal Anda, lalu jalankan perintah Maven ini.

   ```
   mvn archetype:generate "-DgroupId=com.mycompany.app" "-DartifactId=ROOT" "-
   DarchetypeArtifactId=maven-archetype-webapp" "-DinteractiveMode=false"
   ```

   Jika berhasil, struktur direktori dan file ini dibuat.

   ```
   .
   ### ROOT
       ### pom.xml
       ### src
           ### main
               ### resources
               ### webapp
                   ### WEB-INF
                   #   ### web.xml
                   ### index.jsp
   ```

3. Buat file bernama `buildspec.yml` dengan isi berikut ini. Simpan file di *(root directory name)*`/my-web-app` direktori.

   ```
   version: 0.2

   phases:
     install:
       runtime-versions:
         java: corretto8
     build:
       commands:
         - java -version
         - mvn package
   artifacts:
   ```

```
  files:
    - '**/*'
  base-directory: 'target/my-web-app'
```

Dalam file buildspec:

- `runtime-versions`Bagian ini menentukan bahwa proyek menggunakan versi 8 dari runtime Java.
- `- java -version`Perintah menampilkan versi Java yang digunakan oleh proyek Anda saat dibangun.

Struktur file Anda sekarang akan terlihat seperti ini.

```
(root directory name)
### my-web-app
    ### src
    #   ### main
    #   ### resources
    #   ### webapp
    #       ### WEB-INF
    #           ### web.xml
    #               ### index.jsp
    ### buildspec.yml
    ### pom.xml
```

4. Unggah isi `my-web-app` direktori ke bucket input S3 atau repositori CodeCommit, GitHub, atau Bitbucket.

> **Important**
>
> Jangan mengunggah *(root directory name)* atau*(root directory name)*/my-web-app, hanya direktori dan file di*(root directory name)*/my-web-app.
> Jika Anda menggunakan bucket input S3, pastikan untuk membuat file ZIP yang berisi struktur direktori dan file, lalu unggah ke bucket input. Jangan menambahkan *(root directory name)* atau *(root directory name)*/my-web-app ke file ZIP, hanya direktori dan file di*(root directory name)*/my-web-app.

5. Buka AWS CodeBuild konsol di https://console.aws.amazon.com/codesuite/codebuild/home.

6. Buat proyek build. Untuk informasi selengkapnya, lihat Buat proyek build (konsol) dan Jalankan build (konsol). Biarkan semua pengaturan pada nilai default mereka, kecuali untuk pengaturan ini.

   - Untuk Lingkungan:
     - Untuk gambar Lingkungan, pilih Gambar terkelola.
     - Untuk sistem operasi, pilih Amazon Linux 2.
     - Untuk Runtime, pilih Standar.
     - Untuk Gambar, pilih aws/codebuild/amazonlinux-x86_64-standard:4.0.
7. Pilih Mulai membangun.
8. Pada konfigurasi Build, terima default, lalu pilih Start build.
9. Setelah build selesai, lihat output build di tab Build logs. Anda akan melihat output yang serupa dengan yang berikut:

```
[Container] Date Time Phase is DOWNLOAD_SOURCE
[Container] Date Time CODEBUILD_SRC_DIR=/codebuild/output/src460614277/src
[Container] Date Time YAML location is /codebuild/output/src460614277/src/
buildspec.yml
[Container] Date Time Processing environment variables
[Container] Date Time Selecting 'java' runtime version 'corretto8' based on manual
 selections...
[Container] Date Time Running command echo "Installing Java version 8 ..."
Installing Java version 8 ...

[Container] Date Time Running command export JAVA_HOME="$JAVA_8_HOME"

[Container] Date Time Running command export JRE_HOME="$JRE_8_HOME"

[Container] Date Time Running command export JDK_HOME="$JDK_8_HOME"

[Container] Date Time Running command for tool_path in "$JAVA_8_HOME"/bin/*
 "$JRE_8_HOME"/bin/*;
```

10. Perbarui `runtime-versions` bagian dengan Java versi 11:

```
install:
    runtime-versions:
      java: corretto11
```

11. Setelah Anda menyimpan perubahan, jalankan build Anda lagi dan lihat output build. Anda akan melihat bahwa versi Java yang diinstal adalah 11. Anda akan melihat output yang serupa dengan yang berikut:

```
[Container] Date Time Phase is DOWNLOAD_SOURCE
[Container] Date Time CODEBUILD_SRC_DIR=/codebuild/output/src460614277/src
[Container] Date Time YAML location is /codebuild/output/src460614277/src/
buildspec.yml
[Container] Date Time Processing environment variables
[Container] Date Time Selecting 'java' runtime version 'corretto11' based on manual
 selections...
Installing Java version 11 ...

[Container] Date Time Running command export JAVA_HOME="$JAVA_11_HOME"

[Container] Date Time Running command export JRE_HOME="$JRE_11_HOME"

[Container] Date Time Running command export JDK_HOME="$JDK_11_HOME"

[Container] Date Time Running command for tool_path in "$JAVA_11_HOME"/bin/*
 "$JRE_11_HOME"/bin/*;
```

Tentukan dua runtime

Anda dapat menentukan lebih dari satu runtime dalam proyek CodeBuild build yang sama. Proyek contoh ini menggunakan dua file sumber: satu yang menggunakan runtime Go dan satu yang menggunakan Node.js runtime.

1. Membuat sebuah direktori bernama `my-source`.
2. Di dalam `my-source` direktori, buat direktori bernama`golang-app`.
3. Buat file bernama `hello.go` dengan isi berikut ini. Simpan file di `golang-app` direktori.

```
package main
import "fmt"

func main() {
  fmt.Println("hello world from golang")
  fmt.Println("1+1 =", 1+1)
  fmt.Println("7.0/3.0 =", 7.0/3.0)
  fmt.Println(true && false)
```

```
  fmt.Println(true || false)
  fmt.Println(!true)
  fmt.Println("good bye from golang")
}
```

4. Di dalam `my-source` direktori, buat direktori bernama`nodejs-app`. Itu harus pada tingkat yang sama dengan `golang-app` direktori.
5. Buat file bernama `index.js` dengan isi berikut ini. Simpan file di `nodejs-app` direktori.

```
console.log("hello world from nodejs");
console.log("1+1 =" + (1+1));
console.log("7.0/3.0 =" + 7.0/3.0);
console.log(true && false);
console.log(true || false);
console.log(!true);
console.log("good bye from nodejs");
```

6. Buat file bernama `package.json` dengan isi berikut ini. Simpan file di `nodejs-app` direktori.

```
{
  "name": "mycompany-app",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"run some tests here\""
  },
  "author": "",
  "license": "ISC"
}
```

7. Buat file bernama `buildspec.yml` dengan isi berikut ini. Simpan file di `my-source` direktori, pada tingkat yang sama dengan `golang-app` direktori `nodejs-app` dan. `runtime-versions`Bagian ini menentukan runtime Node.js versi 12 dan Go versi 1.13.

```
version: 0.2

phases:
  install:
    runtime-versions:
      golang: 1.13
      nodejs: 12
```

```
  build:
    commands:
      - echo Building the Go code...
      - cd $CODEBUILD_SRC_DIR/golang-app
      - go build hello.go
      - echo Building the Node code...
      - cd $CODEBUILD_SRC_DIR/nodejs-app
      - npm run test
artifacts:
  secondary-artifacts:
    golang_artifacts:
      base-directory: golang-app
      files:
        - hello
    nodejs_artifacts:
      base-directory: nodejs-app
      files:
        - index.js
        - package.json
```

8. Struktur file Anda sekarang akan terlihat seperti ini.

```
my-source
### golang-app
#   ### hello.go
### nodejs.app
#   ### index.js
#   ### package.json
### buildspec.yml
```

9. Unggah isi `my-source` direktori ke bucket input S3 atau repositori CodeCommit, GitHub, atau Bitbucket.

> **Important**
>
> Jika Anda menggunakan bucket input S3, pastikan untuk membuat file ZIP yang berisi struktur direktori dan file, lalu unggah ke bucket input. Jangan tambahkan `my-source` ke file ZIP, hanya direktori dan file di`my-source`.

10. Buka AWS CodeBuild konsol di https://console.aws.amazon.com/codesuite/codebuild/home.

11. Buat proyek build. Untuk informasi selengkapnya, lihat Buat proyek build (konsol) dan Jalankan build (konsol). Biarkan semua pengaturan pada nilai default mereka, kecuali untuk pengaturan ini.

    - Untuk Lingkungan:
      - Untuk gambar Lingkungan, pilih Gambar terkelola.
      - Untuk sistem operasi, pilih Amazon Linux 2.
      - Untuk Runtime, pilih Standar.
      - Untuk Gambar, pilih aws/codebuild/amazonlinux-x86_64-standard:4.0.
12. Pilih Buat proyek build.
13. Pilih Mulai membangun.
14. Pada konfigurasi Build, terima default, lalu pilih Start build.
15. Setelah build selesai, lihat output build di tab Build logs. Anda akan melihat output seperti yang berikut ini. Ini menunjukkan output dari Go dan Node.js runtime. Ini juga menunjukkan output dari Go dan Node.js aplikasi.

```
[Container] Date Time Processing environment variables
[Container] Date Time Selecting 'golang' runtime version '1.13' based on manual
 selections...
[Container] Date Time Selecting 'nodejs' runtime version '12' based on manual
 selections...
[Container] Date Time Running command echo "Installing Go version 1.13 ..."
Installing Go version 1.13 ...

[Container] Date Time Running command echo "Installing Node.js version 12 ..."
Installing Node.js version 12 ...

[Container] Date Time Running command n $NODE_12_VERSION
   installed : v12.20.1 (with npm 6.14.10)

[Container] Date Time Moving to directory /codebuild/output/src819694850/src
[Container] Date Time Registering with agent
[Container] Date Time Phases found in YAML: 2
[Container] Date Time  INSTALL: 0 commands
[Container] Date Time  BUILD: 1 commands
[Container] Date Time Phase complete: DOWNLOAD_SOURCE State: SUCCEEDED
[Container] Date Time Phase context status code:  Message:
[Container] Date Time Entering phase INSTALL
[Container] Date Time Phase complete: INSTALL State: SUCCEEDED
```

```
[Container] Date Time Phase context status code:  Message:
[Container] Date Time Entering phase PRE_BUILD
[Container] Date Time Phase complete: PRE_BUILD State: SUCCEEDED
[Container] Date Time Phase context status code:  Message:
[Container] Date Time Entering phase BUILD
[Container] Date Time Running command echo Building the Go code...
Building the Go code...

[Container] Date Time Running command cd $CODEBUILD_SRC_DIR/golang-app

[Container] Date Time Running command go build hello.go

[Container] Date Time Running command echo Building the Node code...
Building the Node code...

[Container] Date Time Running command cd $CODEBUILD_SRC_DIR/nodejs-app

[Container] Date Time Running command npm run test

> mycompany-app@1.0.0 test /codebuild/output/src924084119/src/nodejs-app
> echo "run some tests here"

run some tests here
```

Contoh versi sumber dengan AWS CodeBuild

Contoh ini menunjukkan cara menentukan versi sumber Anda menggunakan format selain ID komit (juga dikenal sebagai komit SHA). Anda dapat menentukan versi sumber Anda dengan cara berikut:

- Untuk penyedia sumber Amazon S3, gunakan ID versi objek yang merepresentasikan file ZIP input build.
- Untuk CodeCommit, Bitbucket, GitHub, dan GitHub Enterprise Server, gunakan salah satu dari berikut ini:
  - Permintaan pull sebagai referensi permintaan pull (misalnya, `refs/pull/1/head`).
  - Cabang sebagai nama cabang.
  - ID commit.
  - Tanda.
  - Referensi dan ID commit. Referensi dapat berupa salah satu dari berikut:

    - Tanda (misalnya, `refs/tags/mytagv1.0^{full-commit-SHA}`).
    - Cabang (misalnya, `refs/heads/mydevbranch^{full-commit-SHA}`).
    - Permintaan pull (misalnya, `refs/pull/1/head^{full-commit-SHA}`).
- Untuk GitLab dan GitLab Self Managed, gunakan salah satu dari berikut ini:
    - Cabang sebagai nama cabang.
    - ID commit.
    - Tanda.

> **Note**
>
> Anda dapat menentukan versi sumber permintaan tarik hanya jika repositori Anda GitHub atau Server GitHub Perusahaan.

Jika Anda menggunakan referensi dan ID komit untuk menentukan versi, `DOWNLOAD_SOURCE` fase build akan lebih cepat daripada jika Anda hanya menyediakan versi saja. Ini karena ketika Anda menambahkan referensi, CodeBuild tidak perlu mengunduh seluruh repositori untuk menemukan komit.

- Anda dapat menentukan versi sumber hanya dengan ID komit, seperti`12345678901234567890123467890123456789`. Jika Anda melakukan ini, CodeBuild harus men-download seluruh repositori untuk menemukan versi.
- Anda dapat menentukan versi sumber dengan referensi dan ID komit dalam format ini: *`refs`*`/`*`heads`*`/`*`branchname`*`^{`*`full-commit-SHA`*`}` (misalnya,`refs/heads/main^{12345678901234567890123467890123456789}`). Jika Anda melakukan ini, CodeBuild unduh hanya cabang yang ditentukan untuk menemukan versinya.

> **Note**
>
> Untuk mempercepat `DOWNLOAD_SOURCE` fase build Anda, Anda juga dapat menyetel kedalaman klon Git ke angka yang rendah. CodeBuild mengunduh lebih sedikit versi repositori Anda.

Topik

- Tentukan versi GitHub repositori dengan ID komit
- Tentukan versi GitHub repositori dengan referensi dan ID komit

Tentukan versi GitHub repositori dengan ID komit

Anda dapat menentukan versi sumber hanya dengan ID komit, seperti`12345678901234567890123467890123456789`. Jika Anda melakukan ini, CodeBuild harus men-download seluruh repositori untuk menemukan versi.

Untuk menentukan versi GitHub repositori dengan ID komit

1. Buka AWS CodeBuild konsol di https://console.aws.amazon.com/codesuite/codebuild/home.
2. Buat proyek build. Untuk informasi selengkapnya, lihat Buat proyek build (konsol) dan Jalankan build (konsol). Biarkan semua pengaturan pada nilai defaultnya, kecuali untuk pengaturan ini:
   - Dalam Sumber:
     - Untuk penyedia Sumber, pilih GitHub. Jika Anda tidak terhubung GitHub, ikuti instruksi untuk menghubungkan.
     - Untuk Repositori, pilih Repositori publik.
     - Untuk URL Repositori, masukkan. **`https://github.com/aws/aws-sdk-ruby.git`**
   - Di Lingkungan:
     - Untuk gambar Lingkungan, pilih Gambar terkelola.
     - Untuk sistem operasi, pilih Amazon Linux 2.
     - Untuk Runtime, pilih Standar.
     - Untuk Gambar, pilih aws/codebuild/amazonlinux-x86_64-standard:4.0.
3. Untuk spesifikasi Build, pilih Insert build command, lalu pilih Switch to editor.
4. Dalam perintah Build, ganti teks placeholder dengan yang berikut ini:

```
version: 0.2

phases:
  install:
    runtime-versions:
      ruby: 2.6
  build:
    commands:
```

```
        - echo $CODEBUILD_RESOLVED_SOURCE_VERSION
```

`runtime-versions`Bagian ini diperlukan saat Anda menggunakan gambar standar Ubuntu 2.0. Di sini, runtime Ruby versi 2.6 ditentukan, tetapi Anda dapat menggunakan runtime apa pun. `echo`Perintah menampilkan versi kode sumber yang disimpan dalam variabel `CODEBUILD_RESOLVED_SOURCE_VERSION` lingkungan.

5. Pada konfigurasi Build, terima defaultnya, lalu pilih Start build.
6. Untuk versi Sumber, masukkan**`046e8b67481d53bdc86c3f6affdd5d1afae6d369`**. Ini adalah SHA dari komit di `https://github.com/aws/aws-sdk-ruby.git` repositori.
7. Pilih Mulai membangun.
8. Ketika build selesai, Anda akan melihat yang berikut:

   - Pada tab Build logs, versi sumber proyek mana yang digunakan. Inilah contohnya.

     ```
     [Container] Date Time Running command echo $CODEBUILD_RESOLVED_SOURCE_VERSION
     046e8b67481d53bdc86c3f6affdd5d1afae6d369

     [Container] Date Time Phase complete: BUILD State: SUCCEEDED
     ```

   - Pada tab variabel Lingkungan, versi sumber Terselesaikan cocok dengan ID komit yang digunakan untuk membuat build.
   - Pada tab Detail fase, durasi `DOWNLOAD_SOURCE` fase.

Tentukan versi GitHub repositori dengan referensi dan ID komit

Anda dapat menentukan versi sumber dengan referensi dan ID komit dalam format ini: *`refs/heads/branchname`*`^{`*`full-commit-SHA`*`}` (misalnya,`refs/heads/main^{1234567890123456789012346789012345678 9}`). Jika Anda melakukan ini, CodeBuild unduh hanya cabang yang ditentukan untuk menemukan versinya.

Untuk menentukan versi GitHub repositori dengan referensi dan komit ID.

1. Selesaikan langkah-langkah dalam Tentukan versi GitHub repositori dengan ID komit.
2. Dari panel navigasi kiri, pilih Membangun proyek, lalu pilih proyek yang Anda buat sebelumnya.
3. Pilih Mulai membangun.

4. Dalam versi Sumber, masukkan**`refs/heads/main^{046e8b67481d53bdc86c3f6affdd5d1afae6d369}`**. Ini adalah ID komit yang sama dan referensi ke cabang dalam format*`refs/heads/branchname^{full-commit-SHA}`*.
5. Pilih Mulai membangun.
6. Ketika build selesai, Anda akan melihat yang berikut:
   - Pada tab Build logs, versi sumber proyek mana yang digunakan. Inilah contohnya.

     ```
     [Container] Date Time Running command echo $CODEBUILD_RESOLVED_SOURCE_VERSION
     046e8b67481d53bdc86c3f6affdd5d1afae6d369

     [Container] Date Time Phase complete: BUILD State: SUCCEEDED
     ```

   - Pada tab variabel Lingkungan, versi sumber Terselesaikan cocok dengan ID komit yang digunakan untuk membuat build.
   - Pada tab Detail fase, durasi `DOWNLOAD_SOURCE` fase harus lebih pendek dari durasi saat Anda hanya menggunakan ID komit untuk menentukan versi sumber Anda.

Third-party sampel repositori sumber untuk CodeBuild

Bagian ini menjelaskan contoh integrasi antara repositori sumber pihak ketiga dan. CodeBuild

Sampel	Deskripsi
BitBucket permintaan tarik dan sampel filter webhook - lihat Jalankan sampel 'Bitbucket pull request dan webhook filter' untuk CodeBuild	Contoh ini menunjukkan cara membuat permintaan tarik menggunakan repositor i Bitbucket. Ini juga menunjukkan kepada Anda cara menggunakan webhook Bitbucket CodeBuild untuk memicu pembuatan proyek.
GitHub Contoh Server Perusahaan - lihat Jalan GitHub kan sampel Server Perusahaan untuk CodeBuild	Contoh ini menunjukkan cara menyiapkan CodeBuild proyek ketika repositori GitHub Enterprise Server Anda memiliki sertifikat yang diinstal. Ini juga menunjukkan cara mengaktif kan webhook sehingga CodeBuild membangun kembali kode sumber setiap kali perubahan

Sampel	Deskripsi
	kode didorong ke repositori Server Perusahaan Anda GitHub .
GitHub permintaan tarik dan sampel filter webhook - lihat Jalankan permintaan GitHub tarik dan sampel filter webhook untuk CodeBuild	Contoh ini menunjukkan cara membuat permintaan tarik menggunakan repositori GitHub Enterprise Server. Ini juga menunjukk an cara mengaktifkan webhook sehingga CodeBuild membangun kembali kode sumber setiap kali perubahan kode didorong ke repositori Server Perusahaan Anda GitHub .

Jalankan sampel 'Bitbucket pull request dan webhook filter' untuk CodeBuild

AWS CodeBuild mendukung webhook ketika repositori sumber adalah Bitbucket. Ini berarti bahwa untuk proyek CodeBuild build yang memiliki kode sumbernya disimpan dalam repositori Bitbucket, webhook dapat digunakan untuk membangun kembali kode sumber setiap kali perubahan kode didorong ke repositori. Untuk informasi selengkapnya, lihat Acara webhook Bitbucket.

Contoh ini menunjukkan cara membuat permintaan tarik menggunakan repositori Bitbucket. Ini juga menunjukkan kepada Anda cara menggunakan webhook Bitbucket CodeBuild untuk memicu pembuatan proyek.

> **Note**
>
> Saat menggunakan webhook, pengguna dapat memicu build yang tidak terduga. Untuk mengurangi risiko ini, lihat. Praktik terbaik untuk menggunakan webhook

Topik

- Prasyarat
- Langkah 1: Buat proyek build dengan Bitbucket dan aktifkan webhooks
- Langkah 2: Memicu build dengan webhook Bitbucket

Prasyarat

Untuk menjalankan sampel ini, Anda harus menghubungkan AWS CodeBuild proyek Anda dengan akun Bitbucket Anda.

> **Note**
>
> CodeBuild telah memperbarui izinnya dengan Bitbucket. Jika sebelumnya Anda menghubungkan proyek Anda ke Bitbucket dan sekarang menerima kesalahan koneksi Bitbucket, Anda harus menyambung kembali untuk memberikan CodeBuild izin untuk mengelola webhook Anda.

Langkah 1: Buat proyek build dengan Bitbucket dan aktifkan webhooks

Langkah-langkah berikut menjelaskan cara membuat AWS CodeBuild proyek dengan Bitbucket sebagai repositori sumber dan mengaktifkan webhooks.

1. Buka AWS CodeBuild konsol di https://console.aws.amazon.com/codesuite/codebuild/home.
2. Jika halaman CodeBuild informasi ditampilkan, pilih Buat proyek build. Jika tidak, pada panel navigasi, perluas Build, pilih Build projects, lalu pilih Create build project.
3. Pilih Buat proyek build.
4. Dalam konfigurasi Proyek:

   Nama proyek

   Masukkan nama untuk proyek build ini. Membangun nama proyek harus unik di setiap AWS akun. Anda juga dapat menyertakan deskripsi opsional proyek build untuk membantu pengguna lain memahami tujuan proyek ini.
5. Dalam Sumber:

   Penyedia sumber

   Pilih Bitbucket. Ikuti petunjuk untuk menghubungkan (atau menyambung kembali) dengan Bitbucket dan kemudian pilih Otorisasi.

   Repositori

   Pilih Repositori di akun Bitbucket saya.

Jika sebelumnya Anda belum terhubung ke akun Bitbucket Anda, masukkan nama pengguna dan kata sandi aplikasi Bitbucket Anda, lalu pilih Simpan kredensi Bitbucket. Untuk token API Bitbucket, masukkan alamat email akun Atlassian Anda dan token API sebagai gantinya.

Repositori Bitbucket

Masukkan URL untuk repositori Bitbucket Anda.

6. Di Acara webhook sumber utama, pilih yang berikut ini.

> **Note**
>
> Bagian peristiwa webhook sumber utama hanya terlihat jika Anda memilih Repositori di akun Bitbucket saya di langkah sebelumnya.

1. Pilih Bangun ulang setiap kali perubahan kode didorong ke repositori ini saat Anda membuat proyek Anda.
2. Dari jenis Acara, pilih satu atau beberapa acara.
3. Untuk memfilter saat peristiwa memicu build, di bawah Memulai build dalam kondisi ini, tambahkan satu atau beberapa filter opsional.
4. Untuk memfilter saat peristiwa tidak dipicu, di bawah Jangan memulai build dalam kondisi ini, tambahkan satu atau beberapa filter opsional.
5. Pilih Tambahkan grup filter untuk menambahkan grup filter lain, jika diperlukan.

Untuk informasi selengkapnya tentang jenis dan filter acara webhook Bitbucket, lihat. Acara webhook Bitbucket

7. Di Lingkungan:

Citra lingkungan

Pilih salah satu cara berikut:

Untuk menggunakan image Docker yang dikelola oleh AWS CodeBuild:

Pilih Gambar terkelola, lalu buat pilihan dari Sistem operasi, Runtime, Gambar, dan versi Gambar. Buat pilihan dari jenis Lingkungan jika tersedia.

Untuk menggunakan gambar Docker lain:

Pilih gambar Kustom. Untuk jenis Lingkungan, pilih ARM, Linux, GPU Linux, atau Windows. Jika Anda memilih Registri lain, untuk URL registri eksternal, masukkan nama dan tag gambar Docker di Docker Hub, menggunakan format. *docker repository*/*docker image name* Jika Anda memilih Amazon ECR, gunakan repositori Amazon ECR dan gambar Amazon ECR untuk memilih gambar Docker di akun Anda. AWS

Untuk menggunakan image Docker pribadi:

Pilih gambar Kustom. Untuk jenis Lingkungan, pilih ARM, Linux, GPU Linux, atau Windows. Untuk registri Gambar, pilih Registri lain, lalu masukkan ARN kredensi untuk gambar Docker pribadi Anda. Kredensialnya harus dibuat oleh Secrets Manager. Untuk informasi lebih lanjut, lihat Apa itu AWS Secrets Manager? dalam AWS Secrets Manager User Guide.

Peran layanan

Pilih salah satu cara berikut:

- Jika Anda tidak memiliki peran CodeBuild layanan, pilih Peran layanan baru. Di Nama peran, masukkan nama untuk peran baru.
- Jika Anda memiliki peran CodeBuild layanan, pilih Peran layanan yang ada. Di Peran ARN, pilih peran layanan.

> **Note**
>
> Saat menggunakan konsol untuk membuat atau memperbarui proyek build, Anda dapat membuat peran CodeBuild layanan secara bersamaan. Secara default, peran hanya berfungsi dengan proyek build tersebut. Jika Anda menggunakan konsol untuk mengaitkan peran layanan ini dengan proyek build lain, peran akan diperbarui agar berfungsi dengan proyek build lainnya. Peran layanan dapat bekerja dengan hingga 10 proyek pembangunan.

8. Di Buildspec, lakukan salah satu hal berikut:

- Pilih Gunakan file buildspec untuk menggunakan file buildspec.ymldi direktori root kode sumber.
- Pilih Sisipkan perintah build untuk menggunakan konsol untuk menyisipkan perintah build.

Untuk informasi selengkapnya, lihat Referensi Buildspec.

9. Dalam Artefak:

   Jenis

   Pilih salah satu cara berikut:

   - Jika Anda tidak ingin membuat artefak keluaran build, pilih Tidak ada artefak.
   - Untuk menyimpan output build dalam bucket S3, pilih Amazon S3, lalu lakukan hal berikut:
     - Jika Anda ingin menggunakan nama proyek Anda untuk file atau folder ZIP keluaran build, biarkan Nama kosong. Jika tidak, masukkan nama. Secara default, nama artefak adalah nama proyek. Jika Anda ingin menggunakan nama yang berbeda, masukkan di kotak nama artefak. Jika Anda ingin mengeluarkan file ZIP, sertakan ekstensi zip.
     - Untuk nama Bucket, pilih nama bucket keluaran.
     - Jika Anda memilih Sisipkan perintah build sebelumnya dalam prosedur ini, untuk file Output, masukkan lokasi file dari build yang ingin Anda masukkan ke dalam file atau folder ZIP keluaran build. Untuk beberapa lokasi, pisahkan setiap lokasi dengan koma (misalnya,`appspec.yml, target/my-app.jar`). Untuk informasi lebih lanjut, lihat deskripsi `files` diSintaks Buildspec.

   Konfigurasi tambahan

   Perluas Konfigurasi tambahan dan atur opsi yang sesuai.

10. Pilih Buat proyek build. Pada halaman Ulasan, pilih Mulai build untuk menjalankan build.

Langkah 2: Memicu build dengan webhook Bitbucket

Untuk proyek yang menggunakan webhook Bitbucket, AWS CodeBuild buat build saat repositori Bitbucket mendeteksi perubahan dalam kode sumber Anda.

1. Buka AWS CodeBuild konsol di https://console.aws.amazon.com/codesuite/codebuild/home.
2. Pada panel navigasi, pilih Build projects, lalu pilih project yang terkait dengan repositori Bitbucket dengan webhook. Untuk informasi tentang membuat proyek webhook Bitbucket, lihat. the section called “Langkah 1: Buat proyek build dengan Bitbucket dan aktifkan webhooks”
3. Buat beberapa perubahan dalam kode di repositori Bitbucket proyek Anda.

4. Buat permintaan tarik pada repositori Bitbucket Anda. Untuk informasi selengkapnya, lihat Membuat permintaan tarik.
5. Pada halaman webhooks Bitbucket, pilih Lihat permintaan untuk melihat daftar peristiwa terbaru.
6. Pilih Lihat detail untuk melihat detail tentang respons yang dikembalikan oleh CodeBuild. Mungkin terlihat seperti ini:

```
"response":"Webhook received and build started: https://us-
east-1.console.aws.amazon.com/codebuild/home..."
"statusCode":200
```

7. Arahkan ke halaman permintaan tarik Bitbucket untuk melihat status build.

Jalan GitHub kan sampel Server Perusahaan untuk CodeBuild

AWS CodeBuild mendukung GitHub Enterprise Server sebagai repositori sumber. Contoh ini menunjukkan cara mengatur CodeBuild proyek Anda ketika repositori Server GitHub Enterprise Anda memiliki sertifikat yang diinstal. Ini juga menunjukkan cara mengaktifkan webhook sehingga CodeBuild membangun kembali kode sumber setiap kali perubahan kode didorong ke repositori Server GitHub Enterprise Anda.

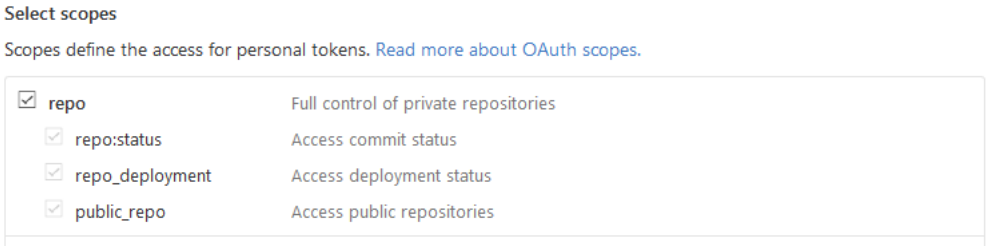
Topik

- Prasyarat
- Langkah 1: Buat proyek build dengan GitHub Enterprise Server dan aktifkan webhook

Prasyarat

1. Hasilkan token akses pribadi untuk CodeBuild proyek Anda. Kami menyarankan Anda membuat pengguna GitHub Enterprise dan membuat token akses pribadi untuk pengguna ini. Salin ke clipboard Anda sehingga dapat digunakan saat Anda membuat CodeBuild proyek Anda. Untuk informasi selengkapnya, lihat Membuat token akses pribadi untuk baris perintah di situs web GitHub Bantuan.

   Saat Anda membuat token akses pribadi, sertakan cak upan repo dalam definisi.



2. Unduh sertifikat Anda dari GitHub Enterprise Server. CodeBuild menggunakan sertifikat untuk membuat koneksi SSL tepercaya ke repositori.

   Linux/macOS klien:

   Dari jendela terminal, jalankan perintah berikut:

   ```
   echo -n | openssl s_client -connect HOST:PORTNUMBER \
       | sed -ne '/-BEGIN CERTIFICATE-/,/-END CERTIFICATE-/p' > /folder/filename.pem
   ```

   Ganti placeholder dalam perintah dengan nilai-nilai berikut:

   HOST. Alamat IP repositori Server GitHub Perusahaan Anda.

   PORTNUMBER. Nomor port yang Anda gunakan untuk menghubungkan (misalnya, 443).

   folder. Folder tempat Anda mengunduh sertifikat Anda.

   filename. Nama file dari file sertifikat Anda.

   > **Important**
   >
   > Simpan sertifikat sebagai file.pem.

   Klien Windows:

   Gunakan browser Anda untuk mengunduh sertifikat Anda dari GitHub Enterprise Server. Untuk melihat detail sertifikat situs, pilih ikon gembok. Untuk informasi tentang cara mengekspor sertifikat, lihat dokumentasi browser Anda.

> **Important**
>
> Simpan sertifikat sebagai file.pem.

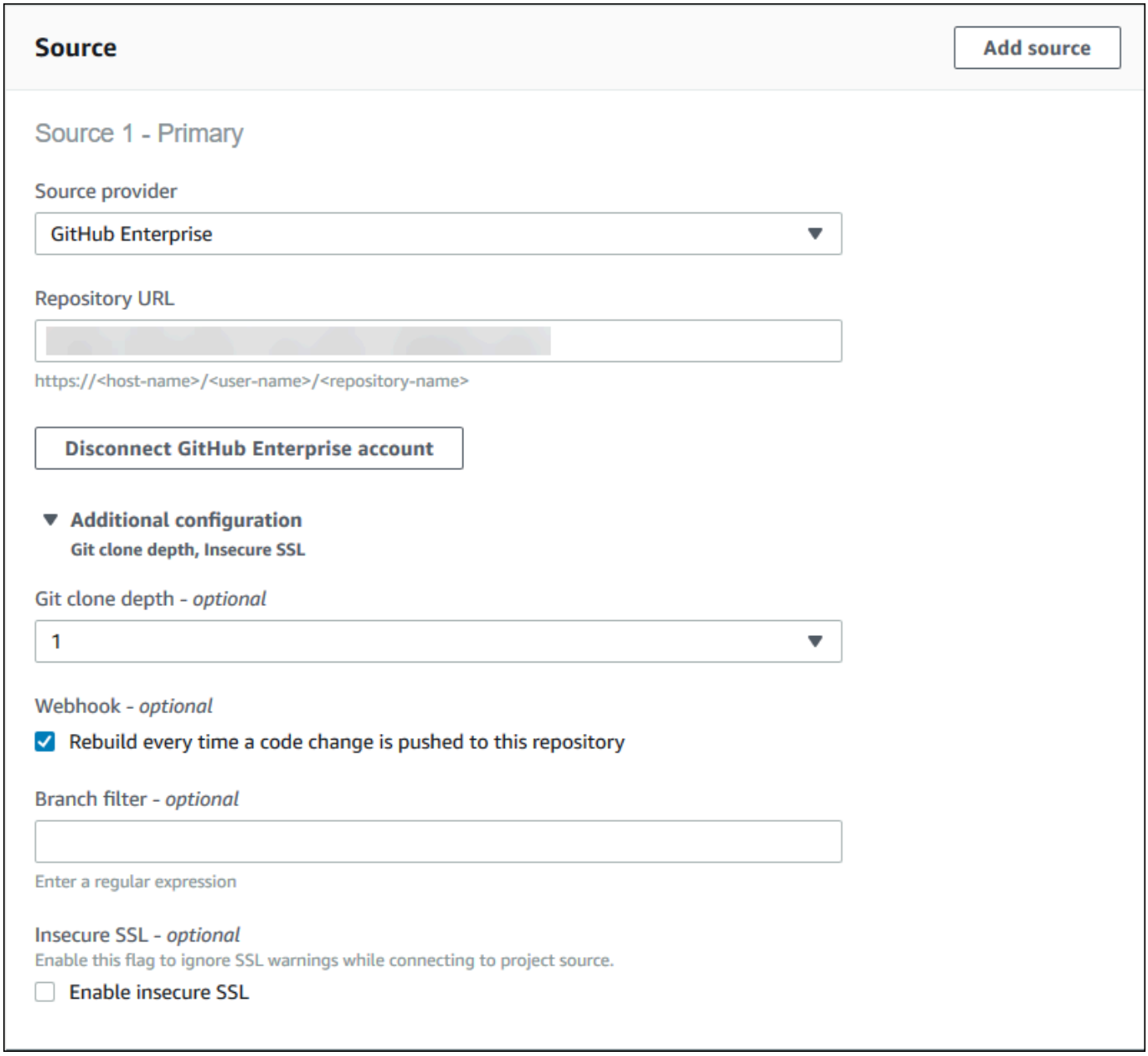
3. Unggah file sertifikat Anda ke bucket S3. Untuk informasi tentang cara membuat bucket S3, lihat Bagaimana cara membuat Bucket S3? Untuk informasi tentang cara mengunggah objek ke bucket S3, lihat Bagaimana cara mengunggah file dan folder ke bucket?

> **Note**
>
> Bucket ini harus berada di AWS wilayah yang sama dengan bangunan Anda. Misalnya, jika Anda menginstruksikan CodeBuild untuk menjalankan build di Wilayah AS Timur (Ohio), bucket harus berada di Wilayah AS Timur (Ohio).

Langkah 1: Buat proyek build dengan GitHub Enterprise Server dan aktifkan webhook

1. Buka AWS CodeBuild konsol di https://console.aws.amazon.com/codesuite/codebuild/home.
2. Jika halaman CodeBuild informasi ditampilkan, pilih Buat proyek build. Jika tidak, pada panel navigasi, per luas Bang un, pilih Bang un proyek, lalu pilih Buat proyek build.
3. Di Nama proyek, masukkan nama untuk proyek pembangunan ini. Nama proyek build harus unik di setiap AWS akun. Anda juga dapat menyertakan deskripsi opsional proyek build untuk membantu pengguna lain memahami untuk apa proyek ini digunakan.
4. Di Sumber, di Penyedia Sumber, pilih Server GitHub Perusahaan.
   - Pilih Kelola kreden tif akun, lalu pilih Token akses pribadi. Untuk Layanan, pilih Manajer Rahasia (disarankan), dan konfigurasikan rahasia Anda. Kemudian di token akses pribadi GitHub perusahaan, masukkan token akses pribadi Anda dan pilih S impan.
   - Di URL Repositori, masukkan jalur ke repositori Anda, termasuk nama repositori.
   - Perluas Additional configuration (Konfigurasi tambahan).
   - Pilih Bangun kembali setiap kali perubahan kode didorong ke repositori ini untuk membangun kembali setiap kali perubahan kode didorong ke repositori ini.
   - Pilih Aktifkan SSL tidak aman untuk mengabaikan peringatan SSL saat Anda terhubung ke repositori proyek Server GitHub Perusahaan Anda.

> **Note**
>
> Kami menyarankan Anda menggunakan Aktifkan SSL tidak aman hanya untuk pengujian. Ini tidak boleh digunakan di lingkungan produksi.



5. Di Lingkungan:

   Untuk gambar Lingkungan, lakukan salah satu hal berikut:

- Untuk menggunakan image Docker yang dikelola oleh AWS CodeBuild, pilih Gambar terkel ola, lalu buat pilihan dari Sistem operasi, Runtime (s), Gambar, dan versi Gambar. Buat pilihan dari jenis Lingkungan jika tersedia.
- Untuk menggunakan gambar Docker lain, pilih Custom image. Untuk tipe Lingkungan, pilih ARM, Linux, Linux GPU, atau Windows. Jika Anda memilih Registry lain, untuk URL registri eksternal, masukkan nama dan tag gambar Docker di Docker Hub, menggunakan format. *`docker repository`*/*`docker image name`* Jika Anda memilih Amazon ECR, gunakan repositori Amazon ECR dan gambar Amazon ECR untuk memilih gambar Docker di AWS akun Anda.
- Untuk menggunakan image Docker pribadi, pilih Gambar kustom. Untuk tipe Lingkungan, pilih ARM, Linux, Linux GPU, atau Windows. Untuk Registri gambar, pilih Registry lain, lalu masukkan ARN kredensional untuk image Docker pribadi Anda. KredenSIAL harus dibuat oleh Manajer Rahasia. Untuk informasi lebih lanjut, lihat Apa itu AWS Secrets Manager? di Panduan Pengguna AWS Secrets Manager .

6. Di Peran Layanan, lakukan salah satu hal berikut:

   - Jika Anda tidak memiliki peran CodeBuild layanan, pilih Per an layanan baru. Di Nama peran, masukkan nama untuk peran baru.
   - Jika Anda memiliki peran CodeBuild layanan, pilih Peran layanan yang ada. Di Role ARN, pilih peran layanan.

   > **Note**
   >
   > Saat Anda menggunakan konsol untuk membuat atau memperbarui proyek build, Anda dapat membuat peran CodeBuild layanan secara bersamaan. Secara default, peran hanya berfungsi dengan proyek build itu. Jika Anda menggunakan konsol untuk mengaitkan peran layanan ini dengan proyek build lain, peran tersebut diperbarui untuk bekerja dengan proyek build lainnya. Peran layanan dapat bekerja dengan hingga 10 proyek pembangunan.

7. Perluas Additional configuration (Konfigurasi tambahan).

   Jika Anda CodeBuild ingin bekerja dengan VPC Anda:

   - Untuk VPC, pilih ID VPC yang CodeBuild digunakan.

- Untuk Subnets VPC, pilih subnet yang menyertakan sumber daya yang menggunakan. CodeBuild
- Untuk grup Keamanan VPC, pilih grup keamanan yang digunakan untuk mengi CodeBuild zinkan akses ke sumber daya di VPC.

Untuk informasi selengkapnya, lihat Gunakan AWS CodeBuild dengan Amazon Virtual Private Cloud.

8. Di Buildspec, lakukan salah satu hal berikut:

   - Pilih Gunakan file buildspec untuk menggunakan file buildspec.yml di direktori root kode sumber.
   - Pilih S isipkan perintah build untuk menggunakan konsol untuk menyisipkan perintah build.

   Untuk informasi selengkapnya, lihat Referensi Buildspec.

9. Di Artifacts, untuk Type, lakukan salah satu hal berikut:

   - Jika Anda tidak ingin membuat artefak keluaran build, pilih Tidak ada artefak.
   - Untuk menyimpan output build dalam bucket S3, pilih Amazon S3, lalu lakukan hal berikut:
     - Jika Anda ingin menggunakan nama proyek Anda untuk file atau folder ZIP keluaran build, biarkan Nama kosong. Jika tidak, masukkan nama. Secara default, nama artefak adalah nama proyek. Jika Anda ingin menggunakan nama yang berbeda, masukkan di kotak nama artefak. Jika Anda ingin menampilkan file ZIP, sertakan ekstensi zip.
     - Untuk nama Bucket, pilih nama bucket keluaran.
     - Jika Anda memilih Sisip kan perintah build sebelumnya dalam prosedur ini, untuk file Output, masukkan lokasi file dari build yang ingin Anda masukkan ke dalam file atau folder ZIP keluaran build. Untuk beberapa lokasi, pisahkan setiap lokasi dengan koma (misalnya,`appspec.yml, target/my-app.jar`). Untuk informasi lebih lanjut, lihat deskripsi `files` diSintaks Buildspec.

10. Untuk jenis Cache, pilih salah satu dari berikut ini:

    - Jika Anda tidak ingin menggunakan cache, pilih Tidak ada cache.
    - Jika Anda ingin menggunakan cache Amazon S3, pilih Amazon S3, lalu lakukan hal berikut:
      - Untuk Bucket, pilih nama bucket S3 tempat cache disimpan.

- (Opsional) Untuk awalan jalur Cache, masukkan awalan jalur Amazon S3. Nilai aw alan Cache path mirip dengan nama direktori. Ini memungkinkan Anda untuk menyimpan cache di bawah direktori yang sama dalam ember.

  > **Important**
  >
  > Jangan menambahkan garis miring (/) pada akhir awalan jalur.

- Jika Anda ingin menggunakan cache lokal, pilih Lokal, lalu pilih satu atau lebih mode cache lokal.

  > **Note**
  >
  > Mode cache lapisan Docker hanya tersedia untuk Linux. Jika Anda memilihnya, proyek Anda harus berjalan dalam mode istimewa.

  Menggunakan cache menghemat waktu pembuatan yang cukup besar karena bagian lingkungan build yang dapat digunakan kembali disimpan dalam cache dan digunakan di seluruh build. Untuk informasi tentang menentukan cache dalam file buildspec, lihat. Sintaks Buildspec Untuk informasi selengkapnya tentang caching, lihatCache dibangun untuk meningkatkan kinerja.

11. Pilih Buat proyek pembangunan. Pada halaman proyek build, pilih Mulai membangun.

Jalankan permintaan GitHub tarik dan sampel filter webhook untuk CodeBuild

AWS CodeBuild mendukung webhook saat repositori sumbernya. GitHub Ini berarti bahwa untuk proyek CodeBuild build yang memiliki kode sumbernya disimpan dalam GitHub repositori, webhook dapat digunakan untuk membangun kembali kode sumber setiap kali perubahan kode didorong ke repositori. Untuk CodeBuild sampel, lihat AWS CodeBuild Sampel.

> **Note**
>
> Saat menggunakan webhook, pengguna dapat memicu build yang tidak terduga. Untuk mengurangi risiko ini, lihat. Praktik terbaik untuk menggunakan webhook

Topik

- Langkah 1: Buat proyek build dengan GitHub dan aktifkan webhooks
- Langkah 2: Verifikasi bahwa webhook diaktifkan

Langkah 1: Buat proyek build dengan GitHub dan aktifkan webhooks

1. Buka AWS CodeBuild konsol di https://console.aws.amazon.com/codesuite/codebuild/home.
2. Jika halaman CodeBuild informasi ditampilkan, pilih Buat proyek build. Jika tidak, pada panel navigasi, perluas Build, pilih Build projects, lalu pilih Create build project.
3. Pilih Buat proyek build.
4. Dalam konfigurasi Proyek:

   Nama proyek

   Masukkan nama untuk proyek build ini. Membangun nama proyek harus unik di setiap AWS akun. Anda juga dapat menyertakan deskripsi opsional proyek build untuk membantu pengguna lain memahami tujuan proyek ini.
5. Dalam Sumber:

   Penyedia sumber

   Pilih GitHub. Ikuti petunjuk untuk menghubungkan (atau menyambung kembali) dengan GitHub dan kemudian pilih Otorisasi.

   Repositori

   Pilih Repositori di akun saya GitHub.

   GitHub repositori

   Masukkan URL untuk GitHub repositori Anda.
6. Di Acara webhook sumber utama, pilih yang berikut ini.

> **Note**
>
> Bagian peristiwa webhook sumber utama hanya terlihat jika Anda memilih Repositori di GitHub akun saya pada langkah sebelumnya.

1. Pilih Rebuild setiap kali perubahan kode didorong ke repositori ini ketika Anda membuat proyek Anda.
2. Dari jenis Acara, pilih satu atau beberapa acara.
3. Untuk memfilter saat peristiwa memicu build, di bawah Memulai build dalam kondisi ini, tambahkan satu atau beberapa filter opsional.
4. Untuk memfilter saat peristiwa tidak dipicu, di bawah Jangan memulai build dalam kondisi ini, tambahkan satu atau beberapa filter opsional.
5. Pilih Tambahkan grup filter untuk menambahkan grup filter lain, jika diperlukan.

Untuk informasi selengkapnya tentang jenis dan filter acara GitHub webhook, lihatGitHub acara webhook.

7. Di Lingkungan:

   Citra lingkungan

   Pilih salah satu cara berikut:

   Untuk menggunakan image Docker yang dikelola oleh AWS CodeBuild:

   Pilih Gambar terkelola, lalu buat pilihan dari Sistem operasi, Runtime, Gambar, dan versi Gambar. Buat pilihan dari jenis Lingkungan jika tersedia.

   Untuk menggunakan gambar Docker lain:

   Pilih gambar Kustom. Untuk jenis Lingkungan, pilih ARM, Linux, GPU Linux, atau Windows. Jika Anda memilih Registri lain, untuk URL registri eksternal, masukkan nama dan tag gambar Docker di Docker Hub, menggunakan format. *`docker repository`*/*`docker image name`* Jika Anda memilih Amazon ECR, gunakan repositori Amazon ECR dan gambar Amazon ECR untuk memilih gambar Docker di akun Anda. AWS

   Untuk menggunakan image Docker pribadi:

   Pilih gambar Kustom. Untuk jenis Lingkungan, pilih ARM, Linux, GPU Linux, atau Windows. Untuk registri Gambar, pilih Registri lain, lalu masukkan ARN kredensi untuk gambar Docker pribadi Anda. Kredensialnya harus dibuat oleh Secrets Manager. Untuk informasi lebih lanjut, lihat Apa itu AWS Secrets Manager? dalam AWS Secrets Manager User Guide.

Peran layanan

Pilih salah satu cara berikut:

- Jika Anda tidak memiliki peran CodeBuild layanan, pilih Peran layanan baru. Di Nama peran, masukkan nama untuk peran baru.
- Jika Anda memiliki peran CodeBuild layanan, pilih Peran layanan yang ada. Di Peran ARN, pilih peran layanan.

> **Note**
>
> Saat menggunakan konsol untuk membuat atau memperbarui proyek build, Anda dapat membuat peran CodeBuild layanan secara bersamaan. Secara default, peran hanya berfungsi dengan proyek build tersebut. Jika Anda menggunakan konsol untuk mengaitkan peran layanan ini dengan proyek build lain, peran akan diperbarui agar berfungsi dengan proyek build lainnya. Peran layanan dapat bekerja dengan hingga 10 proyek pembangunan.

8. Di Buildspec, lakukan salah satu hal berikut:

- Pilih Gunakan file buildspec untuk menggunakan file buildspec.ymldi direktori root kode sumber.
- Pilih Sisipkan perintah build untuk menggunakan konsol untuk menyisipkan perintah build.

Untuk informasi selengkapnya, lihat Referensi Buildspec.

9. Dalam Artefak:

Jenis

Pilih salah satu cara berikut:

- Jika Anda tidak ingin membuat artefak keluaran build, pilih Tidak ada artefak.
- Untuk menyimpan output build dalam bucket S3, pilih Amazon S3, lalu lakukan hal berikut:
  - Jika Anda ingin menggunakan nama proyek Anda untuk file atau folder ZIP keluaran build, biarkan Nama kosong. Jika tidak, masukkan nama. Secara default, nama artefak adalah nama proyek. Jika Anda ingin menggunakan nama yang berbeda, masukkan di kotak nama artefak. Jika Anda ingin mengeluarkan file ZIP, sertakan ekstensi zip.
  - Untuk nama Bucket, pilih nama bucket keluaran.

- Jika Anda memilih Sisipkan perintah build sebelumnya dalam prosedur ini, untuk file Output, masukkan lokasi file dari build yang ingin Anda masukkan ke dalam file atau folder ZIP keluaran build. Untuk beberapa lokasi, pisahkan setiap lokasi dengan koma (misalnya,`appspec.yml, target/my-app.jar`). Untuk informasi lebih lanjut, lihat deskripsi `files` diSintaks Buildspec.

Konfigurasi tambahan

Perluas Konfigurasi tambahan dan atur opsi yang sesuai.

10. Pilih Buat proyek build. Pada halaman Review, pilih Start build untuk menjalankan build.

Langkah 2: Verifikasi bahwa webhook diaktifkan

1. Buka AWS CodeBuild konsol di https://console.aws.amazon.com/codesuite/codebuild/home.
2. Di panel navigasi, pilih Membangun proyek.
3. Lakukan salah satu tindakan berikut:
   - Pilih tautan untuk proyek build dengan webhook yang ingin Anda verifikasi, lalu pilih Detail build.
   - Pilih tombol di samping proyek build dengan webhook yang ingin Anda verifikasi, pilih Lihat detail, lalu pilih tab Build details.
4. Di acara webhook sumber utama, pilih tautan URL Webhook.
5. Di GitHub repositori Anda, pada halaman Pengaturan, di bawah Webhooks, verifikasi bahwa Permintaan Tarik dan Push dipilih.
6. Di pengaturan GitHub profil Anda, di bawah Pengaturan pribadi, Aplikasi, Aplikasi OAuth Resmi, Anda akan melihat bahwa aplikasi Anda telah diizinkan untuk mengakses AWS Wilayah yang Anda pilih.

Tutorial: Penandatanganan kode Apple dengan Fastlane dalam CodeBuild menggunakan S3 untuk penyimpanan sertifikat

fastlane adalah alat otomatisasi sumber terbuka yang populer untuk mengotomatiskan penerapan dan rilis beta untuk aplikasi iOS dan Android Anda. Ini menangani semua tugas yang membosankan, seperti membuat tangkapan layar, menangani penandatanganan kode, dan merilis aplikasi Anda.

Prasyarat

Untuk menyelesaikan tutorial ini, Anda harus terlebih dahulu menyiapkan yang berikut:

- Sebuah Akun AWS
- Akun Pengembang Apple
- Bucket S3 untuk menyimpan sertifikat
- fastlane diinstal di proyek Anda - Panduan untuk menginstal fastlane

Langkah 1: Siapkan Fastlane Match dengan S3 di mesin lokal Anda

Fastlane Match adalah salah satu alat Fastlane, dan memungkinkan konfigurasi yang mulus untuk penandatanganan kode di lingkungan pengembangan lokal Anda dan seterusnya. CodeBuild Fastlane Match menyimpan semua sertifikat penandatanganan kode dan profil penyediaan Anda di Git repository/S3 Bucket/Google Cloud Storage, dan mengunduh serta menginstal sertifikat dan profil yang diperlukan bila diperlukan.

Dalam contoh konfigurasi ini, Anda akan menyiapkan dan menggunakan bucket Amazon S3 untuk penyimpanan.

1. Inisialisasi kecocokan dalam proyek Anda:

```
fastlane match init
```

2. Saat diminta, pilih S3 sebagai mode penyimpanan.
3. Perbarui `Matchfile` Anda untuk menggunakan S3:

```
storage_mode("s3")
   s3_bucket("your-s3-bucket-name")
   s3_region("your-aws-region")
   type("appstore") # The default type, can be: appstore, adhoc, enterprise or
 development
```

Langkah 2: Siapkan Fastfile Anda

Buat atau perbarui `Fastfile` Anda dengan jalur berikut.

Pada CodeBuild, Fastlane Match harus dijalankan setiap kali Anda membuat dan menandatangani aplikasi Anda. Cara termudah untuk melakukannya adalah dengan menambahkan `match` tindakan ke jalur yang membangun aplikasi Anda.

```
default_platform(:ios)

platform :ios do
  before_all do
    setup_ci
  end

  desc "Build and sign the app"
  lane :build do
    match(type: "appstore", readonly: true)
    gym(
      scheme: "YourScheme",
      export_method: "app-store"
    )
  end
end
```

> **Note**
>
> Pastikan untuk menambahkan `setup_ci` ke `before_all` bagian `Fastfile` agar tindakan pertandingan berfungsi dengan benar. Ini memastikan bahwa gantungan kunci Fastlane sementara dengan izin yang sesuai digunakan. Tanpa menggunakan ini, Anda mungkin melihat kegagalan build atau hasil yang tidak konsisten.

Langkah 3: Jalankan **`perintah`** fastlane match untuk menghasilkan sertifikat dan profil masing-masing

Perintah fastlane match untuk tipe yang diberikan (yaitu, development, appstore, adhoc, enterprise) akan menghasilkan sertifikat dan profil jika tidak tersedia di remote store. Sertifikat dan profil akan disimpan di S3 oleh fastlane.

```
bundle exec fastlane match appstore
```

Eksekusi perintah akan interaktif dan fastlane akan meminta untuk mengatur frase sandi untuk mendekripsi sertifikat.

Langkah 4: Buat file aplikasi untuk proyek Anda

Buat atau tambahkan file aplikasi yang sesuai untuk proyek Anda.

1. Buat atau tambahkan Gymfile, Appfile, Snapfile , Deliverfile berdasarkan persyaratan pembuatan proyek Anda.
2. Lakukan perubahan ke repositori jarak jauh Anda

Langkah 5: Buat variabel lingkungan di Manajer Rahasia

Buat dua rahasia untuk menyimpan cookie sesi fastlane dan frase sandi yang cocok. Untuk informasi selengkapnya tentang membuat rahasia di Manajer Rahasia, lihat Membuat AWS Secrets Manager rahasia.

1. Akses cookie sesi fastlane Anda sebagai berikut.
   a. Kunci rahasia - `FASTLANE_SESSION`
   b. Nilai rahasia - cookie sesi yang dihasilkan dari menjalankan perintah berikut pada mesin lokal Anda.

> **Note**
>
> Nilai ini tersedia setelah otentikasi dalam file lokal:`~/.fastlane/spaceship/my_appleid_username/cookie`.

```
fastlane spaceauth -u <apple account>
```

2. Frasa sandi Fastlane Match - Untuk mengaktifkan Fastlane Match untuk mendekripsi sertifikat dan profil yang disimpan di bucket S3, Anda perlu menambahkan frasa sandi enkripsi yang Anda konfigurasikan pada langkah pengaturan Cocokkan ke variabel lingkungan proyek. CodeBuild
   a. Kunci rahasia - `MATCH_PASSWORD`

b. Nilai rahasia -*<match passphrase to decrypt certificates>*. Frasa sandi disetel saat membuat sertifikat di Langkah 3.

> **Note**
>
> Saat membuat rahasia di atas di Secrets Manager, ingatlah untuk memberikan nama rahasia dengan awalan berikut: `/CodeBuild/`

Langkah 6: Buat armada komputasi

Buat armada komputasi untuk proyek Anda.

1. Di konsol, buka CodeBuild dan buat armada komputasi baru.
2. Pilih “macOS” sebagai sistem operasi dan pilih jenis komputasi dan gambar yang sesuai.

Langkah 7: Buat proyek di CodeBuild

Buat proyek Anda di CodeBuild.

1. Buka AWS CodeBuild konsol di https://console.aws.amazon.com/codesuite/codebuild/home.
2. Buat proyek pembangunan. Untuk informasi selengkapnya, lihat Buat proyek build (konsol) dan Jalankan build (konsol).
3. Siapkan penyedia sumber Anda (seperti GitHub, CodeCommit). Ini adalah repositori sumber proyek iOS dan bukan repositori sertifikat.
4. Di Lingkungan:
   - Pilih Kapas itas Cadangan.
   - Untuk Ar mada, pilih armada yang dibuat di atas.
   - Berikan nama peran layanan yang CodeBuild akan dibuat untuk Anda.
   - Berikan variabel lingkungan di bawah ini.
     - Nama:`MATCH_PASSWORD`, Nilai:*<secrets arn>*, Jenis: Manajer Rahasia (Rahasia ARN dibuat pada langkah 5 untuk MATCH_PASSWORD)

- Nama:FASTLANE_SESSION, Nilai:*<secrets arn>*, Jenis: Manajer Rahasia (Rahasia ARN dibuat di Langkah 5 untuk FASTLANE_SESSION)

5. Di Buildspec, tambahkan yang berikut ini:

```
version: 0.2

phases:
  install:
    commands:
      - gem install bundler
      - bundle install
  build:
    commands:
      - echo "Building and signing the app..."
      - bundle exec fastlane build
  post_build:
    commands:
      - echo "Build completed on date"

artifacts:
  files:
    - '*/.ipa'
  name: app-$(date +%Y-%m-%d)
```

Langkah 8: Konfigurasikan peran IAM

Setelah proyek dibuat, pastikan peran layanan CodeBuild proyek Anda memiliki izin untuk mengakses bucket S3 yang berisi sertifikat. Tambahkan kebijakan berikut ke peran:

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetBucketLocation",
                "s3:ListBucket"
            ],
```

```
            "Resource": "arn:aws:s3:::your-s3-bucket-name"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:PutObject",
                "s3:DeleteObject"
            ],
            "Resource": "arn:aws:s3:::your-s3-bucket-name/*"
        }
    ]
}
```

Langkah 9: Jalankan build

Jalankan buildnya. Anda dapat meninjau status build dan log in CodeBuild.

Setelah pekerjaan selesai, Anda akan dapat melihat log pekerjaan.

Pemecahan masalah

- Jika Anda mengalami masalah dengan pengambilan sertifikat, pastikan izin IAM diatur dengan benar untuk akses S3.
- Jika Anda mengalami masalah dengan dekripsi sertifikat, pastikan Anda menetapkan frasa sandi yang benar di variabel lingkungan MATCH_PASSWORD.
- Untuk masalah penandatanganan kode, verifikasi bahwa akun Pengembang Apple Anda memiliki sertifikat dan profil yang diperlukan, dan bahwa pengenal bundel di proyek Xcode Anda cocok dengan yang ada di profil penyediaan Anda.

Pertimbangan keamanan

Berikut ini adalah pertimbangan keamanan untuk tutorial ini.

- Pastikan bucket S3 Anda memiliki pengaturan keamanan yang sesuai, termasuk enkripsi saat diam. Secara khusus, pastikan bucket tidak memiliki akses publik dan membatasi akses ke hanya CodeBuild dan sistem yang perlu memiliki akses.
- Pertimbangkan AWS Secrets Manager untuk menggunakan untuk menyimpan informasi sensitif seperti MATCH_PASSWORD dan FASTLANE_SESSION.

Contoh ini menyediakan pengaturan untuk penandatanganan kode iOS dengan Fastlane dalam CodeBuild menggunakan Amazon S3 untuk penyimpanan sertifikat. Anda mungkin perlu menyesuaikan beberapa langkah berdasarkan persyaratan dan CodeBuild lingkungan proyek spesifik Anda. Pendekatan ini memanfaatkan AWS layanan untuk meningkatkan keamanan dan integrasi dalam AWS ekosistem.

Tutorial: Penandatanganan kode Apple dengan Fastlane dalam CodeBuild penggunaan GitHub untuk penyimpanan sertifikat

fastlane adalah alat otomatisasi sumber terbuka yang populer untuk mengotomatiskan penerapan dan rilis beta untuk aplikasi iOS dan Android Anda. Ini menangani semua tugas yang membosankan, seperti membuat tangkapan layar, menangani penandatanganan kode, dan merilis aplikasi Anda.

Contoh ini menunjukkan cara mengatur penandatanganan kode Apple menggunakan Fastlane dalam CodeBuild proyek yang berjalan di armada Mac, dengan GitHub sebagai penyimpanan untuk sertifikat dan profil penyediaan.

Prasyarat

Untuk menyelesaikan tutorial ini, Anda harus terlebih dahulu menyiapkan yang berikut:

- Sebuah Akun AWS
- Akun Pengembang Apple
- GitHub Repositori pribadi untuk menyimpan sertifikat
- fastlane diinstal di proyek Anda - Panduan untuk menginstal fastlane

Langkah 1: Siapkan Fastlane Match dengan GitHub pada mesin lokal Anda

Fastlane Match adalah salah satu alat Fastlane, dan memungkinkan konfigurasi yang mulus untuk penandatanganan kode di lingkungan pengembangan lokal Anda dan seterusnya. CodeBuild Fastlane Match menyimpan semua sertifikat penandatanganan kode dan profil penyediaan Anda di Git repository/S3 Bucket/Google Cloud Storage, dan mengunduh serta menginstal sertifikat dan profil yang diperlukan bila diperlukan.

Dalam contoh konfigurasi ini, kita akan mengatur dan menggunakan repositori Git untuk penyimpanan.

1. Inisialisasi kecocokan dalam proyek Anda:

```
fastlane match init
```

2. Saat diminta, pilih GitHub sebagai mode penyimpanan.
3. Perbarui `Matchfile` Anda untuk digunakan GitHub:

```
git_url("https://github.com/your-username/your-certificate-repo.git")
storage_mode("git")
type("development") # The default type, can be: appstore, adhoc, enterprise or
 development
```

> **Note**
>
> Pastikan Anda memasukkan URL HTTPS untuk repositori Git Anda agar fastlane berhasil mengotentikasi dan mengkloning. Jika tidak, Anda mungkin melihat kesalahan otentikasi saat mencoba menggunakan kecocokan.

Langkah 2: Siapkan Fastfile Anda

Buat atau perbarui `Fastfile` Anda dengan jalur berikut.

Pada CodeBuild, Fastlane Match harus dijalankan setiap kali Anda membuat dan menandatangani aplikasi Anda. Cara termudah untuk melakukannya adalah dengan menambahkan `match` tindakan ke jalur yang membangun aplikasi Anda.

```
default_platform(:ios)

platform :ios do
  before_all do
    setup_ci
  end

  desc "Build and sign the app"
  lane :build do
    match(type: "appstore", readonly: true)
    gym(
      scheme: "YourScheme",
      export_method: "app-store"
    )
```

```
  end
end
```

> **Note**
>
> Pastikan untuk menambahkan `setup_ci` ke `before_all` bagian `Fastfile` agar tindakan pertandingan berfungsi dengan benar. Ini memastikan bahwa gantungan kunci Fastlane sementara dengan izin yang sesuai digunakan. Tanpa menggunakan ini, Anda mungkin melihat kegagalan build atau hasil yang tidak konsisten.

Langkah 3: Jalankan **`perintah`** fastlane match untuk menghasilkan sertifikat dan profil masing-masing

Perintah fastlane match untuk tipe yang diberikan (yaitu development, appstore, adhoc, enterprise) akan menghasilkan sertifikat dan profil jika tidak tersedia di remote store. Sertifikat dan profil akan disimpan GitHub oleh fastlane.

```
bundle exec fastlane match appstore
```

Eksekusi perintah akan interaktif dan fastlane akan meminta untuk mengatur frase sandi untuk mendekripsi sertifikat.

Langkah 4: Buat file aplikasi untuk proyek Anda

Buat atau tambahkan file aplikasi yang sesuai untuk proyek Anda.

1. Buat atau tambahkan Gymfile, Appfile, Snapfile, Deliverfile berdasarkan persyaratan pembuatan proyek Anda.
2. Lakukan perubahan ke repositori jarak jauh Anda.

Langkah 5: Buat variabel lingkungan di Manajer Rahasia

Buat tiga rahasia untuk menyimpan cookie sesi fastlane dan frase sandi yang cocok. Untuk informasi selengkapnya tentang membuat rahasia di Manajer Rahasia, lihat Membuat AWS Secrets Manager rahasia.

1. Akses cookie sesi fastlane Anda sebagai berikut.

   a. Kunci rahasia - `FASTLANE_SESSION`

   b. Nilai rahasia - cookie sesi yang dihasilkan dari menjalankan perintah berikut pada mesin lokal Anda.

   > **Note**
   >
   > Nilai ini tersedia setelah otentikasi dalam file lokal:`~/.fastlane/spaceship/my_appleid_username/cookie`.

   ```
   fastlane spaceauth -u <Apple_account>
   ```

2. Frasa sandi Fastlane Match - Untuk mengaktifkan Fastlane Match untuk mendekripsi sertifikat dan profil yang disimpan di repositori Git, Anda perlu menambahkan frasa sandi enkripsi yang Anda konfigurasikan pada langkah pengaturan Cocokkan ke variabel lingkungan proyek. CodeBuild

   a. Kunci rahasia - `MATCH_PASSWORD`

   b. Nilai rahasia -`<match passphrase to decrypt certificates>`. Frasa sandi disetel saat membuat sertifikat di Langkah 3.

3. Fastlane `MATCH_GIT_BASIC_AUTHORIZATION` - mengatur otorisasi dasar untuk pertandingan:

   a. Kunci rahasia:

   `MATCH_GIT_BASIC_AUTHORIZATION`

   b. Nilai rahasia - Nilai harus berupa string yang dikodekan base64 dari nama pengguna dan token akses pribadi (PAT) Anda dalam format`username:password`. Anda dapat menghasilkannya menggunakan perintah berikut:

   ```
   echo -n your_github_username:your_personal_access_token | base64
   ```

   Anda dapat membuat PAT di GitHub konsol di Profil Anda> Peng aturan> Pengaturan Pengembang> Token Akses Pribadi. Untuk informasi lebih lanjut, lihat panduan berikut: https://docs.github.com/en/authentication/keeping-your-account-and-data-secure/managing-your-personal-access-tokens.

> **Note**
>
> Saat membuat rahasia di atas di Secrets Manager, ingatlah untuk memberikan nama rahasia dengan awalan berikut: `/CodeBuild/`

Langkah 6: Buat armada komputasi

Buat armada komputasi untuk proyek Anda.

1. Di konsol, buka CodeBuild dan buat armada komputasi baru.
2. Pilih `macOS` sebagai sistem operasi dan pilih jenis dan gambar komputasi yang sesuai.

Langkah 7: Buat proyek di CodeBuild

Buat proyek Anda di CodeBuild.

1. Buka AWS CodeBuild konsol di https://console.aws.amazon.com/codesuite/codebuild/home.
2. Buat proyek pembangunan. Untuk informasi selengkapnya, lihat Buat proyek build (konsol) dan Jalankan build (konsol).
3. Siapkan penyedia sumber Anda (seperti GitHub, CodeCommit). Ini adalah repositori sumber proyek iOS dan bukan repositori sertifikat.
4. Di Lingkungan:
   - Pilih Kapas itas Cadangan.
   - Untuk Ar mada, pilih armada yang dibuat di atas.
   - Berikan nama peran layanan yang CodeBuild akan dibuat untuk Anda.
   - Berikan variabel lingkungan di bawah ini.
     - Nama:`MATCH_PASSWORD`, Nilai:*`<secrets arn>`*, Jenis: Manajer Rahasia (Rahasia ARN dibuat pada langkah 5 untuk MATCH_PASSWORD)
     - Nama:`FASTLANE_SESSION`, Nilai:*`<secrets arn>`*, Jenis: Manajer Rahasia (Rahasia ARN dibuat pada langkah 5 untuk FASTLANE_SESSION)
     - Nama:`MATCH_GIT_BASIC_AUTHORIZATION`, Nilai:*`<secrets ARN>`*, Jenis: Rahasia Manajer Rahasia ARN (dibuat pada langkah 5 untuk`MATCH_GIT_BASIC_AUTHORIZATION`)

5. Di Buildspec, tambahkan yang berikut ini:

```
version: 0.2

phases:
  install:
    commands:
      - gem install bundler
      - bundle install
  build:
    commands:
      - echo "Building and signing the app..."
      - bundle exec fastlane build
  post_build:
    commands:
      - echo "Build completed on date"

artifacts:
  files:
    - '*/.ipa'
  name: app-$(date +%Y-%m-%d)
```

Langkah 8: Jalankan build

Jalankan build. Anda dapat meninjau status build dan log in CodeBuild.

Setelah pekerjaan selesai, Anda akan dapat melihat log pekerjaan.

Pemecahan masalah

- Jika Anda mengalami masalah saat mengakses GitHub repositori, periksa kembali token akses pribadi Anda dan variabel lingkungan MATCH_GIT_BASIC_AUTHORIZATION.
- Jika Anda mengalami masalah dengan dekripsi sertifikat, pastikan Anda menetapkan frasa sandi yang benar di variabel lingkungan MATCH_PASSWORD.
- Untuk masalah penandatanganan kode, verifikasi bahwa akun Pengembang Apple Anda memiliki sertifikat dan profil yang diperlukan, dan bahwa pengenal bundel di proyek Xcode Anda cocok dengan yang ada di profil penyediaan Anda.

Pertimbangan keamanan

Berikut ini adalah pertimbangan keamanan untuk tutorial ini.

- Jaga GitHub repositori Anda untuk sertifikat pribadi dan secara teratur mengaudit akses.
- Pertimbangkan AWS Secrets Manager untuk menggunakan untuk menyimpan informasi sensitif seperti MATCH_PASSWORD dan FASTLANE_SESSION.

Contoh ini menyediakan pengaturan untuk penandatanganan kode iOS dengan Fastlane dalam CodeBuild penggunaan GitHub untuk penyimpanan sertifikat. Anda mungkin perlu menyesuaikan beberapa langkah berdasarkan persyaratan dan CodeBuild lingkungan proyek spesifik Anda. Pendekatan ini memanfaatkan AWS layanan untuk meningkatkan keamanan dan integrasi dalam AWS ekosistem.

Tetapkan nama artefak pada waktu pembuatan menggunakan versi semantik

Contoh ini berisi contoh file buildspec yang menunjukkan cara menentukan nama artefak yang dibuat pada waktu pembuatan. Nama yang ditentukan dalam file buildspec dapat menggabungkan perintah Shell dan variabel lingkungan untuk membuatnya unik. Nama yang Anda tentukan dalam file buildspec akan menggantikan nama yang Anda masukkan di konsol saat membuat project.

Jika Anda membangun beberapa kali, menggunakan nama artefak yang ditentukan dalam file buildspec dapat memastikan nama file artefak keluaran Anda unik. Misalnya, Anda dapat menggunakan tanggal dan stempel waktu yang dimasukkan ke dalam nama artefak pada waktu pembuatan.

Jika Anda ingin mengganti nama artefak yang Anda masukkan di konsol dengan nama di file buildspec, lakukan hal berikut:

1. Tetapkan proyek build Anda untuk mengganti nama artefak dengan nama di file buildspec.

   - Jika Anda menggunakan konsol untuk membuat proyek build, pilih Aktifkan versi semantik. Untuk informasi selengkapnya, lihat Buat proyek build (konsol).
   - Jika Anda menggunakan AWS CLI, atur `overrideArtifactName` ke true dalam JSON-formatted file yang diteruskan ke`create-project`. Untuk informasi selengkapnya, lihat Buat proyek build (AWS CLI).

- Jika Anda menggunakan AWS CodeBuild API, setel `overrideArtifactName` flag pada `ProjectArtifacts` objek saat proyek dibuat atau diperbarui atau build dimulai.

2. Tentukan nama dalam file buildspec. Gunakan contoh file buildspec berikut sebagai panduan.

Contoh Linux ini menunjukkan cara menentukan nama artefak yang menyertakan tanggal pembuatan build:

```
version: 0.2
phases:
  build:
    commands:
      - rspec HelloWorld_spec.rb
artifacts:
  files:
    - '**/*'
  name: myname-$(date +%Y-%m-%d)
```

Contoh Linux ini menunjukkan cara menentukan nama artefak yang menggunakan variabel CodeBuild lingkungan. Untuk informasi selengkapnya, lihat Variabel lingkungan di lingkungan build.

```
version: 0.2
phases:
  build:
    commands:
      - rspec HelloWorld_spec.rb
artifacts:
  files:
    - '**/*'
  name: myname-$AWS_REGION
```

Contoh Windows ini menunjukkan cara menentukan nama artefak yang menyertakan tanggal dan waktu pembuatan pembuatan:

```
version: 0.2
env:
  variables:
    TEST_ENV_VARIABLE: myArtifactName
phases:
  build:
    commands:
```

```
      - cd samples/helloworld
      - dotnet restore
      - dotnet run
artifacts:
  files:
    - '**/*'
  name: $Env:TEST_ENV_VARIABLE-$(Get-Date -UFormat "%Y%m%d-%H%M%S")
```

Contoh Windows ini menunjukkan cara menentukan nama artefak yang menggunakan variabel yang dideklarasikan dalam file buildspec dan variabel lingkungan. CodeBuild Untuk informasi selengkapnya, lihat Variabel lingkungan di lingkungan build.

```
version: 0.2
env:
  variables:
    TEST_ENV_VARIABLE: myArtifactName
phases:
  build:
    commands:
      - cd samples/helloworld
      - dotnet restore
      - dotnet run
artifacts:
  files:
    - '**/*'
  name: $Env:TEST_ENV_VARIABLE-$Env:AWS_REGION
```

Lihat informasi yang lebih lengkap di Buat referensi spesifikasi untuk CodeBuild.

Jalankan sampel Microsoft Windows untuk CodeBuild

Sampel ini menggunakan lingkungan AWS CodeBuild build yang menjalankan Microsoft Windows Server 2019, .NET Framework, dan .NET Core SDK untuk membuat file runtime dari kode yang ditulis dalam F # dan Visual Basic.

> **Important**
>
> Menjalankan sampel ini dapat mengakibatkan biaya ke AWS akun Anda. Ini termasuk kemungkinan biaya untuk CodeBuild dan untuk AWS sumber daya dan tindakan yang terkait dengan Amazon S3, AWS KMS, dan CloudWatch Log. Untuk informasi selengkapnya, lihat CodeBuildharga, harga Amazon S3, AWS Key Management Service harga, dan harga Amazon CloudWatch .

Jalankan sampel Windows

Gunakan prosedur berikut untuk menjalankan sampel Windows.

Untuk menjalankan sampel Windows

1. Buat file seperti yang dijelaskan di Berkas bagian Struktur direktori dan topik ini, lalu unggah ke bucket input S3 atau GitHub repositori CodeCommit atau.

   > **Important**
   >
   > Jangan mengunggah*(root directory name)*, hanya file di dalamnya*(root directory name)*.
   > Jika Anda menggunakan bucket input S3, pastikan untuk membuat file ZIP yang berisi file, lalu unggah ke bucket input. Jangan tambahkan *(root directory name)* ke file ZIP, hanya file di dalamnya*(root directory name)*.

2. Buat proyek build. Proyek build harus menggunakan `mcr.microsoft.com/dotnet/framework/sdk:4.8` image untuk membangun proyek .NET Framework.

   Jika Anda menggunakan AWS CLI untuk membuat proyek build, JSON-formatted input ke `create-project` perintah mungkin terlihat mirip dengan ini. (Ganti placeholder dengan nilai Anda sendiri.)

```
{
  "name": "sample-windows-build-project",
  "source": {
    "type": "S3",
    "location": "codebuild-region-ID-account-ID-input-bucket/windows-build-input-artifact.zip"
  },
  "artifacts": {
    "type": "S3",
    "location": "codebuild-region-ID-account-ID-output-bucket",
    "packaging": "ZIP",
    "name": "windows-build-output-artifact.zip"
  },
  "environment": {
    "type": "WINDOWS_SERVER_2019_CONTAINER",
    "image": "mcr.microsoft.com/dotnet/framework/sdk:4.8",
    "computeType": "BUILD_GENERAL1_MEDIUM"
  },
  "serviceRole": "arn:aws:iam::account-ID:role/role-name",
  "encryptionKey": "arn:aws:kms:region-ID:account-ID:key/key-ID"
}
```

3. Jalankan build, dan ikuti langkah-langkahnyaJalankan build secara manual.
4. Untuk mendapatkan artefak keluaran build, di bucket keluaran S3 Anda, unduh *windows-build-output-artifact*`.zip` file ke komputer atau instance lokal Anda. Ekstrak konten untuk sampai ke runtime dan file lainnya.

   - File runtime untuk sampel F # menggunakan .NET Framework,`FSharpHelloWorld.exe`, dapat ditemukan di `FSharpHelloWorld\bin\Debug` direktori.
   - File runtime untuk sampel Visual Basic menggunakan .NET Framework`VBHelloWorld.exe`,, dapat ditemukan di `VBHelloWorld\bin\Debug` direktori.

Struktur direktori

Sampel ini mengasumsikan struktur direktori berikut.

F # dan .NET Framework

```
(root directory name)
```

```
### buildspec.yml
### FSharpHelloWorld.sln
### FSharpHelloWorld
    ### App.config
    ### AssemblyInfo.fs
    ### FSharpHelloWorld.fsproj
    ### Program.fs
```

Visual Basic dan .NET Framework

```
(root directory name)
### buildspec.yml
### VBHelloWorld.sln
### VBHelloWorld
    ### App.config
    ### HelloWorld.vb
    ### VBHelloWorld.vbproj
    ### My Project
        ### Application.Designer.vb
        ### Application.myapp
        ### AssemblyInfo.vb
        ### Resources.Designer.vb
        ### Resources.resx
        ### Settings.Designer.vb
        ### Settings.settings
```

Berkas

Sampel ini menggunakan file-file berikut.

F # dan .NET Framework

`buildspec.yml`(dalam*`(root directory name)`*):

```
version: 0.2

env:
  variables:
    SOLUTION: .\FSharpHelloWorld.sln
    PACKAGE_DIRECTORY: .\packages
    DOTNET_FRAMEWORK: 4.8
```

```
phases:
  build:
    commands:
      - '& nuget restore $env:SOLUTION -PackagesDirectory $env:PACKAGE_DIRECTORY'
      - '& msbuild -p:FrameworkPathOverride="C:\Program Files (x86)\Reference
 Assemblies\Microsoft\Framework\.NETFramework\v$env:DOTNET_FRAMEWORK" $env:SOLUTION'
artifacts:
  files:
    - .\FSharpHelloWorld\bin\Debug\*
```

FSharpHelloWorld.sln(dalam*(root directory name)*):

```
Microsoft Visual Studio Solution File, Format Version 12.00
# Visual Studio 14
VisualStudioVersion = 14.0.25420.1
MinimumVisualStudioVersion = 10.0.40219.1
Project("{F2A71F9B-5D33-465A-A702-920D77279786}") = "FSharpHelloWorld",
 "FSharpHelloWorld\FSharpHelloWorld.fsproj", "{D60939B6-526D-43F4-9A89-577B2980DF62}"
EndProject
Global
  GlobalSection(SolutionConfigurationPlatforms) = preSolution
    Debug|Any CPU = Debug|Any CPU
    Release|Any CPU = Release|Any CPU
  EndGlobalSection
  GlobalSection(ProjectConfigurationPlatforms) = postSolution
    {D60939B6-526D-43F4-9A89-577B2980DF62}.Debug|Any CPU.ActiveCfg = Debug|Any CPU
    {D60939B6-526D-43F4-9A89-577B2980DF62}.Debug|Any CPU.Build.0 = Debug|Any CPU
    {D60939B6-526D-43F4-9A89-577B2980DF62}.Release|Any CPU.ActiveCfg = Release|Any CPU
    {D60939B6-526D-43F4-9A89-577B2980DF62}.Release|Any CPU.Build.0 = Release|Any CPU
  EndGlobalSection
  GlobalSection(SolutionProperties) = preSolution
    HideSolutionNode = FALSE
  EndGlobalSection
EndGlobal
```

App.config(dalam*(root directory name)*\FSharpHelloWorld):

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <startup>
    <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.8" />
  </startup>
```

```
</configuration>
```

AssemblyInfo.fs(dalam*(root directory name)*\FSharpHelloWorld):

```
namespace FSharpHelloWorld.AssemblyInfo

open System.Reflection
open System.Runtime.CompilerServices
open System.Runtime.InteropServices

// General Information about an assembly is controlled through the following
// set of attributes. Change these attribute values to modify the information
// associated with an assembly.
[<assembly: AssemblyTitle("FSharpHelloWorld")>]
[<assembly: AssemblyDescription("")>]
[<assembly: AssemblyConfiguration("")>]
[<assembly: AssemblyCompany("")>]
[<assembly: AssemblyProduct("FSharpHelloWorld")>]
[<assembly: AssemblyCopyright("Copyright ©  2017")>]
[<assembly: AssemblyTrademark("")>]
[<assembly: AssemblyCulture("")>]

// Setting ComVisible to false makes the types in this assembly not visible
// to COM components.  If you need to access a type in this assembly from
// COM, set the ComVisible attribute to true on that type.
[<assembly: ComVisible(false)>]

// The following GUID is for the ID of the typelib if this project is exposed to COM
[<assembly: Guid("d60939b6-526d-43f4-9a89-577b2980df62")>]

// Version information for an assembly consists of the following four values:
//
// Major Version
// Minor Version
// Build Number
// Revision
//
// You can specify all the values or you can default the Build and Revision Numbers
// by using the '*' as shown below:
// [<assembly: AssemblyVersion("1.0.*")>]
[<assembly: AssemblyVersion("1.0.0.0")>]
[<assembly: AssemblyFileVersion("1.0.0.0")>]
```

```
do
  ()
```

FSharpHelloWorld.fsproj(dalam*(root directory name)*\FSharpHelloWorld):

```
<?xml version="1.0" encoding="utf-8"?>
<Project ToolsVersion="14.0" DefaultTargets="Build" xmlns="http://
schemas.microsoft.com/developer/msbuild/2003">
  <Import Project="$(MSBuildExtensionsPath)\
$(MSBuildToolsVersion)\Microsoft.Common.props"
 Condition="Exists('$(MSBuildExtensionsPath)\
$(MSBuildToolsVersion)\Microsoft.Common.props')" />
  <PropertyGroup>
    <Configuration Condition=" '$(Configuration)' == '' ">Debug</Configuration>
    <Platform Condition=" '$(Platform)' == '' ">AnyCPU</Platform>
    <SchemaVersion>2.0</SchemaVersion>
    <ProjectGuid>d60939b6-526d-43f4-9a89-577b2980df62</ProjectGuid>
    <OutputType>Exe</OutputType>
    <RootNamespace>FSharpHelloWorld</RootNamespace>
    <AssemblyName>FSharpHelloWorld</AssemblyName>
    <TargetFrameworkVersion>v4.8</TargetFrameworkVersion>
    <AutoGenerateBindingRedirects>true</AutoGenerateBindingRedirects>
    <TargetFSharpCoreVersion>4.4.0.0</TargetFSharpCoreVersion>
    <Name>FSharpHelloWorld</Name>
  </PropertyGroup>
  <PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'Debug|AnyCPU' ">
    <DebugSymbols>true</DebugSymbols>
    <DebugType>full</DebugType>
    <Optimize>false</Optimize>
    <Tailcalls>false</Tailcalls>
    <OutputPath>bin\Debug\</OutputPath>
    <DefineConstants>DEBUG;TRACE</DefineConstants>
    <WarningLevel>3</WarningLevel>
    <PlatformTarget>AnyCPU</PlatformTarget>
    <DocumentationFile>bin\Debug\FSharpHelloWorld.XML</DocumentationFile>
    <Prefer32Bit>true</Prefer32Bit>
  </PropertyGroup>
  <PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'Release|AnyCPU' ">
    <DebugType>pdbonly</DebugType>
    <Optimize>true</Optimize>
    <Tailcalls>true</Tailcalls>
    <OutputPath>bin\Release\</OutputPath>
    <DefineConstants>TRACE</DefineConstants>
```

```
    <WarningLevel>3</WarningLevel>
    <PlatformTarget>AnyCPU</PlatformTarget>
    <DocumentationFile>bin\Release\FSharpHelloWorld.XML</DocumentationFile>
    <Prefer32Bit>true</Prefer32Bit>
  </PropertyGroup>
  <ItemGroup>
    <Reference Include="mscorlib" />
    <Reference Include="FSharp.Core, Version=$(TargetFSharpCoreVersion),
 Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a">
      <Private>True</Private>
    </Reference>
    <Reference Include="System" />
    <Reference Include="System.Core" />
    <Reference Include="System.Numerics" />
  </ItemGroup>
  <ItemGroup>
    <Compile Include="AssemblyInfo.fs" />
    <Compile Include="Program.fs" />
    <None Include="App.config" />
  </ItemGroup>
  <PropertyGroup>
    <MinimumVisualStudioVersion Condition="'$(MinimumVisualStudioVersion)' == ''">11</
MinimumVisualStudioVersion>
  </PropertyGroup>
  <Choose>
    <When Condition="'$(VisualStudioVersion)' == '11.0'">
      <PropertyGroup Condition="Exists('$(MSBuildExtensionsPath32)\..\Microsoft SDKs\F#
\3.0\Framework\v4.0\Microsoft.FSharp.Targets')">
        <FSharpTargetsPath>$(MSBuildExtensionsPath32)\..\Microsoft SDKs\F#
\3.0\Framework\v4.0\Microsoft.FSharp.Targets</FSharpTargetsPath>
      </PropertyGroup>
    </When>
    <Otherwise>
      <PropertyGroup Condition="Exists('$(MSBuildExtensionsPath32)\Microsoft
\VisualStudio\v$(VisualStudioVersion)\FSharp\Microsoft.FSharp.Targets')">
        <FSharpTargetsPath>$(MSBuildExtensionsPath32)\Microsoft\VisualStudio\v
$(VisualStudioVersion)\FSharp\Microsoft.FSharp.Targets</FSharpTargetsPath>
      </PropertyGroup>
    </Otherwise>
  </Choose>
  <Import Project="$(FSharpTargetsPath)" />
  <!-- To modify your build process, add your task inside one of the targets below and
 uncomment it.
       Other similar extension points exist, see Microsoft.Common.targets.
```

```
  <Target Name="BeforeBuild">
  </Target>
  <Target Name="AfterBuild">
  </Target>
  -->
</Project>
```

`Program.fs`(dalam*(root directory name)*\FSharpHelloWorld):

```
// Learn more about F# at http://fsharp.org
// See the 'F# Tutorial' project for more help.

[<EntryPoint>]
let main argv =
  printfn "Hello World"
  0 // return an integer exit code
```

Visual Basic dan .NET Framework

`buildspec.yml`(dalam*(root directory name)*):

```
version: 0.2

env:
  variables:
    SOLUTION: .\VBHelloWorld.sln
    PACKAGE_DIRECTORY: .\packages
    DOTNET_FRAMEWORK: 4.8

phases:
  build:
    commands:
      - '& "C:\ProgramData\chocolatey\bin\NuGet.exe" restore $env:SOLUTION -
PackagesDirectory $env:PACKAGE_DIRECTORY'
      - '& "C:\Program Files (x86)\MSBuild\14.0\Bin\MSBuild.exe" -
p:FrameworkPathOverride="C:\Program Files (x86)\Reference Assemblies\Microsoft
\Framework\.NETFramework\v$env:DOTNET_FRAMEWORK" $env:SOLUTION'
artifacts:
  files:
    - .\VBHelloWorld\bin\Debug\*
```

`VBHelloWorld.sln`(dalam*(root directory name)*):

```
Microsoft Visual Studio Solution File, Format Version 12.00
# Visual Studio 14
VisualStudioVersion = 14.0.25420.1
MinimumVisualStudioVersion = 10.0.40219.1
Project("{F184B08F-C81C-45F6-A57F-5ABD9991F28F}") = "VBHelloWorld", "VBHelloWorld
\VBHelloWorld.vbproj", "{4DCEC446-7156-4FE6-8CCC-219E34DD409D}"
EndProject
Global
  GlobalSection(SolutionConfigurationPlatforms) = preSolution
    Debug|Any CPU = Debug|Any CPU
    Release|Any CPU = Release|Any CPU
  EndGlobalSection
  GlobalSection(ProjectConfigurationPlatforms) = postSolution
    {4DCEC446-7156-4FE6-8CCC-219E34DD409D}.Debug|Any CPU.ActiveCfg = Debug|Any CPU
    {4DCEC446-7156-4FE6-8CCC-219E34DD409D}.Debug|Any CPU.Build.0 = Debug|Any CPU
    {4DCEC446-7156-4FE6-8CCC-219E34DD409D}.Release|Any CPU.ActiveCfg = Release|Any CPU
    {4DCEC446-7156-4FE6-8CCC-219E34DD409D}.Release|Any CPU.Build.0 = Release|Any CPU
  EndGlobalSection
  GlobalSection(SolutionProperties) = preSolution
    HideSolutionNode = FALSE
  EndGlobalSection
EndGlobal
```

App.config(dalam*(root directory name)*\VBHelloWorld):

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <startup>
    <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.8" />
  </startup>
</configuration>
```

HelloWorld.vb(dalam*(root directory name)*\VBHelloWorld):

```
Module HelloWorld

  Sub Main()
    MsgBox("Hello World")
  End Sub

End Module
```

VBHelloWorld.vbproj(dalam*(root directory name)*\VBHelloWorld):

```
<?xml version="1.0" encoding="utf-8"?>
<Project ToolsVersion="14.0" DefaultTargets="Build" xmlns="http://
schemas.microsoft.com/developer/msbuild/2003">
  <Import Project="$(MSBuildExtensionsPath)\
$(MSBuildToolsVersion)\Microsoft.Common.props"
 Condition="Exists('$(MSBuildExtensionsPath)\
$(MSBuildToolsVersion)\Microsoft.Common.props')" />
  <PropertyGroup>
    <Configuration Condition=" '$(Configuration)' == '' ">Debug</Configuration>
    <Platform Condition=" '$(Platform)' == '' ">AnyCPU</Platform>
    <ProjectGuid>{4DCEC446-7156-4FE6-8CCC-219E34DD409D}</ProjectGuid>
    <OutputType>Exe</OutputType>
    <StartupObject>VBHelloWorld.HelloWorld</StartupObject>
    <RootNamespace>VBHelloWorld</RootNamespace>
    <AssemblyName>VBHelloWorld</AssemblyName>
    <FileAlignment>512</FileAlignment>
    <MyType>Console</MyType>
    <TargetFrameworkVersion>v4.8</TargetFrameworkVersion>
    <AutoGenerateBindingRedirects>true</AutoGenerateBindingRedirects>
  </PropertyGroup>
  <PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'Debug|AnyCPU' ">
    <PlatformTarget>AnyCPU</PlatformTarget>
    <DebugSymbols>true</DebugSymbols>
    <DebugType>full</DebugType>
    <DefineDebug>true</DefineDebug>
    <DefineTrace>true</DefineTrace>
    <OutputPath>bin\Debug\</OutputPath>
    <DocumentationFile>VBHelloWorld.xml</DocumentationFile>
    <NoWarn>42016,41999,42017,42018,42019,42032,42036,42020,42021,42022</NoWarn>
  </PropertyGroup>
  <PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'Release|AnyCPU' ">
    <PlatformTarget>AnyCPU</PlatformTarget>
    <DebugType>pdbonly</DebugType>
    <DefineDebug>false</DefineDebug>
    <DefineTrace>true</DefineTrace>
    <Optimize>true</Optimize>
    <OutputPath>bin\Release\</OutputPath>
    <DocumentationFile>VBHelloWorld.xml</DocumentationFile>
    <NoWarn>42016,41999,42017,42018,42019,42032,42036,42020,42021,42022</NoWarn>
  </PropertyGroup>
  <PropertyGroup>
    <OptionExplicit>On</OptionExplicit>
```

```
  </PropertyGroup>
  <PropertyGroup>
    <OptionCompare>Binary</OptionCompare>
  </PropertyGroup>
  <PropertyGroup>
    <OptionStrict>Off</OptionStrict>
  </PropertyGroup>
  <PropertyGroup>
    <OptionInfer>On</OptionInfer>
  </PropertyGroup>
  <ItemGroup>
    <Reference Include="System" />
    <Reference Include="System.Data" />
    <Reference Include="System.Deployment" />
    <Reference Include="System.Xml" />
    <Reference Include="System.Core" />
    <Reference Include="System.Xml.Linq" />
    <Reference Include="System.Data.DataSetExtensions" />
    <Reference Include="System.Net.Http" />
  </ItemGroup>
  <ItemGroup>
    <Import Include="Microsoft.VisualBasic" />
    <Import Include="System" />
    <Import Include="System.Collections" />
    <Import Include="System.Collections.Generic" />
    <Import Include="System.Data" />
    <Import Include="System.Diagnostics" />
    <Import Include="System.Linq" />
    <Import Include="System.Xml.Linq" />
    <Import Include="System.Threading.Tasks" />
  </ItemGroup>
  <ItemGroup>
    <Compile Include="HelloWorld.vb" />
    <Compile Include="My Project\AssemblyInfo.vb" />
    <Compile Include="My Project\Application.Designer.vb">
      <AutoGen>True</AutoGen>
      <DependentUpon>Application.myapp</DependentUpon>
    </Compile>
    <Compile Include="My Project\Resources.Designer.vb">
      <AutoGen>True</AutoGen>
      <DesignTime>True</DesignTime>
      <DependentUpon>Resources.resx</DependentUpon>
    </Compile>
    <Compile Include="My Project\Settings.Designer.vb">
```

```
      <AutoGen>True</AutoGen>
      <DependentUpon>Settings.settings</DependentUpon>
      <DesignTimeSharedInput>True</DesignTimeSharedInput>
    </Compile>
  </ItemGroup>
  <ItemGroup>
    <EmbeddedResource Include="My Project\Resources.resx">
      <Generator>VbMyResourcesResXFileCodeGenerator</Generator>
      <LastGenOutput>Resources.Designer.vb</LastGenOutput>
      <CustomToolNamespace>My.Resources</CustomToolNamespace>
      <SubType>Designer</SubType>
    </EmbeddedResource>
  </ItemGroup>
  <ItemGroup>
    <None Include="My Project\Application.myapp">
      <Generator>MyApplicationCodeGenerator</Generator>
      <LastGenOutput>Application.Designer.vb</LastGenOutput>
    </None>
    <None Include="My Project\Settings.settings">
      <Generator>SettingsSingleFileGenerator</Generator>
      <CustomToolNamespace>My</CustomToolNamespace>
      <LastGenOutput>Settings.Designer.vb</LastGenOutput>
    </None>
    <None Include="App.config" />
  </ItemGroup>
  <Import Project="$(MSBuildToolsPath)\Microsoft.VisualBasic.targets" />
  <!-- To modify your build process, add your task inside one of the targets below and
 uncomment it.
       Other similar extension points exist, see Microsoft.Common.targets.
  <Target Name="BeforeBuild">
  </Target>
  <Target Name="AfterBuild">
  </Target>
  -->
</Project>
```

Application.Designer.vb(dalam*(root directory name)*\VBHelloWorld\My Project):

```
'------------------------------------------------------------------------------
' <auto-generated>
'   This code was generated by a tool.
'   Runtime Version:4.0.30319.42000
'
```

```
'   Changes to this file may cause incorrect behavior and will be lost if
'   the code is regenerated.
' </auto-generated>
'------------------------------------------------------------------------------

Option Strict On
Option Explicit On
```

Application.myapp(dalam*(root directory name)*\VBHelloWorld\My Project):

```
<?xml version="1.0" encoding="utf-8"?>
<MyApplicationData xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <MySubMain>false</MySubMain>
  <SingleInstance>false</SingleInstance>
  <ShutdownMode>0</ShutdownMode>
  <EnableVisualStyles>true</EnableVisualStyles>
  <AuthenticationMode>0</AuthenticationMode>
  <ApplicationType>2</ApplicationType>
  <SaveMySettingsOnExit>true</SaveMySettingsOnExit>
</MyApplicationData>
```

AssemblyInfo.vb(dalam*(root directory name)*\VBHelloWorld\My Project):

```
Imports System
Imports System.Reflection
Imports System.Runtime.InteropServices

' General Information about an assembly is controlled through the following
' set of attributes. Change these attribute values to modify the information
' associated with an assembly.

' Review the values of the assembly attributes

<Assembly: AssemblyTitle("VBHelloWorld")>
<Assembly: AssemblyDescription("")>
<Assembly: AssemblyCompany("")>
<Assembly: AssemblyProduct("VBHelloWorld")>
<Assembly: AssemblyCopyright("Copyright ©  2017")>
<Assembly: AssemblyTrademark("")>

<Assembly: ComVisible(False)>
```

```
'The following GUID is for the ID of the typelib if this project is exposed to COM
<Assembly: Guid("137c362b-36ef-4c3e-84ab-f95082487a5a")>

' Version information for an assembly consists of the following four values:
'
' Major Version
' Minor Version
' Build Number
' Revision
'
' You can specify all the values or you can default the Build and Revision Numbers
' by using the '*' as shown below:
' <Assembly: AssemblyVersion("1.0.*")>

<Assembly: AssemblyVersion("1.0.0.0")>
<Assembly: AssemblyFileVersion("1.0.0.0")>
```

Resources.Designer.vb(dalam*(root directory name)*\VBHelloWorld\My Project):

```
'------------------------------------------------------------------------------
' <auto-generated>
'   This code was generated by a tool.
'   Runtime Version:4.0.30319.42000
'
'   Changes to this file may cause incorrect behavior and will be lost if
'   the code is regenerated.
' </auto-generated>
'------------------------------------------------------------------------------

Option Strict On
Option Explicit On

Namespace My.Resources

  'This class was auto-generated by the StronglyTypedResourceBuilder
  'class via a tool like ResGen or Visual Studio.
  'To add or remove a member, edit your .ResX file then rerun ResGen
  'with the /str option, or rebuild your VS project.
  '''<summary>
  '''  A strongly-typed resource class, for looking up localized strings, etc.
  '''</summary>
```

```
<Global.System.CodeDom.Compiler.GeneratedCodeAttribute("System.Resources.Tools.StronglyTypedRe
"4.0.0.0"), _
 Global.System.Diagnostics.DebuggerNonUserCodeAttribute(), _
 Global.System.Runtime.CompilerServices.CompilerGeneratedAttribute(), _
 Global.Microsoft.VisualBasic.HideModuleNameAttribute()> _
 Friend Module Resources

   Private resourceMan As Global.System.Resources.ResourceManager

   Private resourceCulture As Global.System.Globalization.CultureInfo

   '''<summary>
   '''  Returns the cached ResourceManager instance used by this class.
   '''</summary>

<Global.System.ComponentModel.EditorBrowsableAttribute(Global.System.ComponentModel.EditorBrow
_
   Friend ReadOnly Property ResourceManager() As
Global.System.Resources.ResourceManager
     Get
       If Object.ReferenceEquals(resourceMan, Nothing) Then
         Dim temp As Global.System.Resources.ResourceManager = New
Global.System.Resources.ResourceManager("VBHelloWorld.Resources",
GetType(Resources).Assembly)
         resourceMan = temp
       End If
       Return resourceMan
     End Get
   End Property

   '''<summary>
   '''  Overrides the current thread's CurrentUICulture property for all
   '''  resource lookups using this strongly typed resource class.
   '''</summary>

<Global.System.ComponentModel.EditorBrowsableAttribute(Global.System.ComponentModel.EditorBrow
_
   Friend Property Culture() As Global.System.Globalization.CultureInfo
     Get
       Return resourceCulture
     End Get
     Set(ByVal value As Global.System.Globalization.CultureInfo)
       resourceCulture = value
```

```
      End Set
    End Property
  End Module
End Namespace
```

Resources.resx(dalam*(root directory name)*\VBHelloWorld\My Project):

```
<?xml version="1.0" encoding="utf-8"?>
<root>
  <!--
    Microsoft ResX Schema

    Version 2.0

    The primary goals of this format is to allow a simple XML format
    that is mostly human readable. The generation and parsing of the
    various data types are done through the TypeConverter classes
    associated with the data types.

    Example:

    ... ado.net/XML headers & schema ...
    <resheader name="resmimetype">text/microsoft-resx</resheader>
    <resheader name="version">2.0</resheader>
    <resheader name="reader">System.Resources.ResXResourceReader,
 System.Windows.Forms, ...</resheader>
    <resheader name="writer">System.Resources.ResXResourceWriter,
 System.Windows.Forms, ...</resheader>
    <data name="Name1"><value>this is my long string</value><comment>this is a
 comment</comment></data>
    <data name="Color1" type="System.Drawing.Color, System.Drawing">Blue</data>
    <data name="Bitmap1" mimetype="application/x-microsoft.net.object.binary.base64">
      <value>[base64 mime encoded serialized .NET Framework object]</value>
    </data>
    <data name="Icon1" type="System.Drawing.Icon, System.Drawing"
 mimetype="application/x-microsoft.net.object.bytearray.base64">
      <value>[base64 mime encoded string representing a byte array form of the .NET
 Framework object]</value>
      <comment>This is a comment</comment>
    </data>

    There are any number of "resheader" rows that contain simple
    name/value pairs.
```

```
   Each data row contains a name, and value. The row also contains a
   type or mimetype. Type corresponds to a .NET class that support
   text/value conversion through the TypeConverter architecture.
   Classes that don't support this are serialized and stored with the
   mimetype set.

   The mimetype is used for serialized objects, and tells the
   ResXResourceReader how to depersist the object. This is currently not
   extensible. For a given mimetype the value must be set accordingly:

   Note - application/x-microsoft.net.object.binary.base64 is the format
   that the ResXResourceWriter will generate, however the reader can
   read any of the formats listed below.

   mimetype: application/x-microsoft.net.object.binary.base64
   value   : The object must be serialized with
           : System.Serialization.Formatters.Binary.BinaryFormatter
           : and then encoded with base64 encoding.

   mimetype: application/x-microsoft.net.object.soap.base64
   value   : The object must be serialized with
           : System.Runtime.Serialization.Formatters.Soap.SoapFormatter
           : and then encoded with base64 encoding.

   mimetype: application/x-microsoft.net.object.bytearray.base64
   value   : The object must be serialized into a byte array
           : using a System.ComponentModel.TypeConverter
           : and then encoded with base64 encoding.
 -->
 <xsd:schema id="root" xmlns="" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
   <xsd:element name="root" msdata:IsDataSet="true">
     <xsd:complexType>
       <xsd:choice maxOccurs="unbounded">
         <xsd:element name="metadata">
           <xsd:complexType>
             <xsd:sequence>
               <xsd:element name="value" type="xsd:string" minOccurs="0" />
             </xsd:sequence>
             <xsd:attribute name="name" type="xsd:string" />
             <xsd:attribute name="type" type="xsd:string" />
             <xsd:attribute name="mimetype" type="xsd:string" />
           </xsd:complexType>
```

```
          </xsd:element>
          <xsd:element name="assembly">
            <xsd:complexType>
              <xsd:attribute name="alias" type="xsd:string" />
              <xsd:attribute name="name" type="xsd:string" />
            </xsd:complexType>
          </xsd:element>
          <xsd:element name="data">
            <xsd:complexType>
              <xsd:sequence>
                <xsd:element name="value" type="xsd:string" minOccurs="0"
msdata:Ordinal="1" />
                <xsd:element name="comment" type="xsd:string" minOccurs="0"
msdata:Ordinal="2" />
              </xsd:sequence>
              <xsd:attribute name="name" type="xsd:string" msdata:Ordinal="1" />
              <xsd:attribute name="type" type="xsd:string" msdata:Ordinal="3" />
              <xsd:attribute name="mimetype" type="xsd:string" msdata:Ordinal="4" />
            </xsd:complexType>
          </xsd:element>
          <xsd:element name="resheader">
            <xsd:complexType>
              <xsd:sequence>
                <xsd:element name="value" type="xsd:string" minOccurs="0"
msdata:Ordinal="1" />
              </xsd:sequence>
              <xsd:attribute name="name" type="xsd:string" use="required" />
            </xsd:complexType>
          </xsd:element>
        </xsd:choice>
      </xsd:complexType>
    </xsd:element>
  </xsd:schema>
  <resheader name="resmimetype">
    <value>text/microsoft-resx</value>
  </resheader>
  <resheader name="version">
    <value>2.0</value>
  </resheader>
  <resheader name="reader">
    <value>System.Resources.ResXResourceReader, System.Windows.Forms, Version=2.0.0.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089</value>
  </resheader>
  <resheader name="writer">
```

```
    <value>System.Resources.ResXResourceWriter, System.Windows.Forms, Version=2.0.0.0,
 Culture=neutral, PublicKeyToken=b77a5c561934e089</value>
  </resheader>
</root>
```

Settings.Designer.vb(dalam*(root directory name)*\VBHelloWorld\My Project):

```
'------------------------------------------------------------------------------
' <auto-generated>
'     This code was generated by a tool.
'     Runtime Version:4.0.30319.42000
'
'     Changes to this file may cause incorrect behavior and will be lost if
'     the code is regenerated.
' </auto-generated>
'------------------------------------------------------------------------------

Option Strict On
Option Explicit On

Namespace My

  <Global.System.Runtime.CompilerServices.CompilerGeneratedAttribute(), _

 Global.System.CodeDom.Compiler.GeneratedCodeAttribute("Microsoft.VisualStudio.Editors.Settings
 "11.0.0.0"), _

 Global.System.ComponentModel.EditorBrowsableAttribute(Global.System.ComponentModel.EditorBrows
 _
  Partial Friend NotInheritable Class MySettings
    Inherits Global.System.Configuration.ApplicationSettingsBase

    Private Shared defaultInstance As MySettings =
 CType(Global.System.Configuration.ApplicationSettingsBase.Synchronized(New
 MySettings), MySettings)

    #Region "My.Settings Auto-Save Functionality"
      #If _MyType = "WindowsForms" Then
        Private Shared addedHandler As Boolean

        Private Shared addedHandlerLockObject As New Object
```

```
        <Global.System.Diagnostics.DebuggerNonUserCodeAttribute(),
 Global.System.ComponentModel.EditorBrowsableAttribute(Global.System.ComponentModel.EditorBrows
 _
        Private Shared Sub AutoSaveSettings(ByVal sender As Global.System.Object, ByVal
 e As Global.System.EventArgs)
          If My.Application.SaveMySettingsOnExit Then
            My.Settings.Save()
          End If
        End Sub
      #End If
    #End Region

    Public Shared ReadOnly Property [Default]() As MySettings
      Get

        #If _MyType = "WindowsForms" Then
          If Not addedHandler Then
            SyncLock addedHandlerLockObject
              If Not addedHandler Then
                AddHandler My.Application.Shutdown, AddressOf AutoSaveSettings
                addedHandler = True
              End If
            End SyncLock
          End If
        #End If
        Return defaultInstance
      End Get
    End Property
  End Class
End Namespace

Namespace My

  <Global.Microsoft.VisualBasic.HideModuleNameAttribute(), _
  Global.System.Diagnostics.DebuggerNonUserCodeAttribute(), _
  Global.System.Runtime.CompilerServices.CompilerGeneratedAttribute()> _
  Friend Module MySettingsProperty

    <Global.System.ComponentModel.Design.HelpKeywordAttribute("My.Settings")> _
    Friend ReadOnly Property Settings() As Global.VBHelloWorld.My.MySettings
      Get
        Return Global.VBHelloWorld.My.MySettings.Default
      End Get
    End Property
```

```
  End Module
End Namespace
```

Settings.settings(dalam*(root directory name)*\VBHelloWorld\My Project):

```
<?xml version='1.0' encoding='utf-8'?>
<SettingsFile xmlns="http://schemas.microsoft.com/VisualStudio/2004/01/settings"
 CurrentProfile="(Default)" UseMySettingsClassName="true">
  <Profiles>
    <Profile Name="(Default)" />
  </Profiles>
  <Settings />
</SettingsFile>
```

Rencanakan build in AWS CodeBuild

Sebelum Anda menggunakan AWS CodeBuild, Anda harus menjawab pertanyaan-pertanyaan ini:

1. Di mana kode sumber disimpan? CodeBuild saat ini mendukung pembangunan dari penyedia repositori kode sumber berikut. Kode sumber harus berisi file spesifikasi build (buildspec). Buildspec adalah kumpulan perintah build dan pengaturan terkait, dalam format YAMAL, yang CodeBuild digunakan untuk menjalankan build. Anda dapat mendeklarasikan buildspec dalam definisi proyek build.

Penyedia repositori	Diperlukan	Dokumentasi
CodeCommit	Nama repositori. (Opsional) Komit ID yang terkait dengan kode sumber.	Lihat topik ini di Panduan AWS CodeCommit Pengguna: Buat CodeCommit repositori Buat komit di CodeCommit
Amazon S3	Masukkan nama bucket. Nama objek yang sesuai dengan file ZIP input build yang berisi kode sumber. (Opsional) ID Versi yang terkait dengan file ZIP input build.	Lihat topik ini di Panduan Memulai Amazon S3: Membuat bucket Menambahkan objek ke bucket
GitHub	Nama repositori.	Lihat topik ini di situs web GitHub Bantuan:

Penyedia repositori	Diperlukan	Dokumentasi
	(Opsional) Komit ID yang terkait dengan kode sumber.	Buat repo
Bitbucket	Nama repositori. (Opsional) Komit ID yang terkait dengan kode sumber.	Lihat topik ini di situs web dokumentasi Bitbucket Cloud: Membuat repositori

2. Perintah build mana yang perlu Anda jalankan dan dalam urutan apa? Secara default, CodeBuild mengunduh input build dari penyedia yang Anda tentukan dan mengunggah output build ke bucket yang Anda tentukan. Anda menggunakan buildspec untuk menginstruksikan cara mengubah input build yang diunduh menjadi output build yang diharapkan. Untuk informasi selengkapnya, lihat Referensi Buildspec.
3. Runtime dan alat apa yang Anda perlukan untuk menjalankan build? Misalnya, apakah Anda membangun untuk Java, Ruby, Python, atau? Node.js Apakah build membutuhkan Maven atau Ant atau kompiler untuk Java, Ruby, atau Python? Apakah build membutuhkan Git, the AWS CLI, atau alat lainnya?

   CodeBuild menjalankan build di lingkungan build yang menggunakan gambar Docker. Gambar Docker ini harus disimpan dalam jenis repositori yang didukung oleh. CodeBuild Ini termasuk repositori image CodeBuild Docker, Docker Hub, dan Amazon Elastic Container Registry (Amazon ECR) Registry (Amazon ECR). Untuk informasi selengkapnya tentang repositori gambar CodeBuild Docker, lihat. Gambar Docker disediakan oleh CodeBuild
4. Apakah Anda memerlukan AWS sumber daya yang tidak disediakan secara otomatis oleh CodeBuild? Jika demikian, kebijakan keamanan apa yang dibutuhkan sumber daya tersebut? Misalnya, Anda mungkin perlu memodifikasi peran CodeBuild layanan agar memungkinkan CodeBuild untuk bekerja dengan sumber daya tersebut.
5. Apakah Anda CodeBuild ingin bekerja dengan VPC Anda? Jika demikian, Anda memerlukan ID VPC, ID subnet, dan ID grup keamanan untuk konfigurasi VPC Anda. Untuk informasi selengkapnya, lihat Gunakan AWS CodeBuild dengan Amazon Virtual Private Cloud.

Setelah Anda menjawab pertanyaan-pertanyaan ini, Anda harus memiliki pengaturan dan sumber daya yang Anda butuhkan untuk menjalankan build dengan sukses. Untuk menjalankan build Anda, Anda dapat:

- Gunakan AWS CodeBuild konsol, AWS CLI, atau AWS SDK. Untuk informasi selengkapnya, lihat Jalankan build secara manual.
- Buat atau identifikasi pipeline AWS CodePipeline, lalu tambahkan tindakan build atau test yang menginstruksikan CodeBuild untuk menguji kode secara otomatis, menjalankan build, atau keduanya. Lihat informasi yang lebih lengkap di Gunakan CodeBuild dengan CodePipeline.

Buat referensi spesifikasi untuk CodeBuild

Topik ini memberikan informasi referensi penting tentang file spesifikasi build (buildspec). Buildspec adalah kumpulan perintah build dan pengaturan terkait, dalam format YAML, yang digunakan untuk menjalankan CodeBuild build. Anda dapat menyertakan buildspec sebagai bagian dari kode sumber atau Anda dapat menentukan buildspec saat Anda membuat proyek build. Untuk informasi tentang cara kerja spesifikasi build, lihatBagaimana cara CodeBuild kerja.

Topik

- Nama file buildspec dan lokasi penyimpanan
- Sintaks Buildspec
- Contoh Buildspec
- Versi Buildspec
- Referensi buildspec build batch

Nama file buildspec dan lokasi penyimpanan

Jika Anda menyertakan buildspec sebagai bagian dari kode sumber, secara default, file buildspec harus diberi nama `buildspec.yml` dan ditempatkan di root direktori sumber Anda.

Anda dapat mengganti nama dan lokasi file buildspec default. Misalnya, Anda dapat:

- Gunakan file buildspec yang berbeda untuk build yang berbeda di repositori yang sama, seperti dan. `buildspec_debug.yml` `buildspec_release.yml`
- Simpan file buildspec di tempat lain selain root direktori sumber Anda, seperti `config/buildspec.yml` atau dalam bucket S3. Bucket S3 harus berada di Wil AWS ayah yang sama dengan proyek build Anda. Tentukan file buildspec menggunakan ARN-nya (misalnya,). `arn:aws:s3:::<my-codebuild-sample2>/buildspec.yml`

Anda hanya dapat menentukan satu buildspec untuk proyek build, terlepas dari nama file buildspec.

Untuk mengganti nama file buildspec default, lokasi, atau keduanya, lakukan salah satu hal berikut:

- Jalan AWS CLI `create-project` kan `update-project` perintah or, atur `buildspec` nilai ke jalur ke file buildspec alternatif relatif terhadap nilai variabel lingkungan bawaan. `CODEBUILD_SRC_DIR` Anda juga dapat melakukan hal yang setara dengan `create project`

operasi di AWS SDK. Untuk informasi selengkapnya, lihat Membuat proyek build atau Ubah pengaturan proyek build.

- Jalan AWS CLI `start-build` kan perintah, atur `buildspecOverride` nilai ke jalur ke file buildspec alternatif relatif terhadap nilai variabel lingkungan bawaan. `CODEBUILD_SRC_DIR` Anda juga dapat melakukan hal yang setara dengan `start build` operasi di AWS SDK. Untuk informasi selengkapnya, lihat Jalankan build secara manual.
- Dalam AWS CloudFormation template, atur `BuildSpec` properti `Source` dalam sumber daya jenis `AWS::CodeBuild::Project` ke jalur ke file buildspec alternatif relatif terhadap nilai variabel lingkungan bawaan. `CODEBUILD_SRC_DIR` Untuk informasi selengkapnya, lihat BuildSpec properti di sumber AWS CodeBuild proyek di Panduan AWS CloudFormation Pengguna.

Sintaks Buildspec

File buildspec harus dinyatakan dalam format YAML.

Jika perintah berisi karakter, atau string karakter, yang tidak didukung oleh YAML, Anda harus melampirkan perintah dalam tanda kutip (“”). Perintah berikut terlampir dalam tanda kutip karena titik dua (:) diikuti oleh spasi tidak diperbolehkan di YAML. Tanda kutip dalam perintah lolos (\”).

```
"export PACKAGE_NAME=$(cat package.json | grep name | head -1 | awk -F: '{ print $2 }'
 | sed 's/[\",]//g')"
```

Buildspec memiliki sintaks berikut:

```
version: 0.2

run-as: Linux-user-name

env:
  shell: shell-tag
  variables:
    key: "value"
    key: "value"
  parameter-store:
    key: "value"
    key: "value"
  exported-variables:
    - variable
    - variable
```

```
  secrets-manager:
    key: secret-id:json-key:version-stage:version-id
  git-credential-helper: no | yes

proxy:
  upload-artifacts: no | yes
  logs: no | yes

batch:
  fast-fail: false | true
  # build-list:
  # build-matrix:
  # build-graph:
  # build-fanout:

phases:
  install:
    run-as: Linux-user-name
    on-failure: ABORT | CONTINUE | RETRY | RETRY-count | RETRY-regex |
 RETRY-count-regex
    runtime-versions:
      runtime: version
      runtime: version
    commands:
      - command
      - command
    finally:
      - command
      - command

  pre_build:
    run-as: Linux-user-name
    on-failure: ABORT | CONTINUE | RETRY | RETRY-count | RETRY-regex |
 RETRY-count-regex
    commands:
      - command
      - command
    finally:
      - command
      - command

  build:
    run-as: Linux-user-name
```

```
    on-failure: ABORT | CONTINUE | RETRY | RETRY-count | RETRY-regex |
 RETRY-count-regex
    commands:
      - command
      - command
    finally:
      - command
      - command

  post_build:
    run-as: Linux-user-name
    on-failure: ABORT | CONTINUE | RETRY | RETRY-count | RETRY-regex |
 RETRY-count-regex
    commands:
      - command
      - command
    finally:
      - command
      - command

reports:
  report-group-name-or-arn:
    files:
      - location
      - location
    base-directory: location
    discard-paths: no | yes
    file-format: report-format
artifacts:
  files:
    - location
    - location
  name: artifact-name
  discard-paths: no | yes
  base-directory: location
  exclude-paths: excluded paths
  enable-symlinks: no | yes
  s3-prefix: prefix
  secondary-artifacts:
    artifactIdentifier:
      files:
        - location
        - location
      name: secondary-artifact-name
```

```
      discard-paths: no | yes
      base-directory: location
    artifactIdentifier:
      files:
        - location
        - location
      discard-paths: no | yes
      base-directory: location
cache:
  key: key
  fallback-keys:
    - fallback-key
    - fallback-key
  action: restore | save
  paths:
    - path
    - path
```

Buildspec berisi yang berikut:

versi

Pemetaan yang diperlukan. Merupakan versi buildspec. Kami menyarankan Anda menggunakan`0.2`.

> **Note**
>
> Meskipun versi 0.1 masih didukung, kami sarankan Anda menggunakan versi 0.2 bila memungkinkan. Untuk informasi selengkapnya, lihat Versi Buildspec.

berjalan seperti

Urutan opsional. Tersedia untuk pengguna Linux saja. Menentukan pengguna Linux yang menjalankan perintah dalam file buildspec ini. `run-as`memberikan izin baca dan jalankan pengguna yang ditentukan. Ketika Anda menentukan `run-as` di bagian atas file buildspec, itu berlaku secara global untuk semua perintah. Jika Anda tidak ingin menentukan pengguna untuk semua perintah file buildspec, Anda dapat menentukan satu untuk perintah dalam fase dengan menggunakan `run-as` salah satu blok. `phases` Jika tidak `run-as` ditentukan, maka semua perintah berjalan sebagai pengguna root.

env

Urutan opsional. Merupakan informasi untuk satu atau lebih variabel lingkungan khusus.

> **Note**
>
> Untuk melindungi informasi sensitif, berikut ini disembunyikan di CodeBuild log:
>
> - AWS ID kunci akses. Untuk informasi selengkapnya, lihat Mengelola Kunci Akses untuk Pengguna IAM di Panduan AWS Identity and Access Management Pengguna.
> - String ditentukan menggunakan Parameter Store. Untuk informasi selengkapnya, lihat Panduan Konsol Penyimpanan Parameter Manajer Sistem dan Panduan Konsol Penyimpanan Parameter Manajer Sistem di Panduan Pengguna Amazon EC2 Systems Manager.
> - String ditentukan menggunakan AWS Secrets Manager. Untuk informasi selengkapnya, lihat Manajemen kunci.

> **Note**
>
> Masking cocok dengan nilai persis yang disimpan di Amazon EC2 Systems Manager Parameter Store atau AWS Secrets Manager. Jika perintah build mengubah rahasia sebelum muncul di log, hasilnya adalah string berbeda yang tidak ditutupi. Misalnya, mengkodekannya dengan Base64, membalikkannya, mengubah huruf besar/kecil, atau menggabungkannya dengan teks lain menciptakan nilai berbeda yang dapat muncul di log build.
> Untuk menjaga nilai tetap tertutup, simpan dalam bentuk yang digunakan build Anda. Misalnya, jika build Anda memerlukan Base64-encoded nilai, simpan Base64-encoded string langsung di AWS Secrets Manager atau Amazon EC2 Systems Manager Parameter Store. Dengan cara ini, nilai yang dicatat cocok dengan nilai yang disimpan.

lingkungan/cangkang

Urutan opsional. Menentukan shell yang didukung untuk sistem operasi Linux atau Windows.

Untuk sistem operasi Linux, tag shell yang didukung adalah:

- `bash`
- `/bin/sh`

Untuk sistem operasi Windows, tag shell yang didukung adalah:

- `powershell.exe`
- `cmd.exe`

env/variabel

Diperlukan jika `env` ditentukan, dan Anda ingin menentukan variabel lingkungan khusus dalam teks biasa. Berisi pemetaan *`value`* sk *`key`* alar/, di mana setiap pemetaan mewakili satu variabel lingkungan khusus dalam teks biasa. *`key`*adalah nama variabel lingkungan khusus, dan *`value`* merupakan nilai variabel itu.

> **Important**
>
> Kami sangat tidak menyarankan penyimpanan nilai sensitif dalam variabel lingkungan. Variabel lingkungan dapat ditampilkan dalam teks biasa menggunakan alat seperti CodeBuild konsol dan AWS CLI. Untuk nilai sensitif, sebaiknya gunakan `parameter-store` atau `secrets-manager` pemetaan sebagai gantinya, seperti yang dijelaskan nanti di bagian ini.
> Setiap variabel lingkungan yang Anda tetapkan menggantikan variabel lingkungan yang ada. Misalnya, jika image Docker sudah berisi variabel lingkungan bernama `MY_VAR` dengan nilai`my_value`, dan Anda menetapkan variabel lingkungan bernama `MY_VAR` dengan nilai`other_value`, `my_value` maka diganti dengan`other_value`. Demikian pula, jika image Docker sudah berisi variabel lingkungan bernama `PATH` dengan nilai`/usr/local/sbin:/usr/local/bin`, dan Anda menetapkan variabel lingkungan bernama `PATH` dengan nilai`$PATH:/usr/share/ant/bin`, `/usr/local/sbin:/usr/local/bin` maka diganti dengan nilai `$PATH:/usr/share/ant/bin` literal.
> Jangan mengatur variabel lingkungan apa pun dengan nama yang dimulai dengan`CODEBUILD_`. Awalan ini dicadangkan untuk penggunaan internal .
> Jika variabel lingkungan dengan nama yang sama didefinisikan di beberapa tempat, nilainya ditentukan sebagai berikut:
>
> - Nilai dalam panggilan operasi mulai build memiliki prioritas tertinggi. Anda dapat menambahkan atau mengganti variabel lingkungan saat Anda membuat build. Untuk informasi selengkapnya, lihat Jalankan AWS CodeBuild build secara manual.
> - Nilai dalam definisi proyek build akan diutamakan berikutnya. Anda dapat menambahkan variabel lingkungan di tingkat proyek saat Anda membuat atau mengedit proyek. Untuk informasi selengkapnya, lihat Membuat proyek pembangunan di AWS CodeBuild dan Ubah pengaturan proyek build di AWS CodeBuild.

- Nilai dalam deklarasi buildspec memiliki prioritas terendah.

env/penyimpanan parameter

Diperlukan jika `env` ditentukan, dan Anda ingin mengambil variabel lingkungan khusus yang disimpan di Amazon EC2 Systems Manager Parameter Store. Berisi pemetaan *`value`* sk *`key`* alar/, di mana setiap pemetaan mewakili satu variabel lingkungan khusus yang disimpan di Amazon EC2 Systems Manager Parameter Store. *`key`*adalah nama yang Anda gunakan nanti dalam perintah build untuk merujuk ke variabel lingkungan khusus ini, dan *`value`* merupakan nama variabel lingkungan khusus yang disimpan di Amazon EC2 Systems Manager Parameter Store. Untuk menyimpan nilai sensitif, lihat Peny impanan Parameter dan Panduan Manajer Sistem: Membuat dan menguji parameter String (konsol) di Panduan Pengguna Amazon EC2 Systems Manager.

> **Important**
>
> CodeBuild Untuk mengizinkan pengambilan variabel lingkungan khusus yang disimpan di Amazon EC2 Systems Manager Parameter Store, Anda harus menambahkan `ssm:GetParameters` tindakan ke peran CodeBuild layanan Anda. Untuk informasi selengkapnya, lihat Memungkinkan CodeBuild untuk berinteraksi dengan orang lain AWS layanan.
>
> Setiap variabel lingkungan yang Anda ambil dari Amazon EC2 Systems Manager Parameter Store menggantikan variabel lingkungan yang ada. Misalnya, jika image Docker sudah berisi variabel lingkungan bernama `MY_VAR` dengan nilai`my_value`, dan Anda mengambil variabel lingkungan bernama `MY_VAR` dengan nilai`other_value`, `my_value` maka diganti dengan`other_value`. Demikian pula, jika image Docker sudah berisi variabel lingkungan bernama `PATH` dengan nilai`/usr/local/sbin:/usr/local/bin`, dan Anda mengambil variabel lingkungan bernama `PATH` dengan nilai`$PATH:/usr/share/ant/bin`, `/usr/local/sbin:/usr/local/bin` maka diganti dengan nilai `$PATH:/usr/share/ant/bin` literal.
>
> Jangan menyimpan variabel lingkungan apa pun dengan nama yang dimulai dengan`CODEBUILD_`. Awalan ini dicadangkan untuk penggunaan internal .
>
> Jika variabel lingkungan dengan nama yang sama didefinisikan di beberapa tempat, nilainya ditentukan sebagai berikut:

- Nilai dalam panggilan operasi mulai build memiliki prioritas tertinggi. Anda dapat menambahkan atau mengganti variabel lingkungan saat Anda membuat build. Untuk informasi selengkapnya, lihat Jalankan AWS CodeBuild build secara manual.
- Nilai dalam definisi proyek build akan diutamakan berikutnya. Anda dapat menambahkan variabel lingkungan di tingkat proyek saat Anda membuat atau mengedit proyek. Untuk informasi selengkapnya, lihat Membuat proyek pembangunan di AWS CodeBuild dan Ubah pengaturan proyek build di AWS CodeBuild.
- Nilai dalam deklarasi buildspec memiliki prioritas terendah.

pengelola lingkungan/rahasia

Diperlukan jika Anda ingin mengambil variabel lingkungan khusus yang disimpan di AWS Secrets Manager. Tentukan Manajer Rahasia `reference-key` menggunakan pola berikut:

<key>: *<secret-id>*:*<json-key>*:*<version-stage>*:*<version-id>*

<key>

(Diperlukan) Nama variabel lingkungan lokal. Gunakan nama ini untuk mengakses variabel selama build.

<secret-id>

(Diperlukan) Nama atau Nama Sumber Daya Amazon (ARN) yang berfungsi sebagai pengidentifikasi unik untuk rahasia. Untuk mengakses rahasia di AWS akun Anda, cukup tentukan nama rahasia. Untuk mengakses rahasia di AWS akun yang berbeda, tentukan ARN rahasia.

<json-key>

(Opsional) Menentukan nama kunci pasangan kunci-nilai Manajer Rahasia yang nilainya ingin Anda ambil. Jika Anda tidak menentukan `ajson-key`, CodeBuild mengambil seluruh teks rahasia.

<version-stage>

(Opsional) Menentukan versi rahasia yang ingin Anda ambil dengan label pementasan yang dilampirkan ke versi. Label pementasan digunakan untuk melacak versi yang berbeda selama proses rotasi. Jika Anda menggunakan`version-stage`, jangan tentukan`version-id`. Jika Anda tidak menentukan tahap versi atau ID versi, defaultnya adalah mengambil versi dengan nilai tahap versi`AWSCURRENT`.

<version-id>

(Opsional) Menentukan pengenal unik dari versi rahasia yang ingin Anda gunakan. Jika Anda menentukan `version-id`, jangan tentukan `version-stage`. Jika Anda tidak menentukan tahap versi atau ID versi, defaultnya adalah mengambil versi dengan nilai tahap versi`AWSCURRENT`.

Dalam contoh berikut, `TestSecret` adalah nama pasangan kunci-nilai yang disimpan di Secrets Manager. Kuncinya `TestSecret` adalah`MY_SECRET_VAR`. Anda mengakses variabel selama build menggunakan `LOCAL_SECRET_VAR` nama.

```
env:
  secrets-manager:
    LOCAL_SECRET_VAR: "TestSecret:MY_SECRET_VAR"
```

Untuk informasi selengkapnya, lihat Apa itu AWS Secrets Manager dalam Panduan Pengguna AWS Secrets Manager .

env/variabel yang diekspor

Pemetaan opsional. Digunakan untuk membuat daftar variabel lingkungan yang ingin Anda ekspor. Tentukan nama setiap variabel yang ingin Anda ekspor pada baris terpisah di bawahnya`exported-variables`. Variabel yang ingin Anda ekspor harus tersedia di wadah Anda selama pembuatan. Variabel yang Anda ekspor dapat menjadi variabel lingkungan.

Variabel lingkungan yang diekspor digunakan bersama dengan AWS CodePipeline untuk mengekspor variabel lingkungan dari tahap pembuatan saat ini ke tahap berikutnya dalam pipeline. Untuk informasi selengkapnya, lihat Bek erja dengan variabel di Panduan AWS CodePipeline Pengguna.

Selama build, nilai variabel tersedia dimulai dengan `install` fase. Ini dapat diperbarui antara awal `install` fase dan akhir `post_build` fase. Setelah `post_build` fase berakhir, nilai variabel yang diekspor tidak dapat berubah.

> **Note**
>
> Berikut ini tidak dapat diekspor:
>
> - Rahasia Penyimpanan Parameter Pengelola Sistem Amazon EC2 yang ditentukan dalam proyek build.

- Rahasia Manajer Rahasia ditentukan dalam proyek pembangunan
- Variabel lingkungan yang dimulai dengan`AWS_`.

env/git-kredential-pembantu

Pemetaan opsional. Digunakan untuk menunjukkan apakah CodeBuild menggunakan pembantu kredensi Git untuk memberikan kredenSIAL Git. `yes`jika digunakan. Jika tidak, `no` atau tidak ditentukan. Untuk informasi selengkapnya, lihat git di situs web Git.

> **Note**
>
> `git-credential-helper`tidak didukung untuk build yang dipicu oleh webhook untuk repositori Git publik.

proxy

Urutan opsional. Digunakan untuk mewakili pengaturan jika Anda menjalankan build Anda di server proxy eksplisit. Untuk informasi selengkapnya, lihat Jalan CodeBuild kan di server proxy eksplisit.

proksi/unggah-artefak

Pemetaan opsional. Setel ke `yes` jika Anda ingin build Anda di server proxy eksplisit untuk mengunggah artefak. Nilai default-nya `no`.

proksi/log

Pemetaan opsional. Setel ke `yes` untuk build Anda di server proxy eksplisit untuk membuat CloudWatch log. Nilai default-nya `no`.

fase

Urutan yang diperlukan. Merupakan perintah CodeBuild yang berjalan selama setiap fase build.

> **Note**
>
> Di buildspec versi 0.1, CodeBuild jalankan setiap perintah dalam instance terpisah dari shell default di lingkungan build. Ini berarti bahwa setiap perintah berjalan secara

terpisah dari semua perintah lainnya. Oleh karena itu, secara default, Anda tidak dapat menjalankan perintah tunggal yang bergantung pada keadaan perintah sebelumnya (misalnya, mengubah direktori atau mengatur variabel lingkungan). Untuk mengatasi batasan ini, kami sarankan Anda menggunakan versi 0.2, yang memecahkan masalah ini. Jika Anda harus menggunakan buildspec versi 0.1, kami merekomendasikan pendekatan di. Shell dan perintah di lingkungan build

fase/*/run-as

Urutan opsional. Gunakan dalam fase build untuk menentukan pengguna Linux yang menjalankan perintahnya. Jika juga `run-as` ditentukan secara global untuk semua perintah di bagian atas file buildspec, maka pengguna tingkat fase diutamakan. Misalnya, jika secara global `run-as` menentukan User-1, dan untuk `install` fase hanya `run-as` pernyataan yang ditentukan User-2, maka semua perintah dalam file buildspec dijalankan sebagai perintah User-1 kecuali dalam `install` fase, yang dijalankan sebagai. User-2

fase/*/pada kegagalan

Urutan opsional. Menentukan tindakan yang harus diambil jika kegagalan terjadi selama fase. Ini dapat berupa salah satu dari nilai berikut:

- `ABORT`- Batalkan pembangunan.
- `CONTINUE`- Lanjutkan ke fase berikutnya.
- `RETRY`- Coba ulang build hingga 3 kali dengan pesan kesalahan yang cocok dengan ekspresi `.*` reguler.
- `RETRY-`*`count`*- Coba ulang build untuk jumlah yang ditentukan, seperti yang diwakili oleh *`count`* dengan pesan kesalahan yang cocok dengan ekspresi `.*` reguler. Perhatikan bahwa *`count`* harus antara 0 dan 100. Misalnya, nilai yang valid termasuk `RETRY-4` dan`RETRY-8`.
- `RETRY-`*`regex`*- Coba ulang build hingga 3 kali, dan gunakan *`regex`* untuk menyertakan ekspresi reguler agar sesuai dengan pesan kesalahan yang ditentukan. Misalnya, nilai yang valid termasuk `Retry-.*Error: Unable to connect to database.*` dan`RETRY-invalid+`.
- `RETRY-`*`count`*`-`*`regex`*- Coba lagi build untuk sejumlah kali tertentu, seperti yang diwakili oleh*`count`*. Perhatikan bahwa *`count`* harus antara 0 dan 100. Anda juga dapat menggunakan *`regex`* untuk menyertakan ekspresi reguler untuk mencocokkan pesan kesalahan. Misalnya, nilai yang valid termasuk `Retry-3-.*connection timed out.*` dan`RETRY-8-invalid+`.

Jika properti ini tidak ditentukan, proses kegagalan mengikuti fase transisi seperti yang ditunjukkan padaBangun transisi fase.

> **Important**
>
> A `on-failure` tribut tidak didukung saat menggunakan komputasi Lambda atau kapasitas cadangan. Atribut ini hanya berfungsi dengan gambar komputasi EC2 yang disediakan oleh. CodeBuild

fase/*/ akhirnya

Blok opsional. Perintah yang ditentukan dalam `finally` blok dijalankan setelah perintah di `commands` blok. Perintah dalam `finally` blok dijalankan bahkan jika perintah di `commands` blok gagal. Misalnya, jika `commands` blok berisi tiga perintah dan yang pertama gagal, mele CodeBuild watkan dua perintah yang tersisa dan menjalankan perintah apa pun di `finally` blok. Fase ini berhasil ketika semua perintah di `commands` dan `finally` blok berjalan dengan sukses. Jika ada perintah dalam fase gagal, fase gagal.

Nama fase build yang diizinkan adalah:

fase/instal

Urutan opsional. Merupakan perintah, jika ada, yang CodeBuild berjalan selama instalasi. Kami menyarankan Anda menggunakan `install` fase hanya untuk menginstal paket di lingkungan build. Misalnya, Anda dapat menggunakan fase ini untuk menginstal kerangka pengujian kode seperti Mocha atau RSpec.

phases/install/versi runtime-

Urutan opsional. Versi runtime didukung dengan gambar standar Ubuntu 5.0 atau yang lebih baru dan gambar standar Amazon Linux 2 4.0 atau lebih baru. Jika ditentukan, setidaknya satu runtime harus disertakan dalam bagian ini. Tentukan runtime menggunakan versi tertentu, versi utama diikuti oleh `.x` untuk menentukan yang CodeBuild menggunakan versi utama tersebut dengan versi minor terbarunya, atau `latest` untuk menggunakan versi mayor dan minor terbaru (misalnya,`ruby: 3.2`,`nodejs: 18.x`, atau`java: latest`). Anda dapat menentukan runtime menggunakan angka atau variabel lingkungan. Misalnya, jika Anda menggunakan gambar standar Amazon Linux 2 4.0, maka berikut ini menentukan bahwa versi 17 dari Java, versi minor terbaru dari python versi 3, dan versi yang terkandung dalam variabel

lingkungan Ruby diinstal. Untuk informasi selengkapnya, lihat Gambar Docker disediakan oleh CodeBuild.

```
phases:
  install:
    runtime-versions:
      java: corretto8
      python: 3.x
      ruby: "$MY_RUBY_VAR"
```

Anda dapat menentukan satu atau lebih runtime di `runtime-versions` bagian file buildspec Anda. Jika runtime Anda bergantung pada runtime lain, Anda juga dapat menentukan runtime dependen dalam file buildspec. Jika Anda tidak menentukan runtime apa pun dalam file buildspec, pilih runtime CodeBuild default yang tersedia dalam gambar yang Anda gunakan. Jika Anda menentukan satu atau lebih runtime, hanya CodeBuild menggunakan runtime tersebut. Jika runtime dependen tidak ditentukan, CodeBuild coba pilih runtime dependen untuk Anda.

> **Note**
>
> Jika versi runtime tidak ditentukan, CodeBuild gunakan versi default. Versi default dapat berubah ketika versi default sebelumnya mencapai akhir masa pakai (EOL). Untuk menghindari perubahan tak terduga pada lingkungan build, sebaiknya tentukan versi runtime di file buildspec.

Jika dua runtime tertentu bertentangan, build gagal. Misalnya, `android: 29` dan `java: openjdk11` konflik, jadi jika keduanya ditentukan, build gagal.

Untuk informasi selengkapnya tentang runtime yang tersedia, lihatRuntime yang tersedia.

> **Note**
>
> Jika Anda menentukan `runtime-versions` bagian dan menggunakan gambar selain Ubuntu Standard Image 2.0 atau yang lebih baru, atau gambar standar Amazon Linux 2 (AL2) 1.0 atau yang lebih baru, build mengeluarkan peringatan, "`Skipping install of runtimes. Runtime version selection is not supported by this build image`."

phases/install/perintah

Urutan opsional. Berisi urutan skalar, di mana setiap skalar mewakili satu perintah yang CodeBuild berjalan selama instalasi. CodeBuild menjalankan setiap perintah, satu per satu, dalam urutan yang tercantum, dari awal hingga akhir.

fase/pra_build

Urutan opsional. Merupakan perintah, jika ada, yang CodeBuild berjalan sebelum build. Misalnya, Anda dapat menggunakan fase ini untuk masuk ke Amazon ECR, atau Anda dapat menginstal dependensi npm.

phases/pre_build/perintah

Urutan yang `pre_build` diperlukan jika ditentukan. Berisi urutan skalar, di mana setiap skalar mewakili satu perintah yang CodeBuild berjalan sebelum build. CodeBuildmenjalankan setiap perintah, satu per satu, dalam urutan yang tercantum, dari awal hingga akhir.

fase/membangun

Urutan opsional. Merupakan perintah, jika ada, yang CodeBuild berjalan selama pembuatan. Misalnya, Anda dapat menggunakan fase ini untuk menjalankan Mocha, RSpec, atau sbt.

phases/build/perintah

Diperlukan `build` jika ditentukan. Berisi urutan skalar, di mana setiap skalar mewakili satu perintah yang CodeBuild berjalan selama pembuatan. CodeBuild menjalankan setiap perintah, satu per satu, dalam urutan yang tercantum, dari awal hingga akhir.

fase/post_build

Urutan opsional. Merupakan perintah, jika ada, yang CodeBuild berjalan setelah build. Misalnya, Anda dapat menggunakan Maven untuk mengemas artefak build ke dalam file JAR atau WAR, atau Anda dapat mendorong gambar Docker ke Amazon ECR. Kemudian Anda dapat mengirim pemberitahuan build melalui Amazon SNS.

phases/post_build/perintah

Diperlukan `post_build` jika ditentukan. Berisi urutan skalar, di mana setiap skalar mewakili satu perintah yang CodeBuild berjalan setelah build. CodeBuild menjalankan setiap perintah, satu per satu, dalam urutan yang tercantum, dari awal hingga akhir.

melaporkan

laporan-nama-kelompok-atau-arn

Urutan opsional. Menentukan grup laporan tempat laporan dikirim. Sebuah proyek dapat memiliki maksimal lima kelompok laporan. Tentukan ARN dari grup laporan yang ada, atau nama grup laporan baru. Jika Anda menentukan nama, CodeBuild buat grup laporan menggunakan nama proyek Anda dan nama yang Anda tentukan dalam format`<project-name>-<report-group-name>`. Nama grup laporan juga dapat diatur menggunakan variabel lingkungan di buildspec seperti. `$REPORT_GROUP_NAME` Untuk informasi selengkapnya, lihat Penamaan grup laporan.

laporan/<report-group>/file

Urutan yang diperlukan. Merupakan lokasi yang berisi data mentah hasil pengujian yang dihasilkan oleh laporan. Berisi urutan skalar, dengan setiap skalar mewakili lokasi terpisah di mana CodeBuild dapat menemukan file uji, relatif terhadap lokasi build asli atau, jika disetel,. `base-directory` Lokasi dapat mencakup yang berikut:

- Satu file (misalnya,`my-test-report-file.json`).
- Satu file dalam subdirektori (misalnya, *`my-subdirectory`*`/my-test-report-file.json` atau*`my-parent-subdirectory`*`/`*`my-subdirectory`*`/my-test-report-file.json`).
- `'**/*'`mewakili semua file secara rekursif.
- *`my-subdirectory`*`/*`mewakili semua file dalam subdirektori bernama*`my-subdirectory`*.
- *`my-subdirectory`*`/**/*`mewakili semua file secara rekursif mulai dari subdirektori bernama. *`my-subdirectory`*

laporan/<report-group>/file-format

Pemetaan opsional. Merupakan format file laporan. Jika tidak ditentukan, `JUNITXML` digunakan. Nilai ini tidak peka huruf besar/kecil. Kemungkinan nilainya adalah:

Laporan pengujian

`CUCUMBERJSON`

Mentimun JSON

`JUNITXML`

JUnit XML

`NUNITXML`

nUnit XML

`NUNIT3XML`

NUnit 3 XML

`TESTNGXML`

XML TestNG

`VISUALSTUDIOTRX`

Visual Studio TRX

Laporan cakupan kode

`CLOVERXML`

Clover XML

`COBERTURAXML`

Cobertura XML

`JACOCOXML`

JaCoCo XML

`SIMPLECOV`

SimpleCov JSON

> **Note**
>
> CodeBuild menerima laporan cakupan kode JSON yang dihasilkan oleh simplecov, bukan simplecov-json.

laporan/<report-group>/direktori dasar

Pemetaan opsional. Merupakan satu atau lebih direktori tingkat atas, relatif terhadap lokasi build asli, yang CodeBuild digunakan untuk menentukan di mana menemukan file pengujian mentah.

laporan/<report-group>/jalur buang

Tidak wajib. Menentukan apakah direktori file laporan diratakan dalam output. Jika ini tidak ditentukan, atau berisino, file laporan dikeluarkan dengan struktur direktori mereka utuh. Jika ini berisi`yes`, semua file uji ditempatkan di direktori output yang sama. Misalnya, jika jalur ke hasil tes adalah`com/myapp/mytests/TestResult.xml`, menentukan `yes` akan menempatkan file ini di`/TestResult.xml`.

artefak

Urutan opsional. Merupakan informasi tentang di mana CodeBuild dapat menemukan output build dan bagaimana mempersi CodeBuild apkannya untuk diunggah ke bucket keluaran S3. Urutan ini tidak diperlukan jika, misalnya, Anda membangun dan mendorong gambar Docker ke Amazon ECR, atau Anda menjalankan pengujian unit pada kode sumber Anda, tetapi tidak membangunnya.

> **Note**
>
> Metadata Amazon S3 memiliki CodeBuild header bernama `x-amz-meta-codebuild-buildarn` yang berisi `buildArn` CodeBuild build yang menerbitkan artefak ke Amazon S3. Dit `buildArn` ambahkan untuk memungkinkan pelacakan sumber untuk pemberitahuan dan referensi dari mana bangunan artefak dihasilkan.

artefak/file

Urutan yang diperlukan. Merupakan lokasi yang berisi artefak keluaran build di lingkungan build. Berisi urutan skalar, dengan setiap skalar mewakili lokasi terpisah di mana CodeBuild dapat menemukan artefak keluaran build, relatif terhadap lokasi build asli atau, jika disetel, direktori dasar. Lokasi dapat mencakup yang berikut:

- Satu file (misalnya,`my-file.jar`).
- Satu file dalam subdirektori (misalnya, *`my-subdirectory`*`/my-file.jar` atau*`my-parent-subdirectory`*`/`*`my-subdirectory`*`/my-file.jar`).
- `'**/*'`mewakili semua file secara rekursif.
- *`my-subdirectory`*`/*`mewakili semua file dalam subdirektori bernama*`my-subdirectory`*.
- *`my-subdirectory`*`/**/*`mewakili semua file secara rekursif mulai dari subdirektori bernama. *`my-subdirectory`*

Saat Anda menentukan lokasi artefak keluaran build, CodeBuild dapat menemukan lokasi build asli di lingkungan build. Anda tidak perlu menyiapkan lokasi keluaran artefak build Anda dengan jalur ke lokasi build asli atau menentukan `./` atau serupa. Jika Anda ingin mengetahui jalur ke lokasi ini, Anda dapat menjalankan perintah seperti `echo $CODEBUILD_SRC_DIR` selama build. Lokasi untuk setiap lingkungan build mungkin sedikit berbeda.

artefak/nama

Nama opsional. Menentukan nama untuk artefak build Anda. Nama ini digunakan jika salah satu dari berikut ini benar.

- Anda menggunakan CodeBuild API untuk membuat build dan `overrideArtifactName` bendera ditetapkan pada `ProjectArtifacts` objek saat proyek diperbarui, proyek dibuat, atau build dimulai.
- Anda menggunakan CodeBuild konsol untuk membuat build, nama ditentukan dalam file buildspec, dan Anda memilih Aktifkan versi semantik saat Anda membuat atau memperbarui proyek. Untuk informasi selengkapnya, lihat Buat proyek build (konsol).

Anda dapat menentukan nama dalam file buildspec yang dihitung pada waktu pembuatan. Nama yang ditentukan dalam file buildspec menggunakan bahasa perintah Shell. Misalnya, Anda dapat menambahkan tanggal dan waktu untuk nama artefak Anda sehingga selalu unik. Nama artefak yang unik mencegah artefak dituliskan. Untuk informasi selengkapnya, lihat bahasa perintah Shell.

- Ini adalah contoh nama artefak yang ditambahkan dengan tanggal artefak dibuat.

```
version: 0.2
phases:
  build:
    commands:
      - rspec HelloWorld_spec.rb
artifacts:
  files:
    - '**/*'
  name: myname-$(date +%Y-%m-%d)
```

- Ini adalah contoh nama artefak yang menggunakan variabel CodeBuild lingkungan. Untuk informasi selengkapnya, lihat Variabel lingkungan di lingkungan build.

```
version: 0.2
phases:
  build:
```

```
    commands:
      - rspec HelloWorld_spec.rb
artifacts:
  files:
    - '**/*'
  name: myname-$AWS_REGION
```

- Ini adalah contoh nama artefak yang menggunakan variabel CodeBuild lingkungan dengan tanggal pembuatan artefak ditambahkan padanya.

```
version: 0.2
phases:
  build:
    commands:
      - rspec HelloWorld_spec.rb
artifacts:
  files:
    - '**/*'
  name: $AWS_REGION-$(date +%Y-%m-%d)
```

Anda dapat menambahkan informasi jalur ke nama sehingga artefak bernama ditempatkan di direktori berdasarkan jalur dalam nama. Dalam contoh ini, artefak build ditempatkan di output di bawah`builds/<build number>/my-artifacts`.

```
version: 0.2
phases:
  build:
    commands:
      - rspec HelloWorld_spec.rb
artifacts:
  files:
    - '**/*'
  name: builds/$CODEBUILD_BUILD_NUMBER/my-artifacts
```

artefak/jalur buang

Tidak wajib. Menentukan apakah direktori artefak build diratakan dalam output. Jika ini tidak ditentukan, atau berisino, artefak build dikeluarkan dengan struktur direktorinya utuh. Jika ini berisi`yes`, semua artefak build ditempatkan di direktori output yang sama. Misalnya, jika jalur ke file dalam artefak keluaran build adalah`com/mycompany/app/HelloWorld.java`, menentukan `yes` akan menempatkan file ini di`/HelloWorld.java`.

artefak/direktori dasar

Pemetaan opsional. Merupakan satu atau lebih direktori tingkat atas, relatif terhadap lokasi build asli, yang CodeBuild digunakan untuk menentukan file dan subdirektori mana yang akan disertakan dalam artefak keluaran build. Nilai yang valid meliputi:

- Direktori tingkat atas tunggal (misalnya,`my-directory`).
- `'my-directory*'`mewakili semua direktori tingkat atas dengan nama yang dimulai dengan`my-directory`.

Direktori tingkat atas yang cocok tidak disertakan dalam artefak keluaran build, hanya file dan subdirektori mereka.

Anda dapat menggunakan `files` dan `discard-paths` untuk lebih membatasi file dan subdirektori mana yang disertakan. Misalnya, untuk struktur direktori berikut:

```
.
### my-build-1
#   ### my-file-1.txt
### my-build-2
    ### my-file-2.txt
    ### my-subdirectory
        ### my-file-3.txt
```

Dan untuk `artifacts` urutan berikut:

```
artifacts:
  files:
    - '*/my-file-3.txt'
  base-directory: my-build-2
```

Subdirektori dan file berikut akan disertakan dalam artefak keluaran build:

```
.
### my-subdirectory
    ### my-file-3.txt
```

Sedangkan untuk `artifacts` urutan berikut:

```
artifacts:
```

```
  files:
    - '**/*'
  base-directory: 'my-build*'
  discard-paths: yes
```

File-file berikut akan disertakan dalam artefak keluaran build:

```
.
### my-file-1.txt
### my-file-2.txt
### my-file-3.txt
```

artefak/mengecualikan jalur

Pemetaan opsional. Merupakan satu atau lebih jalur, relatif terhadap`base-directory`, yang CodeBuild akan dikecualikan dari artefak build. Karakter tanda bintang (`*`) cocok dengan karakter nol atau karakter lain dari komponen nama tanpa melintasi batas folder. Tanda bintang ganda (`**`) cocok dengan nol atau lebih karakter komponen nama di semua direktori.

Contoh jalur pengecualian meliputi:

- Untuk mengecualikan file dari semua direktori: "`**/`*`file-name`*`/**/*`"
- Untuk mengecualikan semua folder titik: "`**/.*/**/*`"
- Untuk mengecualikan semua file titik: "`**/.*`"

artefak/aktifkan symlink

Tidak wajib. Jika tipe output adalah`ZIP`, menentukan apakah tautan simbolik internal dipertahankan dalam file ZIP. Jika ini berisi`yes`, semua tautan simbolik internal di sumber akan dipertahankan dalam file ZIP artefak.

artefak/prefiks S3

Tidak wajib. Menentukan awalan yang digunakan saat artefak dikeluarkan ke bucket Amazon S3 dan jenis namespace adalah`BUILD_ID`. Saat digunakan, jalur keluaran di bucket adalah`<s3-prefix>/<build-id>/<name>.zip`.

artefak/artefak sekunder

Urutan opsional. Merupakan satu atau lebih definisi artefak sebagai pemetaan antara pengidentifikasi artefak dan definisi artefak. Setiap pengidentifikasi artefak di blok ini harus cocok

dengan artefak yang ditentukan dalam `secondaryArtifacts` atribut proyek Anda. Setiap definisi terpisah memiliki sintaks yang sama dengan `artifacts` blok di atas.

> **Note**
>
> `artifacts/files`Urutan selalu diperlukan, bahkan ketika hanya ada artefak sekunder yang ditentukan.

Misalnya, jika proyek Anda memiliki struktur berikut:

```
{
  "name": "sample-project",
  "secondaryArtifacts": [
    {
      "type": "S3",
      "location": "<output-bucket1>",
      "artifactIdentifier": "artifact1",
      "name": "secondary-artifact-name-1"
    },
    {
      "type": "S3",
      "location": "<output-bucket2>",
      "artifactIdentifier": "artifact2",
      "name": "secondary-artifact-name-2"
    }
  ]
}
```

Kemudian buildspec Anda terlihat seperti berikut:

```
version: 0.2

phases:
build:
  commands:
    - echo Building...
artifacts:
  files:
    - '**/*'
  secondary-artifacts:
    artifact1:
```

```
      files:
        - directory/file1
      name: secondary-artifact-name-1
    artifact2:
      files:
        - directory/file2
      name: secondary-artifact-name-2
```

cache

Urutan opsional. Merupakan informasi tentang di mana CodeBuild dapat mempersiapkan file untuk mengunggah cache ke bucket cache S3. Urutan ini tidak diperlukan jika jenis cache proyek adalahNo `Cache`.

cache/kunci

Urutan opsional. Merupakan kunci utama yang digunakan saat mencari atau mengembalikan cache. CodeBuild melakukan kecocokan yang tepat untuk kunci utama.

Berikut adalah contoh untuk kuncinya:

```
key: npm-key-$(codebuild-hash-files package-lock.json) }
```

cache/kunci fallback

Urutan opsional. Merupakan daftar kunci fallback yang digunakan secara berurutan ketika cache tidak dapat ditemukan menggunakan kunci primer. Hingga lima kunci fallback didukung, dan masing-masing dicocokkan menggunakan pencarian awalan. Urutan ini akan diabaikan jika kunci tidak disediakan.

Berikut adalah contoh untuk kunci fallback:

```
fallback-keys:
    - npm-key-$(codebuild-hash-files package-lock.json) }
    - npm-key-
    - npm-
```

cache/tindakan

Urutan opsional. Menentukan tindakan yang akan dilakukan pada cache. Nilai yang valid meliputi:

- `restore`yang hanya mengembalikan cache tanpa menyimpan pembaruan.
- `save`yang hanya menyimpan cache tanpa memulihkan versi sebelumnya.

Jika tidak ada nilai yang diberikan, CodeBuild default untuk melakukan pemulihan dan penyimpanan.

cache/jalur

Urutan yang diperlukan. Merupakan lokasi cache. Berisi urutan skalar, dengan setiap skalar mewakili lokasi terpisah di mana CodeBuild dapat menemukan artefak keluaran build, relatif terhadap lokasi build asli atau, jika disetel, direktori dasar. Lokasi dapat mencakup yang berikut:

- Satu file (misalnya,`my-file.jar`).
- Satu file dalam subdirektori (misalnya, *`my-subdirectory`*`/my-file.jar` atau*`my-parent-subdirectory`*`/`*`my-subdirectory`*`/my-file.jar`).
- `'**/*'`mewakili semua file secara rekursif.
- *`my-subdirectory`*`/*`mewakili semua file dalam subdirektori bernama*`my-subdirectory`*.
- *`my-subdirectory`*`/**/*`mewakili semua file secara rekursif mulai dari subdirektori bernama. *`my-subdirectory`*

> **Important**
>
> Karena deklarasi buildspec harus YAML yang valid, spasi dalam deklarasi buildspec penting. Jika jumlah spasi dalam deklarasi buildspec Anda tidak valid, build mungkin segera gagal. Anda dapat menggunakan validator YAML untuk menguji apakah deklarasi buildspec Anda adalah YAML yang valid.
> Jika Anda menggunakan AWS CLI, atau AWS SDK untuk mendeklarasikan buildspec saat Anda membuat atau memperbarui proyek build, buildspec harus berupa string tunggal yang dinyatakan dalam format YAML, bersama dengan spasi putih dan karakter escape baris baru yang diperlukan. Ada contoh di bagian selanjutnya.
> Jika Anda menggunakan AWS CodePipeline konsol CodeBuild or alih-alih file buildspec.yml, Anda dapat menyisipkan perintah untuk fase saja. `build` Alih-alih menggunakan sintaks sebelumnya, Anda mencantumkan, dalam satu baris, semua perintah yang ingin Anda jalankan selama fase build. Untuk beberapa perintah, pisahkan setiap perintah dengan && (misalnya,`mvn test && mvn package`).
> Anda dapat menggunakan CodePipeline konsol CodeBuild or alih-alih file buildspec.yml untuk menentukan lokasi artefak keluaran build di lingkungan build. Alih-alih menggunakan

sintaks sebelumnya, Anda mencantumkan, dalam satu baris, semua lokasi. Untuk beberapa lokasi, pisahkan setiap lokasi dengan koma (misalnya,`buildspec.yml, target/my-app.jar`).

Contoh Buildspec

Berikut adalah contoh file buildspec.yml.

```
version: 0.2

env:
  variables:
    JAVA_HOME: "/usr/lib/jvm/java-8-openjdk-amd64"
  parameter-store:
    LOGIN_PASSWORD: /CodeBuild/dockerLoginPassword

phases:
  install:
    commands:
      - echo Entered the install phase...
      - apt-get update -y
      - apt-get install -y maven
    finally:
      - echo This always runs even if the update or install command fails
  pre_build:
    commands:
      - echo Entered the pre_build phase...
      - docker login -u User -p $LOGIN_PASSWORD
    finally:
      - echo This always runs even if the login command fails
  build:
    commands:
      - echo Entered the build phase...
      - echo Build started on `date`
      - mvn install
    finally:
      - echo This always runs even if the install command fails
  post_build:
    commands:
      - echo Entered the post_build phase...
      - echo Build completed on `date`
```

```
reports:
  arn:aws:codebuild:your-region:your-aws-account-id:report-group/report-group-name-1:
    files:
      - "**/*"
    base-directory: 'target/tests/reports'
    discard-paths: no
  reportGroupCucumberJson:
    files:
      - 'cucumber/target/cucumber-tests.xml'
    discard-paths: yes
    file-format: CUCUMBERJSON # default is JUNITXML
artifacts:
  files:
    - target/messageUtil-1.0.jar
  discard-paths: yes
  secondary-artifacts:
    artifact1:
      files:
        - target/artifact-1.0.jar
      discard-paths: yes
    artifact2:
      files:
        - target/artifact-2.0.jar
      discard-paths: yes
cache:
  paths:
    - '/root/.m2/**/*'
```

Berikut adalah contoh dari buildspec sebelumnya, dinyatakan sebagai string tunggal, untuk digunakan dengan AWS CLI, atau SDK. AWS

```
"version: 0.2\n\nenv:\n  variables:\n    JAVA_HOME: \"/usr/lib/jvm/java-8-openjdk-
amd64\\"\n  parameter-store:\n    LOGIN_PASSWORD: /CodeBuild/dockerLoginPassword\n
 phases:\n\n  install:\n    commands:\n      - echo Entered the install phase...\n
 - apt-get update -y\n      - apt-get install -y maven\n    finally:\n      - echo This
 always runs even if the update or install command fails \n  pre_build:\n    commands:
\n      - echo Entered the pre_build phase...\n      - docker login -u User -p
 $LOGIN_PASSWORD\n    finally:\n      - echo This always runs even if the login command
 fails \n  build:\n    commands:\n      - echo Entered the build phase...\n      - echo
 Build started on `date`\n      - mvn install\n    finally:\n      - echo This always
 runs even if the install command fails\n  post_build:\n    commands:\n      - echo
 Entered the post_build phase...\n      - echo Build completed on `date`\n\n reports:
```

```
\n  reportGroupJunitXml:\n    files:\n      - \"**/*\"\n    base-directory: 'target/
tests/reports'\n    discard-paths: false\n  reportGroupCucumberJson:\n    files:\n
 - 'cucumber/target/cucumber-tests.xml'\n    file-format: CUCUMBERJSON\n\nartifacts:\n
 files:\n    - target/messageUtil-1.0.jar\n  discard-paths: yes\n  secondary-artifacts:
\n    artifact1:\n      files:\n       - target/messageUtil-1.0.jar\n      discard-
paths: yes\n    artifact2:\n      files:\n       - target/messageUtil-1.0.jar\n
 discard-paths: yes\n cache:\n  paths:\n    - '/root/.m2/**/*'"
```

Berikut adalah contoh perintah dalam `build` fase, untuk digunakan dengan CodePipeline konsol CodeBuild atau.

```
echo Build started on `date` && mvn install
```

Dalam contoh-contoh ini:

- Variabel lingkungan khusus, dalam teks biasa, dengan kunci `JAVA_HOME` dan nilai`/usr/lib/jvm/java-8-openjdk-amd64`, ditetapkan.
- Variabel lingkungan khusus bernama `dockerLoginPassword` Anda yang disimpan di Amazon EC2 Systems Manager Parameter Store akan direferensikan nanti dalam perintah build dengan menggunakan kunci`LOGIN_PASSWORD`.
- Anda tidak dapat mengubah nama fase build ini. Perintah yang dijalankan dalam contoh ini adalah `apt-get update -y` dan `apt-get install -y maven` (untuk menginstal Apache Maven), `mvn install` (untuk mengkompilasi, menguji, dan mengemas kode sumber ke dalam artefak keluaran build dan untuk menginstal artefak keluaran build di repositori internalnya), `docker login` (untuk masuk ke Docker dengan kata sandi yang sesuai dengan nilai variabel lingkungan khusus yang `dockerLoginPassword` Anda tetapkan di Amazon EC2 Systems Manager Parameter Store), dan beberapa perintah. `echo` Per `echo` intah disertakan di sini untuk menunjukkan bagaimana CodeBuild menjalankan perintah dan urutan menjalankannya.
- `files`mewakili file untuk diunggah ke lokasi keluaran build. Dalam contoh ini, mengun CodeBuild ggah file `messageUtil-1.0.jar` tunggal. `messageUtil-1.0.jar`File dapat ditemukan di direktori relatif yang dinamai `target` di lingkungan build. Karena `discard-paths: yes` ditentukan, `messageUtil-1.0.jar` diunggah langsung (dan bukan ke `target` direktori perantara). Nama file `messageUtil-1.0.jar` dan nama direktori relatif `target` didasarkan pada cara Apache Maven membuat dan menyimpan artefak keluaran build hanya untuk contoh ini. Dalam skenario Anda sendiri, nama file dan direktori ini akan berbeda.
- `reports`mewakili dua grup laporan yang menghasilkan laporan selama pembuatan:

- `arn:aws:codebuild:your-region:your-aws-account-id:report-group/report-group-name-1`menentukan ARN dari grup laporan. Hasil pengujian yang dihasilkan oleh kerangka pengujian ada di `target/tests/reports` direktori. Format file adalah `JunitXml` dan jalur tidak dihapus dari file yang berisi hasil pengujian.
- `reportGroupCucumberJson`menentukan grup laporan baru. Jika nama proyek adalah`my-project`, grup laporan dengan nama `my-project-reportGroupCucumberJson` dibuat saat build dijalankan. Hasil pengujian yang dihasilkan oleh kerangka pengujian ada di`cucumber/target/cucumber-tests.xml`. Format file pengujian adalah `CucumberJson` dan jalur dihapus dari file yang berisi hasil pengujian.

Versi Buildspec

Tabel berikut mencantumkan versi buildspec dan perubahan antar versi.

Versi	Perubahan
0.2	• `environment_variables` telah diganti namanya menjadi`env`. • `plaintext` telah diganti namanya menjadi`variables` . • Pro `type` perti untuk `artifacts` telah usang. • Di versi 0.1, AWS CodeBuild jalankan setiap perintah build dalam instance terpisah dari shell default di lingkungan build. Di versi 0.2, CodeBuild jalankan semua perintah build dalam contoh yang sama dari shell default di lingkungan build.
0.1	Ini adalah definisi awal dari format spesifikasi build.

Referensi buildspec build batch

Topik ini berisi referensi buildspec untuk properti build batch.

batch

Pemetaan opsional. Pengaturan pembuatan batch untuk proyek.

batch/cepat-gagal

Tidak wajib. Menentukan perilaku build batch ketika satu atau lebih tugas build gagal.

`false`

Nilai default. Semua build yang sedang berjalan akan selesai.

`true`

Semua build yang berjalan akan dihentikan ketika salah satu tugas build gagal.

Secara default, semua tugas build batch berjalan dengan pengaturan build seperti `env` dan`phases`, ditentukan dalam file buildspec. Anda dapat mengganti pengaturan build default dengan menentukan `env` nilai yang berbeda atau file buildspec yang berbeda dalam parameter. `batch/<batch-type>/buildspec`

Isi `batch` properti bervariasi berdasarkan jenis build batch yang ditentukan. Jenis build batch yang mungkin adalah:

- `batch/build-grafik`
- `batch/build-daftar`
- `batch/build-matriks`
- `batch/build-fanout`

batch/build-grafik

Mendefinisikan grafik build. Grafik build mendefinisikan serangkaian tugas yang memiliki dependensi pada tugas lain dalam batch. Untuk informasi selengkapnya, lihat Bangun grafik.

Elemen ini berisi array tugas membangun. Setiap tugas build berisi properti berikut.

pengidentifikasi

Wajib. Pengidentifikasi tugas.

spesifikasi bangunan

Tidak wajib. Path dan nama file dari file buildspec yang akan digunakan untuk tugas ini. Jika parameter ini tidak ditentukan, file buildspec saat ini digunakan.

sesi debug

Tidak wajib. Nilai Boolean yang menunjukkan apakah sesi debugging diaktifkan untuk build batch ini. Untuk informasi selengkapnya tentang sesi debugging, lihatDebug dibangun dengan Session Manager.

`false`

Debugging sesi dinonaktifkan.

`true`

Debugging sesi diaktifkan.

bergantung pada

Tidak wajib. Array pengenal tugas yang bergantung pada tugas ini. Tugas ini tidak akan berjalan sampai tugas-tugas ini selesai.

env

Tidak wajib. Lingkungan build menggantikan tugas tersebut. Ini dapat berisi properti berikut:

tipe komputasi

Pengidentifikasi tipe komputasi yang akan digunakan untuk tugas. Lihat ComputeType di the section called “Membangun mode dan tipe komputasi lingkungan” untuk kemungkinan nilai.

armada

Pengidentifikasi armada yang akan digunakan untuk tugas tersebut. Untuk informasi selengkapnya, lihat the section called “Jalankan build pada armada kapasitas yang dicadangkan”.

gambar

Pengidentifikasi gambar yang akan digunakan untuk tugas. Lihat Pengidentifikasi gambar di the section called “Gambar Docker disediakan oleh CodeBuild” untuk kemungkinan nilai.

mode istimewa

Nilai Boolean yang menunjukkan apakah akan menjalankan daemon Docker di dalam wadah Docker. Setel ke `true` hanya jika proyek build digunakan untuk membangun image Docker.

Jika tidak, build yang mencoba berinteraksi dengan daemon Docker akan gagal. Pengaturan default-nya adalah `false`.

jenis

Pengidentifikasi jenis lingkungan yang akan digunakan untuk tugas. Lihat Jenis lingkungan the section called “Membangun mode dan tipe komputasi lingkungan” untuk kemungkinan nilai.

variabel

Variabel lingkungan yang akan hadir di lingkungan build. Untuk informasi selengkapnya, lihat env/variabel .

> **Note**
>
> Perhatikan bahwa tipe komputasi dan armada tidak dapat disediakan dalam pengidentifikasi yang sama dari satu build.

abaikan kegagalan

Tidak wajib. Nilai Boolean yang menunjukkan jika kegagalan tugas build ini dapat diabaikan.

`false`

Nilai default. Jika tugas build ini gagal, pembuatan batch akan gagal.

`true`

Jika tugas build ini gagal, build batch masih bisa berhasil.

Berikut ini adalah contoh entri buildspec build graph:

```
batch:
  fast-fail: false
  build-graph:
    - identifier: build1
      env:
        variables:
          BUILD_ID: build1
      ignore-failure: false
    - identifier: build2
```

```
      buildspec: build2.yml
      env:
        variables:
          BUILD_ID: build2
      depend-on:
        - build1
    - identifier: build3
      env:
        variables:
          BUILD_ID: build3
      depend-on:
        - build2
    - identifier: build4
      env:
        compute-type: ARM_LAMBDA_1GB
    - identifier: build5
      env:
        fleet: fleet_name
```

batch/build-daftar

Mendefinisikan daftar build. Daftar build digunakan untuk menentukan sejumlah tugas yang berjalan secara paralel. Untuk informasi selengkapnya, lihat Buat daftar.

Elemen ini berisi array tugas membangun. Setiap tugas build berisi properti berikut.

pengidentifikasi

Wajib. Pengidentifikasi tugas.

spesifikasi bangunan

Tidak wajib. Path dan nama file dari file buildspec yang akan digunakan untuk tugas ini. Jika parameter ini tidak ditentukan, file buildspec saat ini digunakan.

sesi debug

Tidak wajib. Nilai Boolean yang menunjukkan apakah sesi debugging diaktifkan untuk build batch ini. Untuk informasi selengkapnya tentang sesi debugging, lihatDebug dibangun dengan Session Manager.

`false`

Debugging sesi dinonaktifkan.

`true`

Debugging sesi diaktifkan.

env

Tidak wajib. Lingkungan build menggantikan tugas tersebut. Ini dapat berisi properti berikut:

tipe komputasi

Pengidentifikasi tipe komputasi yang akan digunakan untuk tugas. Lihat ComputeType di the section called “Membangun mode dan tipe komputasi lingkungan” untuk kemungkinan nilai.

armada

Pengidentifikasi armada yang akan digunakan untuk tugas tersebut. Untuk informasi selengkapnya, lihat the section called “Jalankan build pada armada kapasitas yang dicadangkan”.

gambar

Pengidentifikasi gambar yang akan digunakan untuk tugas. Lihat Pengidentifikasi gambar di the section called “Gambar Docker disediakan oleh CodeBuild” untuk kemungkinan nilai.

mode istimewa

Nilai Boolean yang menunjukkan apakah akan menjalankan daemon Docker di dalam wadah Docker. Setel ke `true` hanya jika proyek build digunakan untuk membangun image Docker. Jika tidak, build yang mencoba berinteraksi dengan daemon Docker akan gagal. Pengaturan default-nya adalah `false`.

jenis

Pengidentifikasi jenis lingkungan yang akan digunakan untuk tugas. Lihat Jenis lingkungan the section called “Membangun mode dan tipe komputasi lingkungan” untuk kemungkinan nilai.

variabel

Variabel lingkungan yang akan hadir di lingkungan build. Untuk informasi selengkapnya, lihat env/variabel .

> **Note**
>
> Perhatikan bahwa tipe komputasi dan armada tidak dapat disediakan dalam pengidentifikasi yang sama dari satu build.

abaikan kegagalan

Tidak wajib. Nilai Boolean yang menunjukkan jika kegagalan tugas build ini dapat diabaikan.

`false`

Nilai default. Jika tugas build ini gagal, pembuatan batch akan gagal.

`true`

Jika tugas build ini gagal, build batch masih bisa berhasil.

Berikut ini adalah contoh entri buildspec daftar build:

```
batch:
  fast-fail: false
  build-list:
    - identifier: build1
      env:
        variables:
          BUILD_ID: build1
      ignore-failure: false
    - identifier: build2
      buildspec: build2.yml
      env:
        variables:
          BUILD_ID: build2
      ignore-failure: true
    - identifier: build3
      env:
        compute-type: ARM_LAMBDA_1GB
    - identifier: build4
      env:
        fleet: fleet_name
    - identifier: build5
      env:
        compute-type: GENERAL_LINUX_XLAGRE
```

batch/build-matriks

Mendefinisikan matriks build. Matriks build mendefinisikan tugas dengan konfigurasi berbeda yang berjalan secara paralel. CodeBuild membuat build terpisah untuk setiap kombinasi konfigurasi yang mungkin. Untuk informasi selengkapnya, lihat Bangun matriks.

statis

Properti statis berlaku untuk semua tugas build.

abaikan kegagalan

Tidak wajib. Nilai Boolean yang menunjukkan jika kegagalan tugas build ini dapat diabaikan.

`false`

Nilai default. Jika tugas build ini gagal, pembuatan batch akan gagal.

`true`

Jika tugas build ini gagal, build batch masih bisa berhasil.

env

Tidak wajib. Lingkungan build menggantikan semua tugas.

mode istimewa

Nilai Boolean yang menunjukkan apakah akan menjalankan daemon Docker di dalam wadah Docker. Setel ke `true` hanya jika proyek build digunakan untuk membangun image Docker. Jika tidak, build yang mencoba berinteraksi dengan daemon Docker akan gagal. Pengaturan default-nya adalah `false`.

jenis

Pengidentifikasi jenis lingkungan yang akan digunakan untuk tugas. Lihat Jenis lingkungan the section called “Membangun mode dan tipe komputasi lingkungan” untuk kemungkinan nilai.

dinamis

Properti dinamis menentukan matriks build.

spesifikasi bangunan

Tidak wajib. Array yang berisi path dan nama file dari file buildspec yang akan digunakan untuk tugas-tugas ini. Jika parameter ini tidak ditentukan, file buildspec saat ini digunakan.

env

Tidak wajib. Lingkungan build menggantikan tugas-tugas ini.

tipe komputasi

Array yang berisi pengidentifikasi tipe komputasi untuk digunakan untuk tugas-tugas ini. Lihat ComputeType di the section called “Membangun mode dan tipe komputasi lingkungan” untuk kemungkinan nilai.

gambar

Array yang berisi pengidentifikasi gambar untuk digunakan untuk tugas-tugas ini. Lihat Pengidentifikasi gambar di the section called “Gambar Docker disediakan oleh CodeBuild” untuk kemungkinan nilai.

variabel

Array yang berisi variabel lingkungan yang akan hadir di lingkungan build untuk tugas-tugas ini. Untuk informasi selengkapnya, lihat env/variabel .

Berikut ini adalah contoh entri buildspec build matrix:

```
batch:
  build-matrix:
    static:
      ignore-failure: false
    dynamic:
      buildspec:
        - matrix1.yml
        - matrix2.yml
      env:
        variables:
          MY_VAR:
            - VALUE1
            - VALUE2
            - VALUE3
```

Untuk informasi selengkapnya, lihat Bangun matriks.

batch/build-fanout

Mendefinisikan fan out build. Fanout build digunakan untuk mendefinisikan tugas yang dibagi menjadi beberapa build yang berjalan secara paralel. Untuk informasi selengkapnya, lihat Jalankan tes paralel dalam build batch.

Elemen ini berisi tugas build yang dapat dibagi menjadi beberapa build. `build-fanout`Bagian ini berisi properti berikut.

paralelisme

Wajib. Jumlah build yang akan menjalankan pengujian secara paralel.

abaikan kegagalan

Tidak wajib. Nilai boolean yang menunjukkan jika kegagalan dalam salah satu tugas build fanout dapat diabaikan. Nilai ignore-failure ini akan diterapkan ke semua build fanout.

palsu

Nilai default. Jika ada tugas pembuatan fanout yang gagal, pembuatan batch akan gagal.

benar

Jika ada tugas pembuatan fanout yang gagal, pembuatan batch masih bisa berhasil.

Berikut ini adalah contoh entri buildspec build fanout:

```
version: 0.2

batch:
   fast-fail: false
   build-fanout:
     parallelism: 5
     ignore-failure: false

phases:
  install:
    commands:
      - npm install
   build:
    commands:
      - mkdir -p test-results
      - cd test-results
      - |
        codebuild-tests-run \
         --test-command 'npx jest --runInBand --coverage' \
         --files-search "codebuild-glob-search '**/test/**/*.test.js'" \
         --sharding-strategy 'equal-distribution'
```

Lihat informasi yang lebih lengkap di Bangun fanout dan Gunakan perintah `codebuild-` tests-run CLI.

Buat referensi lingkungan untuk AWS CodeBuild

Saat Anda memang AWS CodeBuild gil untuk menjalankan build, Anda harus memberikan informasi tentang lingkungan build. Lingkungan build mewakili kombinasi sistem operasi, runtime bahasa pemrograman, dan alat CodeBuild yang digunakan untuk menjalankan build. Untuk informasi tentang cara kerja lingkungan build, lihatBagaimana cara CodeBuild kerja.

Lingkungan build berisi gambar Docker. Untuk informasi, lihat glosarium Docker di situs Docker Docker.

Saat Anda memberikan informasi CodeBuild tentang lingkungan build, Anda menentukan pengidentifikasi gambar Docker dalam jenis repositori yang didukung. Ini termasuk repositori gambar CodeBuild Docker, gambar yang tersedia untuk umum di Docker Hub, dan repositori Amazon Elastic Container Registry (Amazon ECR) yang dapat diakses AWS oleh akun Anda.

- Kami menyarankan Anda menggunakan gambar Docker yang disimpan di repositori gambar CodeBuild Docker, karena mereka dioptimalkan untuk digunakan dengan layanan. Untuk informasi selengkapnya, lihat Gambar Docker disediakan oleh CodeBuild.
- Untuk mendapatkan pengenal gambar Docker yang tersedia untuk umum yang disimpan di Docker Hub, lihat Men cari Repositori di situs Docker Docker.
- Untuk mempelajari cara bekerja dengan gambar Docker yang disimpan di repositori Amazon ECR di AWS akun Anda, lihat. Sampel Amazon ECR

Selain pengidentifikasi gambar Docker, Anda juga menentukan sekumpulan sumber daya komputasi yang digunakan lingkungan build. Untuk informasi selengkapnya, lihat Membangun mode dan tipe komputasi lingkungan.

Topik

- Gambar Docker disediakan oleh CodeBuild
- Membangun mode dan tipe komputasi lingkungan
- Shell dan perintah di lingkungan build
- Variabel lingkungan di lingkungan build
- Tugas latar belakang di lingkungan build

Gambar Docker disediakan oleh CodeBuild

Gambar yang didukung adalah versi utama terbaru dari gambar yang tersedia di CodeBuild dan diperbarui dengan pembaruan versi minor dan patch. CodeBuild mengoptimalkan durasi penyediaan build dengan gambar yang didukung dengan menyimpan gambar tersebut di Amazon Machine Images (AMI) mesin. Jika Anda ingin memanfaatkan caching dan meminimalkan durasi penyediaan build Anda, pilih Sel alu gunakan gambar terbaru untuk versi runtime ini di bagian Versi gambar CodeBuild konsol alih-alih versi yang lebih terperinci, seperti. `aws/codebuild/amazonlinux-x86_64-standard:4.0-1.0.0`

Topik

- Dapatkan daftar gambar Docker saat ini
- Gambar komputasi EC2
- Gambar komputasi Lambda
- Gambar yang tidak digunakan lagi CodeBuild
- Runtime yang tersedia
- Versi waktu aktif

Dapatkan daftar gambar Docker saat ini

CodeBuild sering memperbarui daftar gambar Docker untuk menambahkan gambar terbaru dan menghentikan gambar lama. Untuk mendapatkan daftar terbaru, lakukan salah satu hal berikut:

- Di CodeBuild konsol, di panduan Buat proyek build atau halaman Edit Build Project, untuk gambar Lingkungan, pilih Gambar ter kelola. Pilih dari daftar drop-down Sistem Oper asi, Runtime, dan versi Runtime. Untuk informasi selengkapnya, lihat Buat proyek build (konsol) atau Mengubah setelan proyek build (konsol).
- Untuk AWS CLI, jalankan `list-curated-environment-images` perintah:

```
aws codebuild list-curated-environment-images
```

- Untuk AWS SDK, panggil `ListCuratedEnvironmentImages` operasi untuk bahasa pemrograman target Anda. Untuk informasi selengkapnya, lihat AWS Referensi SDK dan alat.

Gambar komputasi EC2

AWS CodeBuild mendukung gambar Docker berikut yang tersedia untuk komputasi EC2 di. CodeBuild

> **Note**
>
> Gambar dasar platform Windows Server Core 2019 hanya tersedia di wilayah berikut:
>
> - Timur AS (N. Virginia)
> - AS Timur (Ohio)
> - AS Barat (Oregon)
> - Eropa (Irlandia)

Platform	Pengidentifikasi gambar	Definisi
Amazon Linux 2	`aws/codebuild/amazonlinux-x86_64-standard:4.0`	al/standard/4.0
Amazon Linux 2023	`aws/codebuild/amazonlinux-x86_64-standard:5.0`	al/standard/5.0
Amazon Linux 2023	`aws/codebuild/amazonlinux-x86_64-standard:6.0`	al/standard/6,0
Amazon Linux 2	`aws/codebuild/amazonlinux-x86_64-standard:corretto8`	al/standard/mail8
Amazon Linux 2	`aws/codebuild/amazonlinux-x86_64-standard:corretto11`	al/standard/mail11

Platform	Pengidentifikasi gambar	Definisi
Amazon Linux 2	`aws/codebuild/amazonlinux-aarch64-standard:2.0`	al/aarch64/standard/2.0
Amazon Linux 2023	`aws/codebuild/amazonlinux-aarch64-standard:3.0`	al/aarch64/standard/3.0
Ubuntu 20.04	`aws/codebuild/standard:5.0`	ubuntu/standard/5.0
Ubuntu 22.04	`aws/codebuild/standard:6.0`	ubuntu/standard/6,0
Ubuntu 22.04	`aws/codebuild/standard:7.0`	ubuntu/standard/7.0
Ubuntu 24.04	`aws/codebuild/standard:8.0`	ubuntu/standard/8.0
Inti Windows Server 2019	`aws/codebuild/windows-base:2019-1.0`	N/A
Inti Windows Server 2019	`aws/codebuild/windows-base:2019-2.0`	N/A
Inti Windows Server 2019	`aws/codebuild/windows-base:2019-3.0`	N/A
Inti Server Windows 2022	`aws/codebuild/windows-base:2022-1.0`	N/A

> **Note**
>
> Pada 22 November 2024, alias untuk gambar runtime Linux-based standar diperbarui dari `amazonlinux2` keamazonlinux. Tidak diperlukan pembaruan manual karena alias sebelumnya masih valid.

Gambar komputasi Lambda

AWS CodeBuild mendukung gambar Docker berikut yang tersedia untuk AWS Lambda komputasi di. CodeBuild

> **Important**
>
> A `on-failure` tribut tidak didukung saat menggunakan komputasi Lambda atau kapasitas cadangan. Atribut ini hanya berfungsi dengan gambar komputasi EC2 yang disediakan oleh. CodeBuild

`aarch64` Arsitektur

Platform	Pengidentifikasi gambar	Definisi
Amazon Linux 2	`aws/codebuild/amazonlinux-aarch64-lambda-standard:dotnet6`	al- lambda/aarch64 /dotnet6
Amazon Linux 2023	`aws/codebuild/amazonlinux-aarch64-lambda-standard:dotnet8`	al- lambda/aarch64 /dotnet8
Amazon Linux 2023	`aws/codebuild/amazonlinux-aarch64-lambda-standard:dotnet10`	al- lambda/aarch64 /dotnet10

Platform	Pengidentifikasi gambar	Definisi
Amazon Linux 2	`aws/codebuild/amazonlinux-aarch64-lambda-standard:go1.21`	al- lambda/aarch64 /go1,21
Amazon Linux 2023	`aws/codebuild/amazonlinux-aarch64-lambda-standard:go1.24`	al- lambda/aarch64 /go1,24
Amazon Linux 2023	`aws/codebuild/amazonlinux-aarch64-lambda-standard:go1.25`	al- lambda/aarch64 /go1,25
Amazon Linux 2	`aws/codebuild/amazonlinux-aarch64-lambda-standard:corretto11`	al- lambda/aarch64 /corect11
Amazon Linux 2	`aws/codebuild/amazonlinux-aarch64-lambda-standard:corretto17`	al- lambda/aarch64 /corect17
Amazon Linux 2023	`aws/codebuild/amazonlinux-aarch64-lambda-standard:corretto21`	al- lambda/aarch64 /korretto21
Amazon Linux 2023	`aws/codebuild/amazonlinux-aarch64-lambda-standard:corretto25`	al- lambda/aarch64 /corect25

Platform	Pengidentifikasi gambar	Definisi
Amazon Linux 2	`aws/codebuild/amazonlinux-aarch64-lambda-standard:nodejs18`	al- lambda/aarch64 /nodejs18
Amazon Linux 2023	`aws/codebuild/amazonlinux-aarch64-lambda-standard:nodejs20`	al- lambda/aarch64 /nodejs20
Amazon Linux 2023	`aws/codebuild/amazonlinux-aarch64-lambda-standard:nodejs22`	al- lambda/aarch64 /nodejs22
Amazon Linux 2023	`aws/codebuild/amazonlinux-aarch64-lambda-standard:nodejs24`	al- lambda/aarch64 /nodejs24
Amazon Linux 2	`aws/codebuild/amazonlinux-aarch64-lambda-standard:python3.11`	al- lambda/aarch64 /python3.11
Amazon Linux 2023	`aws/codebuild/amazonlinux-aarch64-lambda-standard:python3.12`	al- lambda/aarch64 /python3.12
Amazon Linux 2023	`aws/codebuild/amazonlinux-aarch64-lambda-standard:python3.13`	al- lambda/aarch64 /python3.13

Platform	Pengidentifikasi gambar	Definisi
Amazon Linux 2023	`aws/codebuild/amazonlinux-aarch64-lambda-standard:python3.14`	al- lambda/aarch64 /python3, 14
Amazon Linux 2	`aws/codebuild/amazonlinux-aarch64-lambda-standard:ruby3.2`	al- lambda/aarch64 /ruby3.2
Amazon Linux 2023	`aws/codebuild/amazonlinux-aarch64-lambda-standard:ruby3.4`	al- lambda/aarch64 /ruby3.4

`x86_64 Arsitektur`

Platform	Pengidentifikasi gambar	Definisi
Amazon Linux 2	`aws/codebuild/amazonlinux-x86_64-lambda-standard:dotnet6`	al- lambda/x86_64 /dotnet6
Amazon Linux 2023	`aws/codebuild/amazonlinux-x86_64-lambda-standard:dotnet8`	al- lambda/x86_64 /dotnet8
Amazon Linux 2023	`aws/codebuild/amazonlinux-x86_64-lambda-standard:dotnet10`	al- lambda/x86_64 /dotnet10

Platform	Pengidentifikasi gambar	Definisi
Amazon Linux 2	`aws/codebuild/amazonlinux-x86_64-lambda-standard:go1.21`	al- lambda/x86_64 /go1,21
Amazon Linux 2023	`aws/codebuild/amazonlinux-x86_64-lambda-standard:go1.24`	al- lambda/x86_64 /go1,24
Amazon Linux 2023	`aws/codebuild/amazonlinux-x86_64-lambda-standard:go1.25`	al- lambda/x86_64 /go1,25
Amazon Linux 2	`aws/codebuild/amazonlinux-x86_64-lambda-standard:corretto11`	al- lambda/x86_64 /corect11
Amazon Linux 2	`aws/codebuild/amazonlinux-x86_64-lambda-standard:corretto17`	al- lambda/x86_64 /corect17
Amazon Linux 2023	`aws/codebuild/amazonlinux-x86_64-lambda-standard:corretto21`	al- lambda/x86_64 /korretto21
Amazon Linux 2023	`aws/codebuild/amazonlinux-x86_64-lambda-standard:corretto25`	al- lambda/x86_64 /corect25
Amazon Linux 2	`aws/codebuild/amazonlinux-x86_64-lambda-standard:nodejs18`	al- lambda/x86_64 /nodejs18

Platform	Pengidentifikasi gambar	Definisi
Amazon Linux 2023	`aws/codebuild/amazonlinux-x86_64-lambda-standard:nodejs20`	al- lambda/x86_64 /nodejs20
Amazon Linux 2023	`aws/codebuild/amazonlinux-x86_64-lambda-standard:nodejs22`	al- lambda/x86_64 /nodejs22
Amazon Linux 2023	`aws/codebuild/amazonlinux-x86_64-lambda-standard:nodejs24`	al- lambda/x86_64 /nodejs24
Amazon Linux 2	`aws/codebuild/amazonlinux-x86_64-lambda-standard:python3.11`	al- lambda/x86_64 /python3.11
Amazon Linux 2023	`aws/codebuild/amazonlinux-x86_64-lambda-standard:python3.12`	al- lambda/x86_64 /python3.12
Amazon Linux 2023	`aws/codebuild/amazonlinux-x86_64-lambda-standard:python3.13`	al- lambda/x86_64 /python3.13
Amazon Linux 2023	`aws/codebuild/amazonlinux-x86_64-lambda-standard:python3.14`	al- lambda/x86_64 /python3,14

Platform	Pengidentifikasi gambar	Definisi
Amazon Linux 2	`aws/codebuild/amazonlinux-x86_64-lambda-standard:ruby3.2`	al- lambda/x86_64 /ruby3.2
Amazon Linux 2023	`aws/codebuild/amazonlinux-x86_64-lambda-standard:ruby3.4`	al- lambda/x86_64 /ruby3.4

Gambar yang tidak digunakan lagi CodeBuild

Gambar yang tidak digunakan lagi adalah gambar yang tidak lagi di-cache atau diperbarui oleh. CodeBuild Gambar yang tidak digunakan lagi tidak lagi menerima pembaruan versi minor atau pembaruan versi patch, dan karena tidak lagi diperbarui, penggunaannya mungkin tidak aman. Jika CodeBuild proyek Anda dikonfigurasi untuk menggunakan versi gambar yang lebih lama, proses penyediaan akan mengunduh image docker ini dan menggunakannya untuk membuat lingkungan runtime dalam wadah, yang dapat meningkatkan durasi penyediaan dan durasi build secara keseluruhan.

CodeBuild telah menghentikan gambar Docker berikut. Anda masih dapat menggunakan gambar-gambar ini, tetapi mereka tidak akan di-cache di host build dan akan menghasilkan waktu penyediaan yang lebih tinggi.

Platform	Pengidentifikasi gambar	Definisi	Tanggal pengusangan
Amazon Linux 2	`aws/codebuild/amazonlinux2-x86_64-standard:3.0`	al2/standard/3.0	9 Mei 2023
Ubuntu 18.04	`aws/codebuild/standard:4.0`	ubuntu/standard/4.0	31 Maret 2023

Platform	Pengidentifikasi gambar	Definisi	Tanggal pengusangan
Amazon Linux 2	`aws/codebuild/amazonlinux2-aarch64-standard:1.0`	al2/aarch64/standard/1.0	31 Maret 2023
Ubuntu 18.04	`aws/codebuild/standard:3.0`	ubuntu/standard/3.0	30 Juni 2022
Amazon Linux 2	`aws/codebuild/amazonlinux2-x86_64-standard:2.0`	al2/standard/2.0	30 Juni 2022

Topik

- Runtime yang tersedia
- Versi waktu aktif

Runtime yang tersedia

Anda dapat menentukan satu atau lebih runtime di `runtime-versions` bagian file buildspec Anda. Jika runtime Anda bergantung pada runtime lain, Anda juga dapat menentukan runtime dependen dalam file buildspec. Jika Anda tidak menentukan runtime apa pun dalam file buildspec, pilih runtime CodeBuild default yang tersedia dalam gambar yang Anda gunakan. Jika Anda menentukan satu atau lebih runtime, hanya CodeBuild menggunakan runtime tersebut. Jika runtime dependen tidak ditentukan, CodeBuild coba pilih runtime dependen untuk Anda.

> **Note**
>
> Jika versi runtime tidak ditentukan, CodeBuild gunakan versi default. Versi default dapat berubah ketika versi default sebelumnya mencapai akhir masa pakai (EOL). Untuk menghindari perubahan tak terduga pada lingkungan build, sebaiknya tentukan versi runtime di file buildspec.

Untuk informasi selengkapnya, lihat Specify runtime versions in the buildspec file.

Topik

- Runtime gambar Linux
- Runtime gambar macOS
- Waktu proses gambar Windows

Runtime gambar Linux

Tabel berikut berisi runtime yang tersedia dan gambar Linux standar yang mendukungnya.

Runtime platform Ubuntu dan Amazon Linux

Nama runtime	Versi	Image
dotnet	6.0	Amazon Linux 2 x86_64 Lambda standar: dotnet6 Amazon Linux 2 AArch64 Lambda standar: dotnet6 Amazon Linux 2023 x86_64 standar:5.0 Amazon Linux 2023 AArch64 standar: 3.0 Ubuntu 22.04 standar:7.0
	8.0	Amazon Linux 2023 x86_64 standar:5.0 Amazon Linux 2023 x86_64 standar:6.0 Amazon Linux 2023 AArch64 standar: 3.0 Amazon Linux 2023 AArch64 standar: 4.0

Nama runtime	Versi	Image
		Ubuntu 22.04 standar:7.0 Ubuntu 24.04 standar:8.0
	10.0	Amazon Linux 2023 x86_64 standar:5.0 Amazon Linux 2023 x86_64 standar:6.0 Amazon Linux 2023 AArch64 standar: 3.0 Amazon Linux 2023 AArch64 standar: 4.0 Ubuntu 22.04 standar:7.0 Ubuntu 24.04 standar:8.0
Cucking	1.20	Amazon Linux 2023 x86_64 standar:5.0 Amazon Linux 2023 AArch64 standar: 3.0 Ubuntu 22.04 standar:7.0

Nama runtime	Versi	Image
	1.21	Amazon Linux 2 x86_64 Lambda standar: go1.21 Amazon Linux 2 AArch64 Lambda standar: go1.21 Amazon Linux 2023 x86_64 standar:5.0 Amazon Linux 2023 AArch64 standar: 3.0 Ubuntu 22.04 standar:7.0
	1.22	Amazon Linux 2023 x86_64 standar:5.0 Amazon Linux 2023 AArch64 standar: 3.0 Ubuntu 22.04 standar:7.0
	1.23	Amazon Linux 2023 x86_64 standar:5.0 Amazon Linux 2023 AArch64 standar: 3.0 Ubuntu 22.04 standar:7.0

Nama runtime	Versi	Image
	1.24	Amazon Linux 2023 x86_64 Standar Lambda: go1.24 Amazon Linux 2023 AArch64 Standar Lambda: go1.24 Amazon Linux 2023 x86_64 standar:5.0 Amazon Linux 2023 x86_64 standar:6.0 Amazon Linux 2023 AArch64 standar: 3.0 Amazon Linux 2023 AArch64 standar: 4.0 Ubuntu 22.04 standar:7.0 Ubuntu 24.04 standar:8.0
	1.25	Amazon Linux 2023 x86_64 standar:5.0 Amazon Linux 2023 x86_64 standar:6.0 Amazon Linux 2023 AArch64 standar: 3.0 Amazon Linux 2023 AArch64 standar: 4.0 Ubuntu 22.04 standar:7.0 Ubuntu 24.04 standar:8.0

Nama runtime	Versi	Image
	1.26	Amazon Linux 2023 x86_64 standar:5.0 Amazon Linux 2023 x86_64 standar:6.0 Amazon Linux 2023 AArch64 standar: 3.0 Amazon Linux 2023 AArch64 standar: 4.0 Ubuntu 22.04 standar:7.0 Ubuntu 24.04 standar:8.0
java	benar8	Amazon Linux 2 x86_64 standar: corretto8 Amazon Linux 2023 x86_64 standar:5.0 Amazon Linux 2023 x86_64 standar:6.0 Amazon Linux 2023 AArch64 standar: 3.0 Amazon Linux 2023 AArch64 standar: 4.0 Ubuntu 22.04 standar:7.0 Ubuntu 24.04 standar:8.0

Nama runtime	Versi	Image
	corretto11	Amazon Linux 2 x86_64 standar: corretto11 Amazon Linux 2 x86_64 Lambda standar: corretto11 Amazon Linux 2 AArch64 Lambda standar: corretto11 Amazon Linux 2023 x86_64 standar:5.0 Amazon Linux 2023 x86_64 standar:6.0 Amazon Linux 2023 AArch64 standar: 3.0 Amazon Linux 2023 AArch64 standar: 4.0 Ubuntu 22.04 standar:7.0 Ubuntu 24.04 standar:8.0

Nama runtime	Versi	Image
	corretto17	Amazon Linux 2 x86_64 Lambda standar: corretto17 Amazon Linux 2 AArch64 Lambda standar: corretto17 Amazon Linux 2023 x86_64 standar:5.0 Amazon Linux 2023 x86_64 standar:6.0 Amazon Linux 2023 AArch64 standar: 3.0 Amazon Linux 2023 AArch64 standar: 4.0 Ubuntu 22.04 standar:7.0 Ubuntu 24.04 standar:8.0

Nama runtime	Versi	Image
	corretto21	Amazon Linux 2 x86_64 Lambda standar: corretto21 Amazon Linux 2 AArch64 Lambda standar: corretto21 Amazon Linux 2023 x86_64 standar:5.0 Amazon Linux 2023 x86_64 standar:6.0 Amazon Linux 2023 AArch64 standar: 3.0 Amazon Linux 2023 AArch64 standar: 4.0 Ubuntu 22.04 standar:7.0 Ubuntu 24.04 standar:8.0
	corretto25	Amazon Linux 2023 x86_64 standar:5.0 Amazon Linux 2023 x86_64 standar:6.0 Amazon Linux 2023 AArch64 standar: 3.0 Amazon Linux 2023 AArch64 standar: 4.0 Ubuntu 22.04 standar:7.0 Ubuntu 24.04 standar:8.0

Nama runtime	Versi	Image
nodejs	18	Amazon Linux 2 x86_64 Lambda standar: nodejs18 Amazon Linux 2 AArch64 Lambda standar: nodejs18 Amazon Linux 2023 x86_64 standar:5.0 Amazon Linux 2023 AArch64 standar: 3.0 Ubuntu 22.04 standar:7.0
	20	Amazon Linux 2 x86_64 Lambda standar: nodejs20 Amazon Linux 2 AArch64 Lambda standar: nodejs20 Amazon Linux 2023 x86_64 standar:5.0 Amazon Linux 2023 AArch64 standar: 3.0 Ubuntu 22.04 standar:7.0

Nama runtime	Versi	Image
	22	Amazon Linux 2023 x86_64 Standar Lambda: nodejs22 Amazon Linux 2023 AArch64 Standar Lambda: nodejs22 Amazon Linux 2023 x86_64 standar:5.0 Amazon Linux 2023 x86_64 standar:6.0 Amazon Linux 2023 AArch64 standar: 3.0 Amazon Linux 2023 AArch64 standar: 4.0 Ubuntu 22.04 standar:7.0 Ubuntu 24.04 standar:8.0
	24	Amazon Linux 2023 x86_64 standar:5.0 Amazon Linux 2023 x86_64 standar:6.0 Amazon Linux 2023 AArch64 standar: 3.0 Amazon Linux 2023 AArch64 standar: 4.0 Ubuntu 22.04 standar:7.0 Ubuntu 24.04 standar:8.0

Nama runtime	Versi	Image
	26	Amazon Linux 2023 x86_64 standar:5.0 Amazon Linux 2023 x86_64 standar:6.0 Ubuntu 22.04 standar:7.0 Ubuntu 24.04 standar:8.0
php	8.1	Amazon Linux 2023 AArch64 standar: 3.0
	8.2	Amazon Linux 2023 x86_64 standar:5.0 Amazon Linux 2023 x86_64 standar:6.0 Amazon Linux 2023 AArch64 standar: 3.0 Amazon Linux 2023 AArch64 standar: 4.0 Ubuntu 22.04 standar:7.0 Ubuntu 24.04 standar:8.0

Nama runtime	Versi	Image
	8.3	Amazon Linux 2023 x86_64 standar:5.0 Amazon Linux 2023 x86_64 standar:6.0 Amazon Linux 2023 AArch64 standar: 3.0 Amazon Linux 2023 AArch64 standar: 4.0 Ubuntu 22.04 standar:7.0 Ubuntu 24.04 standar:8.0
	8.4	Amazon Linux 2023 x86_64 standar:5.0 Amazon Linux 2023 x86_64 standar:6.0 Amazon Linux 2023 AArch64 standar: 3.0 Amazon Linux 2023 AArch64 standar: 4.0 Ubuntu 22.04 standar:7.0 Ubuntu 24.04 standar:8.0

Nama runtime	Versi	Image
	8.5	Amazon Linux 2023 x86_64 standar:5.0 Amazon Linux 2023 x86_64 standar:6.0 Amazon Linux 2023 AArch64 standar: 3.0 Amazon Linux 2023 AArch64 standar: 4.0 Ubuntu 22.04 standar:7.0 Ubuntu 24.04 standar:8.0
python	3.9	Amazon Linux 2023 x86_64 standar:5.0 Amazon Linux 2023 AArch64 standar: 3.0 Ubuntu 22.04 standar:7.0
	3.10	Amazon Linux 2023 x86_64 standar:5.0 Amazon Linux 2023 x86_64 standar:6.0 Amazon Linux 2023 AArch64 standar: 3.0 Amazon Linux 2023 AArch64 standar: 4.0 Ubuntu 22.04 standar:7.0 Ubuntu 24.04 standar:8.0

Nama runtime	Versi	Image
	3.11	Amazon Linux 2 x86_64 Standar Lambda: python3.11 Amazon Linux 2 AArch64 Lambda standar: python3.11 Amazon Linux 2023 x86_64 standar:5.0 Amazon Linux 2023 x86_64 standar:6.0 Amazon Linux 2023 AArch64 standar: 3.0 Amazon Linux 2023 AArch64 standar: 4.0 Ubuntu 22.04 standar:7.0 Ubuntu 24.04 standar:8.0

Nama runtime	Versi	Image
	3.12	Amazon Linux 2 x86_64 Standar Lambda: python3.12 Amazon Linux 2 AArch64 Lambda standar: python3.12 Amazon Linux 2023 x86_64 standar:5.0 Amazon Linux 2023 x86_64 standar:6.0 Amazon Linux 2023 AArch64 standar: 3.0 Amazon Linux 2023 AArch64 standar: 4.0 Ubuntu 22.04 standar:7.0 Ubuntu 24.04 standar:8.0

Nama runtime	Versi	Image
	3.13	Amazon Linux 2023 x86_64 Standar Lambda: python3.13 Amazon Linux 2023 AArch64 Standar Lambda: python3.13 Amazon Linux 2023 x86_64 standar:5.0 Amazon Linux 2023 x86_64 standar:6.0 Amazon Linux 2023 AArch64 standar: 3.0 Amazon Linux 2023 AArch64 standar: 4.0 Ubuntu 22.04 standar:7.0 Ubuntu 24.04 standar:8.0
	3.14	Amazon Linux 2023 x86_64 standar:5.0 Amazon Linux 2023 x86_64 standar:6.0 Amazon Linux 2023 AArch64 standar: 3.0 Amazon Linux 2023 AArch64 standar: 4.0 Ubuntu 22.04 standar:7.0 Ubuntu 24.04 standar:8.0

Nama runtime	Versi	Image
rubi	3.1	Amazon Linux 2023 x86_64 standar:5.0 Amazon Linux 2023 AArch64 standar: 3.0 Ubuntu 22.04 standar:7.0
	3.2	Amazon Linux 2 x86_64 Standar Lambda: ruby3.2 Amazon Linux 2 AArch64 Lambda standar: ruby3.2 Amazon Linux 2023 x86_64 standar:5.0 Amazon Linux 2023 AArch64 standar: 3.0 Ubuntu 22.04 standar:7.0
	3.3	Amazon Linux 2023 x86_64 standar:5.0 Amazon Linux 2023 x86_64 standar:6.0 Amazon Linux 2023 AArch64 standar: 3.0 Amazon Linux 2023 AArch64 standar: 4.0 Ubuntu 22.04 standar:7.0 Ubuntu 24.04 standar:8.0

Nama runtime	Versi	Image
	3.4	Amazon Linux 2023 x86_64 Standar Lambda: ruby3.4 Amazon Linux 2023 AArch64 Standar Lambda: ruby3.4 Amazon Linux 2023 x86_64 standar:5.0 Amazon Linux 2023 x86_64 standar:6.0 Amazon Linux 2023 AArch64 standar: 3.0 Amazon Linux 2023 AArch64 standar: 4.0 Ubuntu 22.04 standar:7.0 Ubuntu 24.04 standar:8.0
	4.0	Amazon Linux 2023 x86_64 standar:5.0 Amazon Linux 2023 x86_64 standar:6.0 Amazon Linux 2023 AArch64 standar: 3.0 Amazon Linux 2023 AArch64 standar: 4.0 Ubuntu 22.04 standar:7.0 Ubuntu 24.04 standar:8.0

Nama runtime	Versi	Image
berkarat	1.94	Amazon Linux 2023 AArch64 standar: 3.0 Amazon Linux 2023 AArch64 standar: 4.0
	1.97	Amazon Linux 2023 x86_64 standar:6.0 Ubuntu 24.04 standar:8.0

Runtime gambar macOS

> **Important**
>
> Gambar yang CodeBuild dikuratori untuk build Mac berisi macOS dan Xcode yang sudah diinstal sebelumnya. Dengan menggunakan perangkat lunak Xcode, Anda mengakui, memahami, dan menyetujui Perjanjian Xcode dan Apple SDK. Jika Anda tidak menerima syarat dan ketentuan perjanjian, jangan gunakan perangkat lunak Xcode. Sebagai gantinya, berikan Amazon Machine Images (AMI) Anda sendiri. Untuk informasi selengkapnya, lihat Bagaimana cara mengonfigurasi armada macOS kapasitas cadangan?

Tabel berikut berisi runtime yang tersedia yang didukung oleh macOS.

Runtime platform macOS

Nama runtime	Versi	Image	Catatan tambahan
bash	3.2.57	basis lengan macos:15 basis lengan macos:26	
dentang	17.0.0	basis lengan macos:15	

Nama runtime	Versi	Image	Catatan tambahan
		basis lengan macos:26	
dot net sdk	8.0.416	basis lengan macos:15	
	8.0.417	basis lengan macos:26	
	10.0.101	basis lengan macos:15	
	10.0.102	basis lengan macos:26	
gcc	11.5.0	basis lengan macos:15 basis lengan macos:26	Tersedia dengan menggunakan `gcc-11` alias
	12.4.0	basis lengan macos:15 basis lengan macos:26	Tersedia dengan menggunakan `gcc-12` alias
	13.4.0	basis lengan macos:15 basis lengan macos:26	Tersedia dengan menggunakan `gcc-13` alias
	14.3.0	basis lengan macos:15 basis lengan macos:26	Tersedia dengan menggunakan `gcc-14` alias

Nama runtime	Versi	Image	Catatan tambahan
gnu	11.5.0	basis lengan macos:15 basis lengan macos:26	Tersedia dengan menggunakan `gfortran-11` alias
	12.4.0	basis lengan macos:15 basis lengan macos:26	Tersedia dengan menggunakan `gfortran-12` alias
	13.4.0	basis lengan macos:15 basis lengan macos:26	Tersedia dengan menggunakan `gfortran-13` alias
	14.3.0	basis lengan macos:15 basis lengan macos:26	Tersedia dengan menggunakan `gfortran-14` alias
Cucking	1.24.11	basis lengan macos:15 basis lengan macos:26	
	1.25.4	basis lengan macos:15 basis lengan macos:26	

Nama runtime	Versi	Image	Catatan tambahan
java	Corretto8	basis lengan macos:15 basis lengan macos:26	
	Corretto11	basis lengan macos:15 basis lengan macos:26	
	Corretto17	basis lengan macos:15 basis lengan macos:26	
	Corretto21	basis lengan macos:15 basis lengan macos:26	
	Corretto25	basis lengan macos:15 basis lengan macos:26	
kottlin	2.21	basis lengan macos:15	
	2.3.0	basis lengan macos:26	

Nama runtime	Versi	Image	Catatan tambahan
mono	6.14.1	basis lengan macos:15 basis lengan macos:26	
nodejs	20.19.6	basis lengan macos:15 basis lengan macos:26	
	22.21.1	basis lengan macos:15 basis lengan macos:26	
	24.11.1	basis lengan macos:15 basis lengan macos:26	
perla	5.34.1	basis lengan macos:15 basis lengan macos:26	
php	8.229	basis lengan macos:15	
	8.2.30	basis lengan macos:26	
	8.3.28	basis lengan macos:15	

Nama runtime	Versi	Image	Catatan tambahan
	8.3.30	basis lengan macos:26	
	8.4.15	basis lengan macos:15	
	8.4.17	basis lengan macos:26	
	8.5.0	basis lengan macos:15	
	8.5.2	basis lengan macos:26	
python	3.10.19	basis lengan macos:15 basis lengan macos:26	
	3.11.14	basis lengan macos:15 basis lengan macos:26	
	3.12.12	basis lengan macos:15 basis lengan macos:26	
	3.13.10	basis lengan macos:15 basis lengan macos:26	

Nama runtime	Versi	Image	Catatan tambahan
	3.14.1	basis lengan macos:15 basis lengan macos:26	
rubi	3.2.9	basis lengan macos:15 basis lengan macos:26	
	3.3.10	basis lengan macos:15 basis lengan macos:26	
	3.4.7	basis lengan macos:15 basis lengan macos:26	
berkarat	1.91.1	basis lengan macos:15	
	1.92.0	basis lengan macos:26	
cepat	6.2.1	basis lengan macos:15	
	6.2.3	basis lengan macos:26	
Xcode	26.1.1	basis lengan macos:15	

Nama runtime	Versi	Image	Catatan tambahan
	26.2	basis lengan macos:26	

Waktu proses gambar Windows

Gambar dasar Windows Server Core 2019 berisi runtime berikut.

Runtime platform Windows

Nama runtime	Standar Windows Server Core 2019: versi 1.0	Standar Windows Server Core 2019: versi 2.0	Standar Windows Server Core 2019: versi 3.0
dotnet	3.1 5.0	3.1 6.0 7.0	8.0
dot net sdk	3.1 5.0	3.1 6.0 7.0	8.0
Cucking	1.14	1.18	1.21 1.22 1.23
grado	6.7	7.6	8.12
java	Corretto11	Corretto11 Corretto17	Corretto8 Corretto11 Corretto17

Nama runtime	Standar Windows Server Core 2019: versi 1.0	Standar Windows Server Core 2019: versi 2.0	Standar Windows Server Core 2019: versi 3.0
			Corretto21
maven	3.6	3.8	3.9
nodejs	14.15	16.19	20.18 22.13
php	7.4	8.1	8.3 8.4
Powershell	7.1	7.2	7.4
python	3.8	3.10	3.10 3.11 3.12 3.13
rubi	2.7	3.1	3.2 3.3 3.4

Versi waktu aktif

Saat Anda menentukan runtime di `runtime-versions` bagian file buildspec Anda, Anda dapat menentukan versi tertentu, versi utama tertentu dan versi minor terbaru, atau versi terbaru. Tabel berikut mencantumkan runtime yang tersedia dan cara menentukannya. Tidak semua versi runtime tersedia di semua gambar. Pemilihan versi runtime juga tidak didukung untuk gambar kustom. Untuk informasi selengkapnya, lihat Runtime yang tersedia. Jika Anda ingin menginstal dan menggunakan versi runtime khusus alih-alih versi runtime pra-instal, lihatVersi runtime khusus.

Versi runtime platform Ubuntu dan Amazon Linux 2

Nama runtime	Versi	Versi spesifik	Major khusus dan versi minor terbaru	Versi Terbaru
dotnet	6.0	`dotnet: 6.0`	`dotnet: 6.x`	`dotnet: latest`
	8.0	`dotnet: 8.0`	`dotnet: 8.x`	
	10.0	`dotnet: 10.0`	`dotnet: 10.x`	
Cucking	1.20	`golang: 1.20`	`golang: 1.x`	`golang: latest`
	1.21	`golang: 1.21`		
	1.22	`golang: 1.22`		
	1.23	`golang: 1.23`		
	1.24	`golang: 1.24`		
	1.25	`golang: 1.25`		
	1.26	`golang: 1.26`		
java	benar8	`java: corretto`	`java: corretto .x`	`java: latest`
	corretto11	`java: corretto 1`	`java: corretto 1.x`	
	corretto17	`java: corretto 7`	`java: corretto 7.x`	
	corretto21	`java: corretto 1`	`java: corretto 1.x`	
	corretto25	`java: corretto 5`	`java: corretto 5.x`	

Nama runtime	Versi	Versi spesifik	Major khusus dan versi minor terbaru	Versi Terbaru
nodejs	18	`nodejs: 18`	`nodejs: 18.x`	`nodejs: latest`
	20	`nodejs: 20`	`nodejs: 20.x`	
	22	`nodejs: 22`	`nodejs: 22.x`	
	24	`nodejs: 24`	`nodejs: 24.x`	
php	8.1	`php: 8.1`	`php: 8.x`	`php: latest`
	8.2	`php: 8.2`		
	8.3	`php: 8.3`		
	8.4	`php: 8.4`		
	8.5	`php: 8.5`		
python	3.9	`python: 3.9`	`python: 3.x`	`python: latest`
	3.10	`python: 3.10`		
	3.11	`python: 3.11`		
	3.12	`python: 3.12`		
	3.13	`python: 3.13`		
	3.14	`python: 3.14`		
rubi	3.1	`ruby: 3.1`	`ruby: 3.x`	`ruby: latest`
	3.2	`ruby: 3.2`		
	3.3	`ruby: 3.3`		
	3.4	`ruby: 3.4`		

Nama runtime	Versi	Versi spesifik	Major khusus dan versi minor terbaru	Versi Terbaru
	4.0	`ruby: 4.0`	`ruby: 4.x`	
berkarat	1.94	`rust: 1.94`	`rust: 1.x`	`rust: latest`

Anda dapat menggunakan spesifikasi build untuk menginstal komponen lain (misalnya, Apache Maven AWS CLI, Apache Ant, Mocha, RSpec, atau sejenisnya) selama fase build. `install` Untuk informasi selengkapnya, lihat Contoh Buildspec.

Versi runtime khusus

Alih-alih menggunakan versi runtime pra-instal dalam CodeBuild-managed gambar, Anda dapat menginstal dan menggunakan versi khusus pilihan Anda. Tabel berikut mencantumkan runtime kustom yang tersedia dan cara menentukannya.

> **Note**
>
> Pemilihan versi runtime khusus hanya didukung untuk gambar Ubuntu dan Amazon Linux.

Versi runtime khusus

Nama runtime	Sintaksis	Contoh
dotnet	`<major>.<minor>.<patch>`	`5.0.408`
Cucking	`<major>.<minor>`	`1.19`
	`<major>.<minor>.<patch>`	`1.19.1`
java	`corretto<major>`	`corretto15`
nodejs	`<major>`	`14`
	`<major>.<minor>`	`14.21`
	`<major>.<minor>.<patch>`	`14.21.3`

Nama runtime	Sintaksis	Contoh
php	*<major>*.*<minor>*.*<patch>*	8.0.30
python	*<major>* *<major>*.*<minor>* *<major>*.*<minor>*.*<patch>*	3 3.7 3.7.16
rubi	*<major>*.*<minor>*.*<patch>*	3.0.6

Contoh buildspec runtime khusus

Berikut adalah contoh buildspec yang menentukan versi runtime kustom.

```
version: 0.2
phases:
  install:
    runtime-versions:
      java: corretto15
      php: 8.0.30
      ruby: 3.0.6
      golang: 1.19
      python: 3.7
      nodejs: 14
      dotnet: 5.0.408
```

Membangun mode dan tipe komputasi lingkungan

Di CodeBuild, Anda dapat menentukan image lingkungan komputasi dan runtime yang digunakan CodeBuild untuk menjalankan build Anda. Compute mengacu pada mesin komputasi (CPU, memori, dan sistem operasi) yang dikelola dan dikelola oleh CodeBuild. Gambar lingkungan runtime adalah gambar kontainer yang berjalan di atas platform komputasi pilihan Anda, dan menyertakan alat tambahan yang mungkin dibutuhkan build Anda, seperti. AWS CLI

Topik

- Tentang komputasi
- Tentang jenis lingkungan kapasitas yang dicadangkan

- Tentang jenis lingkungan sesuai permintaan

Tentang komputasi

CodeBuild menawarkan EC2 dan mode AWS Lambda komputasi. EC2 menawarkan fleksibilitas yang dioptimalkan selama pembuatan Anda dan AWS Lambda menawarkan kecepatan start-up yang dioptimalkan. AWS Lambda mendukung build yang lebih cepat karena latensi start-up yang lebih rendah. AWS Lambda juga secara otomatis menskalakan, jadi build tidak menunggu dalam antrian untuk dijalankan. Untuk informasi selengkapnya, lihat Jalankan dibangun di AWS Lambda komputasi.

Dengan mode komputasi EC2, Anda dapat menjalankan build dengan armada kapasitas sesuai permintaan atau cadangan. Untuk armada sesuai permintaan, Anda dapat memilih jenis komputasi yang telah ditentukan sebelumnya seperti atau. `BUILD_GENERAL1_SMALL` `BUILD_GENERAL1_LARGE` Untuk informasi selengkapnya, lihat Tentang jenis lingkungan sesuai permintaan. Untuk armada kapasitas cadangan, Anda dapat memilih konfigurasi komputasi termasuk vCPU, memori, dan ruang disk. Setelah menentukan konfigurasi, CodeBuild akan memilih jenis komputasi yang didukung yang sesuai dengan kebutuhan Anda. Untuk informasi selengkapnya, lihat Tentang jenis lingkungan kapasitas yang dicadangkan.

Tentang jenis lingkungan kapasitas yang dicadangkan

AWS CodeBuild menyediakan jenis lingkungan Linux x86, Arm, GPU, Windows dan macOS untuk armada kapasitas cadangan. Tabel berikut menunjukkan jenis mesin yang tersedia, memori, vCPU, dan ruang disk yang diurutkan berdasarkan wilayah:

US East (N. Virginia)

Tipe lingkungan	vCPUs	Memori	Ruang disk	Jenis mesin	Jenis instans komputasi
LENGAN	2	4 GiB	64GB	`GENERAL`	`reserved.arm.2cpu.4gib`
LENGAN	4	8 GiB	128GB	`GENERAL`	`reserved.arm.4cpu.8gib`

Tipe lingkungan	vCPUs	Memori	Ruang disk	Jenis mesin	Jenis instans komputasi
LENGAN	8	16 GiB	128GB	GENERAL	reserved.arm.8cpu.16gib
LENGAN	16	32 GiB	256GB	GENERAL	reserved.arm.16cpu.32gib
LENGAN	32	64 GiB	256GB	GENERAL	reserved.arm.32cpu.64gib
LENGAN	48	96 GiB	512GB	GENERAL	reserved.arm.48cpu.96gib
LENGAN	64	128 GiB	824GB	GENERAL	reserved.arm.64cpu.128gib
LENGAN EC2	2	4 GiB	64GB	GENERAL	reserved.arm.2cpu.4gib
LENGAN EC2	4	8 GiB	128GB	GENERAL	reserved.arm.4cpu.8gib
LENGAN EC2	8	16 GiB	128GB	GENERAL	reserved.arm.8cpu.16gib
Linux	2	4 GiB	64GB	GENERAL	reserved.x86-64.2cpu.4gib

Tipe lingkungan	vCPUs	Memori	Ruang disk	Jenis mesin	Jenis instans komputasi
Linux	4	8 GiB	128GB	GENERAL	reserved.x86-64.4cpu.8gib
Linux	8	16 GiB	128GB	GENERAL	reserved.x86-64.8cpu.16gib
Linux	16	32 GiB	256GB	GENERAL	reserved.x86-64.16cpu.32gib
Linux	36	72 GiB	256GB	GENERAL	reserved.x86-64.36cpu.72gib
Linux	48	96 GiB	512GB	GENERAL	reserved.x86-64.48cpu.96gib
Linux	72	144 GiB	824GB	GENERAL	reserved.x86-64.72cpu.144gib
Linux	96	192 GiB	824GB	GENERAL	reserved.x86-64.96cpu.192gib

Tipe lingkungan	vCPUs	Memori	Ruang disk	Jenis mesin	Jenis instans komputasi
Linux	48	96 GiB	824GB (SSD)	NVME	`reserved.x86-64.48cpu.96gib.nvme`
Linux	72	144 GiB	824GB (SSD)	NVME	`reserved.x86-64.72cpu.144gib.nvme`
Linux EC2	2	4 GiB	64GB	GENERAL	`reserved.x86-64.2cpu.4gib`
Linux EC2	4	8 GiB	128GB	GENERAL	`reserved.x86-64.4cpu.8gib`
Linux EC2	8	16 GiB	128GB	GENERAL	`reserved.x86-64.8cpu.16gib`
GPU Linux	4	16 GiB	235GB (SSD)	NVME	`reserved.gpu.4cpu.16gib.nvme`
GPU Linux	8	32 GiB	435GB (SSD)	NVME	`reserved.gpu.8cpu.32gib.nvme`

Tipe lingkungan	vCPUs	Memori	Ruang disk	Jenis mesin	Jenis instans komputasi
GPU Linux	16	64 GiB	585GB (SSD)	NVME	reserved.gpu.16cpu.64gib.nvme
GPU Linux	32	128 GiB	885GB (SSD)	NVME	reserved.gpu.32cpu.128gib.nvme
GPU Linux	48	192 GiB	3785GB (SSD)	NVME	reserved.gpu.48cpu.192gib.nvme
GPU Linux	64	256 GiB	1885GB (SSD)	NVME	reserved.gpu.64cpu.256gib.nvme
GPU Linux	96	384 GiB	3785GB (SSD)	NVME	reserved.gpu.96cpu.384gib.nvme
macOS	8	24 GiB	128GB	GENERAL	reserved.arm.m2.8cpu.24gib
macOS	12	32 GiB	256GB	GENERAL	reserved.arm.m2.12cpu.32gib

Tipe lingkungan	vCPUs	Memori	Ruang disk	Jenis mesin	Jenis instans komputasi
Windows	2	4 GiB	64GB	GENERAL	reserved.x86-64.2cpu.4gib
Windows	4	8 GiB	128GB	GENERAL	reserved.x86-64.4cpu.8gib
Windows	8	16 GiB	128GB	GENERAL	reserved.x86-64.8cpu.16gib
Windows	16	32 GiB	256GB	GENERAL	reserved.x86-64.16cpu.32gib
Windows	32	64 GiB	256GB	GENERAL	reserved.x86-64.32cpu.64gib
Windows	36	72 GiB	256GB	GENERAL	reserved.x86-64.36cpu.72gib
Windows	48	96 GiB	512GB	GENERAL	reserved.x86-64.48cpu.96gib

Tipe lingkungan	vCPUs	Memori	Ruang disk	Jenis mesin	Jenis instans komputasi
Windows	64	128 GiB	512GB	GENERAL	reserved.x86-64.64cpu.128gib
Windows	72	144 GiB	824GB	GENERAL	reserved.x86-64.72cpu.144gib
Windows	96	192 GiB	824GB	GENERAL	reserved.x86-64.96cpu.192gib
Windows EC2	4	8 GiB	128GB	GENERAL	reserved.x86-64.4cpu.8gib
Windows EC2	8	16 GiB	128GB	GENERAL	reserved.x86-64.8cpu.16gib

Untuk informasi selengkapnya tentang pengenal harga, lihat https://aws.amazon.com/codebuild/pricing/.

US East (Ohio)

Tipe lingkungan	vCPUs	Memori	Ruang disk	Jenis mesin	Jenis instans komputasi
LENGAN	2	4 GiB	64GB	GENERAL	reserved.arm.2cpu.4gib

Tipe lingkungan	vCPUs	Memori	Ruang disk	Jenis mesin	Jenis instans komputasi
LENGAN	4	8 GiB	128GB	GENERAL	reserved.arm.4cpu.8gib
LENGAN	8	16 GiB	128GB	GENERAL	reserved.arm.8cpu.16gib
LENGAN	16	32 GiB	256GB	GENERAL	reserved.arm.16cpu.32gib
LENGAN	32	64 GiB	256GB	GENERAL	reserved.arm.32cpu.64gib
LENGAN	48	96 GiB	512GB	GENERAL	reserved.arm.48cpu.96gib
LENGAN	64	128 GiB	824GB	GENERAL	reserved.arm.64cpu.128gib
LENGAN EC2	2	4 GiB	64GB	GENERAL	reserved.arm.2cpu.4gib
LENGAN EC2	4	8 GiB	128GB	GENERAL	reserved.arm.4cpu.8gib
LENGAN EC2	8	16 GiB	128GB	GENERAL	reserved.arm.8cpu.16gib

Tipe lingkungan	vCPUs	Memori	Ruang disk	Jenis mesin	Jenis instans komputasi
Linux	2	4 GiB	64GB	GENERAL	reserved.x86-64.2cpu.4gib
Linux	4	8 GiB	128GB	GENERAL	reserved.x86-64.4cpu.8gib
Linux	8	16 GiB	128GB	GENERAL	reserved.x86-64.8cpu.16gib
Linux	16	32 GiB	256GB	GENERAL	reserved.x86-64.16cpu.32gib
Linux	36	72 GiB	256GB	GENERAL	reserved.x86-64.36cpu.72gib
Linux	48	96 GiB	512GB	GENERAL	reserved.x86-64.48cpu.96gib
Linux	96	192 GiB	824GB	GENERAL	reserved.x86-64.96cpu.192gib

Tipe lingkungan	vCPUs	Memori	Ruang disk	Jenis mesin	Jenis instans komputasi
Linux	48	96 GiB	824GB (SSD)	NVME	reserved.x86-64.48cpu.96gib.nvme
Linux	72	144 GiB	824GB (SSD)	NVME	reserved.x86-64.72cpu.144gib.nvme
Linux EC2	2	4 GiB	64GB	GENERAL	reserved.x86-64.2cpu.4gib
Linux EC2	4	8 GiB	128GB	GENERAL	reserved.x86-64.4cpu.8gib
Linux EC2	8	16 GiB	128GB	GENERAL	reserved.x86-64.8cpu.16gib
GPU Linux	4	16 GiB	235GB (SSD)	NVME	reserved.gpu.4cpu.16gib.nvme
GPU Linux	8	32 GiB	435GB (SSD)	NVME	reserved.gpu.8cpu.32gib.nvme

Tipe lingkungan	vCPUs	Memori	Ruang disk	Jenis mesin	Jenis instans komputasi
GPU Linux	16	64 GiB	585GB (SSD)	NVME	reserved.gpu.16cpu.64gib.nvme
GPU Linux	32	128 GiB	885GB (SSD)	NVME	reserved.gpu.32cpu.128gib.nvme
GPU Linux	48	192 GiB	3785GB (SSD)	NVME	reserved.gpu.48cpu.192gib.nvme
macOS	8	24 GiB	128GB	GENERAL	reserved.arm.m2.8cpu.24gib
macOS	12	32 GiB	256GB	GENERAL	reserved.arm.m2.12cpu.32gib
Windows	2	4 GiB	64GB	GENERAL	reserved.x86-64.2cpu.4gib
Windows	4	8 GiB	128GB	GENERAL	reserved.x86-64.4cpu.8gib
Windows	8	16 GiB	128GB	GENERAL	reserved.x86-64.8cpu.16gib

Tipe lingkungan	vCPUs	Memori	Ruang disk	Jenis mesin	Jenis instans komputasi
Windows	16	32 GiB	256GB	GENERAL	reserved.x86-64.16cpu.32gib
Windows	32	64 GiB	256GB	GENERAL	reserved.x86-64.32cpu.64gib
Windows	36	72 GiB	256GB	GENERAL	reserved.x86-64.36cpu.72gib
Windows	48	96 GiB	512GB	GENERAL	reserved.x86-64.48cpu.96gib
Windows	64	128 GiB	512GB	GENERAL	reserved.x86-64.64cpu.128gib
Windows	96	192 GiB	824GB	GENERAL	reserved.x86-64.96cpu.192gib
Windows EC2	4	8 GiB	128GB	GENERAL	reserved.x86-64.4cpu.8gib

Tipe lingkungan	vCPUs	Memori	Ruang disk	Jenis mesin	Jenis instans komputasi
Windows EC2	8	16 GiB	128GB	GENERAL	reserved.x86-64.8cpu.16gib

Untuk informasi selengkapnya tentang pengenal harga, lihat https://aws.amazon.com/codebuild/pricing/.

US West (N. California)

Tipe lingkungan	vCPUs	Memori	Ruang disk	Jenis mesin	Jenis instans komputasi
LENGAN	2	4 GiB	64GB	GENERAL	reserved.arm.2cpu.4gib
LENGAN	4	8 GiB	128GB	GENERAL	reserved.arm.4cpu.8gib
LENGAN	8	16 GiB	128GB	GENERAL	reserved.arm.8cpu.16gib
LENGAN	16	32 GiB	256GB	GENERAL	reserved.arm.16cpu.32gib
LENGAN	32	64 GiB	256GB	GENERAL	reserved.arm.32cpu.64gib
LENGAN	48	96 GiB	824GB	GENERAL	reserved.arm.48cpu.96gib

Tipe lingkungan	vCPUs	Memori	Ruang disk	Jenis mesin	Jenis instans komputasi
LENGAN	64	128 GiB	824GB	GENERAL	reserved.arm.64cpu.128gib
LENGAN EC2	2	4 GiB	64GB	GENERAL	reserved.arm.2cpu.4gib
LENGAN EC2	4	8 GiB	128GB	GENERAL	reserved.arm.4cpu.8gib
LENGAN EC2	8	16 GiB	128GB	GENERAL	reserved.arm.8cpu.16gib
Linux	2	4 GiB	64GB	GENERAL	reserved.x86-64.2cpu.4gib
Linux	4	8 GiB	128GB	GENERAL	reserved.x86-64.4cpu.8gib
Linux	8	16 GiB	128GB	GENERAL	reserved.x86-64.8cpu.16gib
Linux	16	32 GiB	256GB	GENERAL	reserved.x86-64.16cpu.32gib

Tipe lingkungan	vCPUs	Memori	Ruang disk	Jenis mesin	Jenis instans komputasi
Linux	32	64 GiB	256GB	GENERAL	reserved.x86-64.32cpu.64gib
Linux	36	72 GiB	256GB	GENERAL	reserved.x86-64.36cpu.72gib
Linux	48	96 GiB	512GB	GENERAL	reserved.x86-64.48cpu.96gib
Linux	64	128 GiB	512GB	GENERAL	reserved.x86-64.64cpu.128gib
Linux	72	144 GiB	824GB	GENERAL	reserved.x86-64.72cpu.144gib
Linux	96	192 GiB	824GB	GENERAL	reserved.x86-64.96cpu.192gib
Linux	72	144 GiB	824GB (SSD)	NVME	reserved.x86-64.72cpu.144gib.nvme

Tipe lingkungan	vCPUs	Memori	Ruang disk	Jenis mesin	Jenis instans komputasi
Linux EC2	2	4 GiB	64GB	GENERAL	reserved.x86-64.2cpu.4gib
Linux EC2	4	8 GiB	128GB	GENERAL	reserved.x86-64.4cpu.8gib
Linux EC2	8	16 GiB	128GB	GENERAL	reserved.x86-64.8cpu.16gib
Windows	2	4 GiB	64GB	GENERAL	reserved.x86-64.2cpu.4gib
Windows	4	8 GiB	128GB	GENERAL	reserved.x86-64.4cpu.8gib
Windows	8	16 GiB	128GB	GENERAL	reserved.x86-64.8cpu.16gib
Windows	16	32 GiB	256GB	GENERAL	reserved.x86-64.16cpu.32gib
Windows	32	64 GiB	256GB	GENERAL	reserved.x86-64.32cpu.64gib

Tipe lingkungan	vCPUs	Memori	Ruang disk	Jenis mesin	Jenis instans komputasi
Windows	36	72 GiB	256GB	GENERAL	reserved.x86-64.36cpu.72gib
Windows	48	96 GiB	512GB	GENERAL	reserved.x86-64.48cpu.96gib
Windows	64	128 GiB	512GB	GENERAL	reserved.x86-64.64cpu.128gib
Windows	72	144 GiB	824GB	GENERAL	reserved.x86-64.72cpu.144gib
Windows	96	192 GiB	824GB	GENERAL	reserved.x86-64.96cpu.192gib
Windows EC2	4	8 GiB	128GB	GENERAL	reserved.x86-64.4cpu.8gib
Windows EC2	8	16 GiB	128GB	GENERAL	reserved.x86-64.8cpu.16gib

Untuk informasi selengkapnya tentang pengenal harga, lihat https://aws.amazon.com/codebuild/pricing/.

US West (Oregon)

Tipe lingkungan	vCPUs	Memori	Ruang disk	Jenis mesin	Jenis instans komputasi
LENGAN	2	4 GiB	64GB	GENERAL	reserved.arm.2cpu.4gib
LENGAN	4	8 GiB	128GB	GENERAL	reserved.arm.4cpu.8gib
LENGAN	8	16 GiB	128GB	GENERAL	reserved.arm.8cpu.16gib
LENGAN	16	32 GiB	256GB	GENERAL	reserved.arm.16cpu.32gib
LENGAN	32	64 GiB	256GB	GENERAL	reserved.arm.32cpu.64gib
LENGAN	48	96 GiB	512GB	GENERAL	reserved.arm.48cpu.96gib
LENGAN	64	128 GiB	824GB	GENERAL	reserved.arm.64cpu.128gib
LENGAN EC2	2	4 GiB	64GB	GENERAL	reserved.arm.2cpu.4gib

Tipe lingkungan	vCPUs	Memori	Ruang disk	Jenis mesin	Jenis instans komputasi
LENGAN EC2	4	8 GiB	128GB	GENERAL	reserved.arm.4cpu.8gib
LENGAN EC2	8	16 GiB	128GB	GENERAL	reserved.arm.8cpu.16gib
Linux	2	4 GiB	64GB	GENERAL	reserved.x86-64.2cpu.4gib
Linux	4	8 GiB	128GB	GENERAL	reserved.x86-64.4cpu.8gib
Linux	8	16 GiB	128GB	GENERAL	reserved.x86-64.8cpu.16gib
Linux	16	32 GiB	256GB	GENERAL	reserved.x86-64.16cpu.32gib
Linux	36	72 GiB	256GB	GENERAL	reserved.x86-64.36cpu.72gib
Linux	48	96 GiB	512GB	GENERAL	reserved.x86-64.48cpu.96gib

Tipe lingkungan	vCPUs	Memori	Ruang disk	Jenis mesin	Jenis instans komputasi
Linux	72	144 GiB	824GB	GENERAL	reserved.x86-64.72cpu.144gib
Linux	96	192 GiB	824GB	GENERAL	reserved.x86-64.96cpu.192gib
Linux	48	96 GiB	824GB (SSD)	NVME	reserved.x86-64.48cpu.96gib.nvme
Linux	72	144 GiB	824GB (SSD)	NVME	reserved.x86-64.72cpu.144gib.nvme
Linux EC2	2	4 GiB	64GB	GENERAL	reserved.x86-64.2cpu.4gib
Linux EC2	4	8 GiB	128GB	GENERAL	reserved.x86-64.4cpu.8gib
Linux EC2	8	16 GiB	128GB	GENERAL	reserved.x86-64.8cpu.16gib

Tipe lingkungan	vCPUs	Memori	Ruang disk	Jenis mesin	Jenis instans komputasi
GPU Linux	4	16 GiB	235GB (SSD)	NVME	reserved.gpu.4cpu.16gib.nvme
GPU Linux	8	32 GiB	435GB (SSD)	NVME	reserved.gpu.8cpu.32gib.nvme
GPU Linux	16	64 GiB	585GB (SSD)	NVME	reserved.gpu.16cpu.64gib.nvme
GPU Linux	32	128 GiB	885GB (SSD)	NVME	reserved.gpu.32cpu.128gib.nvme
GPU Linux	48	192 GiB	3785GB (SSD)	NVME	reserved.gpu.48cpu.192gib.nvme
GPU Linux	64	256 GiB	1885GB (SSD)	NVME	reserved.gpu.64cpu.256gib.nvme
macOS	8	24 GiB	128GB	GENERAL	reserved.arm.m2.8cpu.24gib

Tipe lingkungan	vCPUs	Memori	Ruang disk	Jenis mesin	Jenis instans komputasi
macOS	12	32 GiB	256GB	`GENERAL`	`reserved.arm.m2.12cpu.32gib`
Windows	2	4 GiB	64GB	`GENERAL`	`reserved.x86-64.2cpu.4gib`
Windows	4	8 GiB	128GB	`GENERAL`	`reserved.x86-64.4cpu.8gib`
Windows	8	16 GiB	128GB	`GENERAL`	`reserved.x86-64.8cpu.16gib`
Windows	16	32 GiB	256GB	`GENERAL`	`reserved.x86-64.16cpu.32gib`
Windows	32	64 GiB	256GB	`GENERAL`	`reserved.x86-64.32cpu.64gib`
Windows	36	72 GiB	256GB	`GENERAL`	`reserved.x86-64.36cpu.72gib`

Tipe lingkungan	vCPUs	Memori	Ruang disk	Jenis mesin	Jenis instans komputasi
Windows	48	96 GiB	512GB	GENERAL	reserved.x86-64.48cpu.96gib
Windows	64	128 GiB	512GB	GENERAL	reserved.x86-64.64cpu.128gib
Windows	72	144 GiB	824GB	GENERAL	reserved.x86-64.72cpu.144gib
Windows	96	192 GiB	824GB	GENERAL	reserved.x86-64.96cpu.192gib
Windows EC2	4	8 GiB	128GB	GENERAL	reserved.x86-64.4cpu.8gib
Windows EC2	8	16 GiB	128GB	GENERAL	reserved.x86-64.8cpu.16gib

Untuk informasi selengkapnya tentang pengenal harga, lihat https://aws.amazon.com/codebuild/pricing/.

Asia Pacific (Tokyo)

Tipe lingkungan	vCPUs	Memori	Ruang disk	Jenis mesin	Jenis instans komputasi
LENGAN	2	4 GiB	64GB	GENERAL	reserved.arm.2cpu.4gib
LENGAN	4	8 GiB	128GB	GENERAL	reserved.arm.4cpu.8gib
LENGAN	8	16 GiB	128GB	GENERAL	reserved.arm.8cpu.16gib
LENGAN	16	32 GiB	256GB	GENERAL	reserved.arm.16cpu.32gib
LENGAN	32	64 GiB	256GB	GENERAL	reserved.arm.32cpu.64gib
LENGAN	48	96 GiB	512GB	GENERAL	reserved.arm.48cpu.96gib
LENGAN	64	128 GiB	824GB	GENERAL	reserved.arm.64cpu.128gib
LENGAN EC2	2	4 GiB	64GB	GENERAL	reserved.arm.2cpu.4gib

Tipe lingkungan	vCPUs	Memori	Ruang disk	Jenis mesin	Jenis instans komputasi
LENGAN EC2	4	8 GiB	128GB	GENERAL	reserved.arm.4cpu.8gib
LENGAN EC2	8	16 GiB	128GB	GENERAL	reserved.arm.8cpu.16gib
Linux	2	4 GiB	64GB	GENERAL	reserved.x86-64.2cpu.4gib
Linux	4	8 GiB	128GB	GENERAL	reserved.x86-64.4cpu.8gib
Linux	8	16 GiB	128GB	GENERAL	reserved.x86-64.8cpu.16gib
Linux	16	32 GiB	256GB	GENERAL	reserved.x86-64.16cpu.32gib
Linux	36	72 GiB	256GB	GENERAL	reserved.x86-64.36cpu.72gib
Linux	48	96 GiB	512GB	GENERAL	reserved.x86-64.48cpu.96gib

Tipe lingkungan	vCPUs	Memori	Ruang disk	Jenis mesin	Jenis instans komputasi
Linux	72	144 GiB	824GB	GENERAL	reserved.x86-64.72cpu.144gib
Linux	96	192 GiB	824GB	GENERAL	reserved.x86-64.96cpu.192gib
Linux	48	96 GiB	824GB (SSD)	NVME	reserved.x86-64.48cpu.96gib.nvme
Linux	72	144 GiB	824GB (SSD)	NVME	reserved.x86-64.72cpu.144gib.nvme
Linux EC2	2	4 GiB	64GB	GENERAL	reserved.x86-64.2cpu.4gib
Linux EC2	4	8 GiB	128GB	GENERAL	reserved.x86-64.4cpu.8gib
Linux EC2	8	16 GiB	128GB	GENERAL	reserved.x86-64.8cpu.16gib

Tipe lingkungan	vCPUs	Memori	Ruang disk	Jenis mesin	Jenis instans komputasi
GPU Linux	4	16 GiB	235GB (SSD)	NVME	reserved.gpu.4cpu.16gib.nvme
GPU Linux	8	32 GiB	435GB (SSD)	NVME	reserved.gpu.8cpu.32gib.nvme
GPU Linux	48	192 GiB	3785GB (SSD)	NVME	reserved.gpu.48cpu.192gib.nvme
Windows	2	4 GiB	64GB	GENERAL	reserved.x86-64.2cpu.4gib
Windows	4	8 GiB	128GB	GENERAL	reserved.x86-64.4cpu.8gib
Windows	8	16 GiB	128GB	GENERAL	reserved.x86-64.8cpu.16gib
Windows	16	32 GiB	256GB	GENERAL	reserved.x86-64.16cpu.32gib

Tipe lingkungan	vCPUs	Memori	Ruang disk	Jenis mesin	Jenis instans komputasi
Windows	32	64 GiB	256GB	GENERAL	reserved.x86-64.32cpu.64gib
Windows	36	72 GiB	256GB	GENERAL	reserved.x86-64.36cpu.72gib
Windows	48	96 GiB	512GB	GENERAL	reserved.x86-64.48cpu.96gib
Windows	64	128 GiB	512GB	GENERAL	reserved.x86-64.64cpu.128gib
Windows	72	144 GiB	824GB	GENERAL	reserved.x86-64.72cpu.144gib
Windows	96	192 GiB	824GB	GENERAL	reserved.x86-64.96cpu.192gib
Windows EC2	4	8 GiB	128GB	GENERAL	reserved.x86-64.4cpu.8gib

Tipe lingkungan	vCPUs	Memori	Ruang disk	Jenis mesin	Jenis instans komputasi
Windows EC2	8	16 GiB	128GB	GENERAL	reserved.x86-64.8cpu.16gib

Untuk informasi selengkapnya tentang pengenal harga, lihat https://aws.amazon.com/codebuild/pricing/.

Asia Pacific (Malaysia)

Tipe lingkungan	vCPUs	Memori	Jenis mesin	Jenis instans komputasi
LENGAN	2	4 GiB	GENERAL	reserved:arm.g1.small
LENGAN	4	8 GiB	GENERAL	reserved:arm.g1.medium
LENGAN	8	16 GiB	GENERAL	reserved:arm.g1.large
LENGAN	32	64 GiB	GENERAL	reserved:arm.g1.xlarge
LENGAN	48	96 GiB	GENERAL	reserved:arm.g1.2xlarge
Linux	2	4 GiB	GENERAL	reserved:x86-64.g1.small

Tipe lingkungan	vCPUs	Memori	Jenis mesin	Jenis instans komputasi
Linux	4	8 GiB	GENERAL	reserved: x86-64.g1 .medium
Linux	8	16 GiB	GENERAL	reserved: x86-64.g1 .large

Untuk informasi selengkapnya tentang pengenal harga, lihat https://aws.amazon.com/codebuild/pricing/.

Asia Pacific (New Zealand)

Tipe lingkungan	vCPUs	Memori	Jenis mesin	Jenis instans komputasi
LENGAN	2	4 GiB	GENERAL	reserved: arm.g1.sm all
LENGAN	4	8 GiB	GENERAL	reserved: arm.g1.me dium
LENGAN	8	16 GiB	GENERAL	reserved: arm.g1.la rge
LENGAN	32	64 GiB	GENERAL	reserved: arm.g1.xl arge
LENGAN	48	96 GiB	GENERAL	reserved: arm.g1.2x large

Tipe lingkungan	vCPUs	Memori	Jenis mesin	Jenis instans komputasi
Linux	2	4 GiB	GENERAL	reserved: x86-64.g1 .small
Linux	4	8 GiB	GENERAL	reserved: x86-64.g1 .medium
Linux	8	16 GiB	GENERAL	reserved: x86-64.g1 .large

Untuk informasi selengkapnya tentang pengenal harga, lihat https://aws.amazon.com/codebuild/pricing/.

Asia Pacific (Thailand)

Tipe lingkungan	vCPUs	Memori	Jenis mesin	Jenis instans komputasi
LENGAN	2	4 GiB	GENERAL	reserved: arm.g1.sm all
LENGAN	4	8 GiB	GENERAL	reserved: arm.g1.me dium
LENGAN	8	16 GiB	GENERAL	reserved: arm.g1.la rge
LENGAN	32	64 GiB	GENERAL	reserved: arm.g1.xl arge

Tipe lingkungan	vCPUs	Memori	Jenis mesin	Jenis instans komputasi
LENGAN	48	96 GiB	`GENERAL`	`reserved: arm.g1.2x large`
Linux	2	4 GiB	`GENERAL`	`reserved: x86-64.g1 .small`
Linux	4	8 GiB	`GENERAL`	`reserved: x86-64.g1 .medium`
Linux	8	16 GiB	`GENERAL`	`reserved: x86-64.g1 .large`

Untuk informasi selengkapnya tentang pengenal harga, lihat https://aws.amazon.com/codebuild/pricing/.

Asia Pacific (Mumbai)

Tipe lingkungan	vCPUs	Memori	Ruang disk	Jenis mesin	Jenis instans komputasi
LENGAN	2	4 GiB	64GB	`GENERAL`	`reserved. arm.2cpu. 4gib`
LENGAN	4	8 GiB	128GB	`GENERAL`	`reserved. arm.4cpu. 8gib`
LENGAN	8	16 GiB	128GB	`GENERAL`	`reserved. arm.8cpu. 16gib`

Tipe lingkungan	vCPUs	Memori	Ruang disk	Jenis mesin	Jenis instans komputasi
LENGAN	16	32 GiB	256GB	GENERAL	reserved.arm.16cpu.32gib
LENGAN	32	64 GiB	256GB	GENERAL	reserved.arm.32cpu.64gib
LENGAN	48	96 GiB	512GB	GENERAL	reserved.arm.48cpu.96gib
LENGAN	64	128 GiB	824GB	GENERAL	reserved.arm.64cpu.128gib
LENGAN EC2	2	4 GiB	64GB	GENERAL	reserved.arm.2cpu.4gib
LENGAN EC2	4	8 GiB	128GB	GENERAL	reserved.arm.4cpu.8gib
LENGAN EC2	8	16 GiB	128GB	GENERAL	reserved.arm.8cpu.16gib
Linux	2	4 GiB	64GB	GENERAL	reserved.x86-64.2cpu.4gib
Linux	4	8 GiB	128GB	GENERAL	reserved.x86-64.4cpu.8gib

Tipe lingkungan	vCPUs	Memori	Ruang disk	Jenis mesin	Jenis instans komputasi
Linux	8	16 GiB	128GB	GENERAL	reserved.x86-64.8cpu.16gib
Linux	16	32 GiB	256GB	GENERAL	reserved.x86-64.16cpu.32gib
Linux	36	72 GiB	256GB	GENERAL	reserved.x86-64.36cpu.72gib
Linux	48	96 GiB	512GB	GENERAL	reserved.x86-64.48cpu.96gib
Linux	72	144 GiB	824GB	GENERAL	reserved.x86-64.72cpu.144gib
Linux	96	192 GiB	824GB	GENERAL	reserved.x86-64.96cpu.192gib
Linux	72	144 GiB	824GB (SSD)	NVME	reserved.x86-64.72cpu.144gib.nvme

Tipe lingkungan	vCPUs	Memori	Ruang disk	Jenis mesin	Jenis instans komputasi
Linux EC2	2	4 GiB	64GB	GENERAL	reserved.x86-64.2cpu.4gib
Linux EC2	4	8 GiB	128GB	GENERAL	reserved.x86-64.4cpu.8gib
Linux EC2	8	16 GiB	128GB	GENERAL	reserved.x86-64.8cpu.16gib
GPU Linux	4	16 GiB	235GB (SSD)	NVME	reserved.gpu.4cpu.16gib.nvme
GPU Linux	8	32 GiB	435GB (SSD)	NVME	reserved.gpu.8cpu.32gib.nvme
GPU Linux	16	64 GiB	585GB (SSD)	NVME	reserved.gpu.16cpu.64gib.nvme
Windows	2	4 GiB	64GB	GENERAL	reserved.x86-64.2cpu.4gib
Windows	4	8 GiB	128GB	GENERAL	reserved.x86-64.4cpu.8gib

Tipe lingkungan	vCPUs	Memori	Ruang disk	Jenis mesin	Jenis instans komputasi
Windows	8	16 GiB	128GB	GENERAL	reserved.x86-64.8cpu.16gib
Windows	16	32 GiB	256GB	GENERAL	reserved.x86-64.16cpu.32gib
Windows	32	64 GiB	256GB	GENERAL	reserved.x86-64.32cpu.64gib
Windows	36	72 GiB	256GB	GENERAL	reserved.x86-64.36cpu.72gib
Windows	48	96 GiB	512GB	GENERAL	reserved.x86-64.48cpu.96gib
Windows	64	128 GiB	512GB	GENERAL	reserved.x86-64.64cpu.128gib
Windows	72	144 GiB	824GB	GENERAL	reserved.x86-64.72cpu.144gib

Tipe lingkungan	vCPUs	Memori	Ruang disk	Jenis mesin	Jenis instans komputasi
Windows	96	192 GiB	824GB	GENERAL	reserved.x86-64.96cpu.192gib
Windows EC2	4	8 GiB	128GB	GENERAL	reserved.x86-64.4cpu.8gib
Windows EC2	8	16 GiB	128GB	GENERAL	reserved.x86-64.8cpu.16gib

Untuk informasi selengkapnya tentang pengenal harga, lihat https://aws.amazon.com/codebuild/pricing/.

Asia Pacific (Singapore)

Tipe lingkungan	vCPUs	Memori	Ruang disk	Jenis mesin	Jenis instans komputasi
LENGAN	2	4 GiB	64GB	GENERAL	reserved.arm.2cpu.4gib
LENGAN	4	8 GiB	128GB	GENERAL	reserved.arm.4cpu.8gib
LENGAN	8	16 GiB	128GB	GENERAL	reserved.arm.8cpu.16gib

Tipe lingkungan	vCPUs	Memori	Ruang disk	Jenis mesin	Jenis instans komputasi
LENGAN	16	32 GiB	256GB	GENERAL	reserved.arm.16cpu.32gib
LENGAN	32	64 GiB	256GB	GENERAL	reserved.arm.32cpu.64gib
LENGAN	48	96 GiB	512GB	GENERAL	reserved.arm.48cpu.96gib
LENGAN	64	128 GiB	824GB	GENERAL	reserved.arm.64cpu.128gib
LENGAN EC2	2	4 GiB	64GB	GENERAL	reserved.arm.2cpu.4gib
LENGAN EC2	4	8 GiB	128GB	GENERAL	reserved.arm.4cpu.8gib
LENGAN EC2	8	16 GiB	128GB	GENERAL	reserved.arm.8cpu.16gib
Linux	2	4 GiB	64GB	GENERAL	reserved.x86-64.2cpu.4gib
Linux	4	8 GiB	128GB	GENERAL	reserved.x86-64.4cpu.8gib

Tipe lingkungan	vCPUs	Memori	Ruang disk	Jenis mesin	Jenis instans komputasi
Linux	8	16 GiB	128GB	GENERAL	reserved.x86-64.8cpu.16gib
Linux	16	32 GiB	256GB	GENERAL	reserved.x86-64.16cpu.32gib
Linux	36	72 GiB	256GB	GENERAL	reserved.x86-64.36cpu.72gib
Linux	48	96 GiB	512GB	GENERAL	reserved.x86-64.48cpu.96gib
Linux	72	144 GiB	824GB	GENERAL	reserved.x86-64.72cpu.144gib
Linux	96	192 GiB	824GB	GENERAL	reserved.x86-64.96cpu.192gib
Linux	48	96 GiB	824GB (SSD)	NVME	reserved.x86-64.48cpu.96gib.nvme

Tipe lingkungan	vCPUs	Memori	Ruang disk	Jenis mesin	Jenis instans komputasi
Linux	72	144 GiB	824GB (SSD)	NVME	`reserved.x86-64.72cpu.144gib.nvme`
Linux EC2	2	4 GiB	64GB	GENERAL	`reserved.x86-64.2cpu.4gib`
Linux EC2	4	8 GiB	128GB	GENERAL	`reserved.x86-64.4cpu.8gib`
Linux EC2	8	16 GiB	128GB	GENERAL	`reserved.x86-64.8cpu.16gib`
Windows	2	4 GiB	64GB	GENERAL	`reserved.x86-64.2cpu.4gib`
Windows	4	8 GiB	128GB	GENERAL	`reserved.x86-64.4cpu.8gib`
Windows	8	16 GiB	128GB	GENERAL	`reserved.x86-64.8cpu.16gib`
Windows	16	32 GiB	256GB	GENERAL	`reserved.x86-64.16cpu.32gib`

Tipe lingkungan	vCPUs	Memori	Ruang disk	Jenis mesin	Jenis instans komputasi
Windows	32	64 GiB	256GB	GENERAL	reserved.x86-64.32cpu.64gib
Windows	36	72 GiB	256GB	GENERAL	reserved.x86-64.36cpu.72gib
Windows	48	96 GiB	512GB	GENERAL	reserved.x86-64.48cpu.96gib
Windows	64	128 GiB	512GB	GENERAL	reserved.x86-64.64cpu.128gib
Windows	72	144 GiB	824GB	GENERAL	reserved.x86-64.72cpu.144gib
Windows	96	192 GiB	824GB	GENERAL	reserved.x86-64.96cpu.192gib
Windows EC2	4	8 GiB	128GB	GENERAL	reserved.x86-64.4cpu.8gib

Tipe lingkungan	vCPUs	Memori	Ruang disk	Jenis mesin	Jenis instans komputasi
Windows EC2	8	16 GiB	128GB	GENERAL	reserved.x86-64.8cpu.16gib

Untuk informasi selengkapnya tentang pengenal harga, lihat https://aws.amazon.com/codebuild/pricing/.

Asia Pacific (Sydney)

Tipe lingkungan	vCPUs	Memori	Ruang disk	Jenis mesin	Jenis instans komputasi
LENGAN	2	4 GiB	64GB	GENERAL	reserved.arm.2cpu.4gib
LENGAN	4	8 GiB	128GB	GENERAL	reserved.arm.4cpu.8gib
LENGAN	8	16 GiB	128GB	GENERAL	reserved.arm.8cpu.16gib
LENGAN	16	32 GiB	256GB	GENERAL	reserved.arm.16cpu.32gib
LENGAN	32	64 GiB	256GB	GENERAL	reserved.arm.32cpu.64gib
LENGAN	48	96 GiB	512GB	GENERAL	reserved.arm.48cpu.96gib

Tipe lingkungan	vCPUs	Memori	Ruang disk	Jenis mesin	Jenis instans komputasi
LENGAN	64	128 GiB	824GB	GENERAL	reserved.arm.64cpu.128gib
LENGAN EC2	2	4 GiB	64GB	GENERAL	reserved.arm.2cpu.4gib
LENGAN EC2	4	8 GiB	128GB	GENERAL	reserved.arm.4cpu.8gib
LENGAN EC2	8	16 GiB	128GB	GENERAL	reserved.arm.8cpu.16gib
Linux	2	4 GiB	64GB	GENERAL	reserved.x86-64.2cpu.4gib
Linux	4	8 GiB	128GB	GENERAL	reserved.x86-64.4cpu.8gib
Linux	8	16 GiB	128GB	GENERAL	reserved.x86-64.8cpu.16gib
Linux	16	32 GiB	256GB	GENERAL	reserved.x86-64.16cpu.32gib

Tipe lingkungan	vCPUs	Memori	Ruang disk	Jenis mesin	Jenis instans komputasi
Linux	36	72 GiB	256GB	GENERAL	reserved.x86-64.36cpu.72gib
Linux	48	96 GiB	512GB	GENERAL	reserved.x86-64.48cpu.96gib
Linux	72	144 GiB	824GB	GENERAL	reserved.x86-64.72cpu.144gib
Linux	96	192 GiB	824GB	GENERAL	reserved.x86-64.96cpu.192gib
Linux	48	96 GiB	824GB (SSD)	NVME	reserved.x86-64.48cpu.96gib.nvme
Linux	72	144 GiB	824GB (SSD)	NVME	reserved.x86-64.72cpu.144gib.nvme
Linux EC2	2	4 GiB	64GB	GENERAL	reserved.x86-64.2cpu.4gib

Tipe lingkungan	vCPUs	Memori	Ruang disk	Jenis mesin	Jenis instans komputasi
Linux EC2	4	8 GiB	128GB	`GENERAL`	`reserved.x86-64.4cpu.8gib`
Linux EC2	8	16 GiB	128GB	`GENERAL`	`reserved.x86-64.8cpu.16gib`
GPU Linux	4	16 GiB	235GB (SSD)	`NVME`	`reserved.gpu.4cpu.16gib.nvme`
GPU Linux	8	32 GiB	435GB (SSD)	`NVME`	`reserved.gpu.8cpu.32gib.nvme`
GPU Linux	16	64 GiB	585GB (SSD)	`NVME`	`reserved.gpu.16cpu.64gib.nvme`
GPU Linux	48	192 GiB	3785GB (SSD)	`NVME`	`reserved.gpu.48cpu.192gib.nvme`
macOS	8	24 GiB	128GB	`GENERAL`	`reserved.arm.m2.8cpu.24gib`

Tipe lingkungan	vCPUs	Memori	Ruang disk	Jenis mesin	Jenis instans komputasi
macOS	12	32 GiB	256GB	GENERAL	reserved.arm.m2.12cpu.32gib
Windows	2	4 GiB	64GB	GENERAL	reserved.x86-64.2cpu.4gib
Windows	4	8 GiB	128GB	GENERAL	reserved.x86-64.4cpu.8gib
Windows	8	16 GiB	128GB	GENERAL	reserved.x86-64.8cpu.16gib
Windows	16	32 GiB	256GB	GENERAL	reserved.x86-64.16cpu.32gib
Windows	32	64 GiB	256GB	GENERAL	reserved.x86-64.32cpu.64gib
Windows	36	72 GiB	256GB	GENERAL	reserved.x86-64.36cpu.72gib

Tipe lingkungan	vCPUs	Memori	Ruang disk	Jenis mesin	Jenis instans komputasi
Windows	48	96 GiB	512GB	GENERAL	reserved.x86-64.48cpu.96gib
Windows	64	128 GiB	512GB	GENERAL	reserved.x86-64.64cpu.128gib
Windows	72	144 GiB	824GB	GENERAL	reserved.x86-64.72cpu.144gib
Windows	96	192 GiB	824GB	GENERAL	reserved.x86-64.96cpu.192gib
Windows EC2	4	8 GiB	128GB	GENERAL	reserved.x86-64.4cpu.8gib
Windows EC2	8	16 GiB	128GB	GENERAL	reserved.x86-64.8cpu.16gib

Untuk informasi selengkapnya tentang pengenal harga, lihat https://aws.amazon.com/codebuild/pricing/.

Europe (Frankfurt)

Tipe lingkungan	vCPUs	Memori	Ruang disk	Jenis mesin	Jenis instans komputasi
LENGAN	2	4 GiB	64GB	GENERAL	reserved.arm.2cpu.4gib
LENGAN	4	8 GiB	128GB	GENERAL	reserved.arm.4cpu.8gib
LENGAN	8	16 GiB	128GB	GENERAL	reserved.arm.8cpu.16gib
LENGAN	16	32 GiB	256GB	GENERAL	reserved.arm.16cpu.32gib
LENGAN	32	64 GiB	256GB	GENERAL	reserved.arm.32cpu.64gib
LENGAN	48	96 GiB	512GB	GENERAL	reserved.arm.48cpu.96gib
LENGAN	64	128 GiB	824GB	GENERAL	reserved.arm.64cpu.128gib
LENGAN EC2	2	4 GiB	64GB	GENERAL	reserved.arm.2cpu.4gib

Tipe lingkungan	vCPUs	Memori	Ruang disk	Jenis mesin	Jenis instans komputasi
LENGAN EC2	4	8 GiB	128GB	GENERAL	reserved.arm.4cpu.8gib
LENGAN EC2	8	16 GiB	128GB	GENERAL	reserved.arm.8cpu.16gib
Linux	2	4 GiB	64GB	GENERAL	reserved.x86-64.2cpu.4gib
Linux	4	8 GiB	128GB	GENERAL	reserved.x86-64.4cpu.8gib
Linux	8	16 GiB	128GB	GENERAL	reserved.x86-64.8cpu.16gib
Linux	16	32 GiB	256GB	GENERAL	reserved.x86-64.16cpu.32gib
Linux	36	72 GiB	256GB	GENERAL	reserved.x86-64.36cpu.72gib
Linux	48	96 GiB	512GB	GENERAL	reserved.x86-64.48cpu.96gib

Tipe lingkungan	vCPUs	Memori	Ruang disk	Jenis mesin	Jenis instans komputasi
Linux	72	144 GiB	824GB	GENERAL	reserved.x86-64.72cpu.144gib
Linux	96	192 GiB	824GB	GENERAL	reserved.x86-64.96cpu.192gib
Linux	48	96 GiB	824GB (SSD)	NVME	reserved.x86-64.48cpu.96gib.nvme
Linux	72	144 GiB	824GB (SSD)	NVME	reserved.x86-64.72cpu.144gib.nvme
Linux EC2	2	4 GiB	64GB	GENERAL	reserved.x86-64.2cpu.4gib
Linux EC2	4	8 GiB	128GB	GENERAL	reserved.x86-64.4cpu.8gib
Linux EC2	8	16 GiB	128GB	GENERAL	reserved.x86-64.8cpu.16gib

Tipe lingkungan	vCPUs	Memori	Ruang disk	Jenis mesin	Jenis instans komputasi
GPU Linux	4	16 GiB	235GB (SSD)	NVME	reserved.gpu.4cpu.16gib.nvme
GPU Linux	8	32 GiB	435GB (SSD)	NVME	reserved.gpu.8cpu.32gib.nvme
GPU Linux	16	64 GiB	585GB (SSD)	NVME	reserved.gpu.16cpu.64gib.nvme
GPU Linux	32	128 GiB	885GB (SSD)	NVME	reserved.gpu.32cpu.128gib.nvme
GPU Linux	48	192 GiB	3785GB (SSD)	NVME	reserved.gpu.48cpu.192gib.nvme
macOS	8	24 GiB	128GB	GENERAL	reserved.arm.m2.8cpu.24gib
Windows	2	4 GiB	64GB	GENERAL	reserved.x86-64.2cpu.4gib

Tipe lingkungan	vCPUs	Memori	Ruang disk	Jenis mesin	Jenis instans komputasi
Windows	4	8 GiB	128GB	GENERAL	reserved.x86-64.4cpu.8gib
Windows	8	16 GiB	128GB	GENERAL	reserved.x86-64.8cpu.16gib
Windows	16	32 GiB	256GB	GENERAL	reserved.x86-64.16cpu.32gib
Windows	32	64 GiB	256GB	GENERAL	reserved.x86-64.32cpu.64gib
Windows	36	72 GiB	256GB	GENERAL	reserved.x86-64.36cpu.72gib
Windows	48	96 GiB	512GB	GENERAL	reserved.x86-64.48cpu.96gib
Windows	64	128 GiB	512GB	GENERAL	reserved.x86-64.64cpu.128gib

Tipe lingkungan	vCPUs	Memori	Ruang disk	Jenis mesin	Jenis instans komputasi
Windows	72	144 GiB	824GB	GENERAL	reserved.x86-64.72cpu.144gib
Windows	96	192 GiB	824GB	GENERAL	reserved.x86-64.96cpu.192gib
Windows EC2	4	8 GiB	128GB	GENERAL	reserved.x86-64.4cpu.8gib
Windows EC2	8	16 GiB	128GB	GENERAL	reserved.x86-64.8cpu.16gib

Untuk informasi selengkapnya tentang pengenal harga, lihat https://aws.amazon.com/codebuild/pricing/.

Europe (Ireland)

Tipe lingkungan	vCPUs	Memori	Ruang disk	Jenis mesin	Jenis instans komputasi
LENGAN	2	4 GiB	64GB	GENERAL	reserved.arm.2cpu.4gib
LENGAN	4	8 GiB	128GB	GENERAL	reserved.arm.4cpu.8gib

Tipe lingkungan	vCPUs	Memori	Ruang disk	Jenis mesin	Jenis instans komputasi
LENGAN	8	16 GiB	128GB	GENERAL	reserved.arm.8cpu.16gib
LENGAN	16	32 GiB	256GB	GENERAL	reserved.arm.16cpu.32gib
LENGAN	32	64 GiB	256GB	GENERAL	reserved.arm.32cpu.64gib
LENGAN	48	96 GiB	512GB	GENERAL	reserved.arm.48cpu.96gib
LENGAN	64	128 GiB	824GB	GENERAL	reserved.arm.64cpu.128gib
LENGAN EC2	2	4 GiB	64GB	GENERAL	reserved.arm.2cpu.4gib
LENGAN EC2	4	8 GiB	128GB	GENERAL	reserved.arm.4cpu.8gib
LENGAN EC2	8	16 GiB	128GB	GENERAL	reserved.arm.8cpu.16gib
Linux	2	4 GiB	64GB	GENERAL	reserved.x86-64.2cpu.4gib

Tipe lingkungan	vCPUs	Memori	Ruang disk	Jenis mesin	Jenis instans komputasi
Linux	4	8 GiB	128GB	GENERAL	reserved.x86-64.4cpu.8gib
Linux	8	16 GiB	128GB	GENERAL	reserved.x86-64.8cpu.16gib
Linux	16	32 GiB	256GB	GENERAL	reserved.x86-64.16cpu.32gib
Linux	36	72 GiB	256GB	GENERAL	reserved.x86-64.36cpu.72gib
Linux	48	96 GiB	512GB	GENERAL	reserved.x86-64.48cpu.96gib
Linux	72	144 GiB	824GB	GENERAL	reserved.x86-64.72cpu.144gib
Linux	96	192 GiB	824GB	GENERAL	reserved.x86-64.96cpu.192gib

Tipe lingkungan	vCPUs	Memori	Ruang disk	Jenis mesin	Jenis instans komputasi
Linux	48	96 GiB	824GB (SSD)	NVME	reserved.x86-64.48cpu.96gib.nvme
Linux	72	144 GiB	824GB (SSD)	NVME	reserved.x86-64.72cpu.144gib.nvme
Linux EC2	2	4 GiB	64GB	GENERAL	reserved.x86-64.2cpu.4gib
Linux EC2	4	8 GiB	128GB	GENERAL	reserved.x86-64.4cpu.8gib
Linux EC2	8	16 GiB	128GB	GENERAL	reserved.x86-64.8cpu.16gib
GPU Linux	4	16 GiB	235GB (SSD)	NVME	reserved.gpu.4cpu.16gib.nvme
GPU Linux	8	32 GiB	435GB (SSD)	NVME	reserved.gpu.8cpu.32gib.nvme

Tipe lingkungan	vCPUs	Memori	Ruang disk	Jenis mesin	Jenis instans komputasi
GPU Linux	16	64 GiB	585GB (SSD)	NVME	reserved.gpu.16cpu.64gib.nvme
GPU Linux	32	128 GiB	885GB (SSD)	NVME	reserved.gpu.32cpu.128gib.nvme
GPU Linux	48	192 GiB	3785GB (SSD)	NVME	reserved.gpu.48cpu.192gib.nvme
Windows	2	4 GiB	64GB	GENERAL	reserved.x86-64.2cpu.4gib
Windows	4	8 GiB	128GB	GENERAL	reserved.x86-64.4cpu.8gib
Windows	8	16 GiB	128GB	GENERAL	reserved.x86-64.8cpu.16gib
Windows	16	32 GiB	256GB	GENERAL	reserved.x86-64.16cpu.32gib

Tipe lingkungan	vCPUs	Memori	Ruang disk	Jenis mesin	Jenis instans komputasi
Windows	32	64 GiB	256GB	GENERAL	reserved.x86-64.32cpu.64gib
Windows	36	72 GiB	256GB	GENERAL	reserved.x86-64.36cpu.72gib
Windows	48	96 GiB	512GB	GENERAL	reserved.x86-64.48cpu.96gib
Windows	64	128 GiB	512GB	GENERAL	reserved.x86-64.64cpu.128gib
Windows	72	144 GiB	824GB	GENERAL	reserved.x86-64.72cpu.144gib
Windows	96	192 GiB	824GB	GENERAL	reserved.x86-64.96cpu.192gib
Windows EC2	4	8 GiB	128GB	GENERAL	reserved.x86-64.4cpu.8gib

Tipe lingkungan	vCPUs	Memori	Ruang disk	Jenis mesin	Jenis instans komputasi
Windows EC2	8	16 GiB	128GB	GENERAL	reserved.x86-64.8cpu.16gib

Untuk informasi selengkapnya tentang pengenal harga, lihat https://aws.amazon.com/codebuild/pricing/.

South America (São Paulo)

Tipe lingkungan	vCPUs	Memori	Ruang disk	Jenis mesin	Jenis instans komputasi
LENGAN	2	4 GiB	64GB	GENERAL	reserved.arm.2cpu.4gib
LENGAN	4	8 GiB	128GB	GENERAL	reserved.arm.4cpu.8gib
LENGAN	8	16 GiB	128GB	GENERAL	reserved.arm.8cpu.16gib
LENGAN	16	32 GiB	256GB	GENERAL	reserved.arm.16cpu.32gib
LENGAN	32	64 GiB	256GB	GENERAL	reserved.arm.32cpu.64gib
LENGAN	48	96 GiB	512GB	GENERAL	reserved.arm.48cpu.96gib

Tipe lingkungan	vCPUs	Memori	Ruang disk	Jenis mesin	Jenis instans komputasi
LENGAN EC2	2	4 GiB	64GB	`GENERAL`	`reserved.arm.2cpu.4gib`
LENGAN EC2	4	8 GiB	128GB	`GENERAL`	`reserved.arm.4cpu.8gib`
LENGAN EC2	8	16 GiB	128GB	`GENERAL`	`reserved.arm.8cpu.16gib`
Linux	2	4 GiB	64GB	`GENERAL`	`reserved.x86-64.2cpu.4gib`
Linux	4	8 GiB	128GB	`GENERAL`	`reserved.x86-64.4cpu.8gib`
Linux	8	16 GiB	128GB	`GENERAL`	`reserved.x86-64.8cpu.16gib`
Linux	16	32 GiB	256GB	`GENERAL`	`reserved.x86-64.16cpu.32gib`
Linux	36	72 GiB	256GB	`GENERAL`	`reserved.x86-64.36cpu.72gib`

Tipe lingkungan	vCPUs	Memori	Ruang disk	Jenis mesin	Jenis instans komputasi
Linux	48	96 GiB	512GB	`GENERAL`	`reserved.x86-64.48cpu.96gib`
Linux	72	144 GiB	824GB	`GENERAL`	`reserved.x86-64.72cpu.144gib`
Linux	96	192 GiB	824GB	`GENERAL`	`reserved.x86-64.96cpu.192gib`
Linux	48	96 GiB	824GB (SSD)	`NVME`	`reserved.x86-64.48cpu.96gib.nvme`
Linux	72	144 GiB	824GB (SSD)	`NVME`	`reserved.x86-64.72cpu.144gib.nvme`
Linux EC2	2	4 GiB	64GB	`GENERAL`	`reserved.x86-64.2cpu.4gib`
Linux EC2	4	8 GiB	128GB	`GENERAL`	`reserved.x86-64.4cpu.8gib`

Tipe lingkungan	vCPUs	Memori	Ruang disk	Jenis mesin	Jenis instans komputasi
Linux EC2	8	16 GiB	128GB	GENERAL	reserved.x86-64.8cpu.16gib
Windows	2	4 GiB	64GB	GENERAL	reserved.x86-64.2cpu.4gib
Windows	4	8 GiB	128GB	GENERAL	reserved.x86-64.4cpu.8gib
Windows	8	16 GiB	128GB	GENERAL	reserved.x86-64.8cpu.16gib
Windows	16	32 GiB	256GB	GENERAL	reserved.x86-64.16cpu.32gib
Windows	32	64 GiB	256GB	GENERAL	reserved.x86-64.32cpu.64gib
Windows	36	72 GiB	256GB	GENERAL	reserved.x86-64.36cpu.72gib
Windows	48	96 GiB	512GB	GENERAL	reserved.x86-64.48cpu.96gib

Tipe lingkungan	vCPUs	Memori	Ruang disk	Jenis mesin	Jenis instans komputasi
Windows	64	128 GiB	512GB	GENERAL	reserved.x86-64.64cpu.128gib
Windows	72	144 GiB	824GB	GENERAL	reserved.x86-64.72cpu.144gib
Windows	96	192 GiB	824GB	GENERAL	reserved.x86-64.96cpu.192gib
Windows EC2	4	8 GiB	128GB	GENERAL	reserved.x86-64.4cpu.8gib
Windows EC2	8	16 GiB	128GB	GENERAL	reserved.x86-64.8cpu.16gib

Untuk informasi selengkapnya tentang pengenal harga, lihat https://aws.amazon.com/codebuild/pricing/.

Untuk memilih jenis komputasi:

- Di CodeBuild konsol, di halaman konfigurasi armada Compute, pilih salah satu opsi dari vCPU, Memori, dan Disk. Untuk informasi selengkapnya, lihat Buat armada kapasitas yang dicadangkan.
- Untuk AWS CLI, jalankan `update-fleet` perintah `create-fleet` atau, tentukan nilai `computeType` to`ATTRIBUTE_BASED_COMPUTE`. Untuk informasi selengkapnya, lihat create-fleet atau update-fleet.

- Untuk AWS SDK, panggil `UpdateFleet` operasi yang setara dengan `CreateFleet` atau untuk bahasa pemrograman target Anda, tentukan nilai `computeType` to`ATTRIBUTE_BASED_COMPUTE`. Untuk informasi selengkapnya, lihat AWS Referensi SDK dan alat.

> **Note**
>
> Untuk AWS SDK AWS CLI dan, Anda masih dapat menggunakan `computeType` input seperti`BUILD_GENERAL1_SMALL`, untuk memilih jenis komputasi alih-alih. `ATTRIBUTE_BASED_COMPUTE` Untuk informasi selengkapnya, lihat Tentang jenis lingkungan sesuai permintaan.

Keluarga instans yang didukung

AWS CodeBuild mendukung contoh berikut untuk armada kapasitas cadangan:

- Tujuan umum: M5 | M5a | M5ad | M5d | M5dn | M5n | M5Zn | M6a | M6g | M6gd | M6i | M6id | M6idn | M6in | M7a | M7g | M7gd | M7i | | M8g | T3 | T3a | T4g M7i-flex
- Komputasi dioptimalkan: C5 | C5a | C5ad | C5d | C5n | C6a | C6g | C6gD | C6gN | C6i | C6id | C6in | C7a | C7g | C7gD | C7gN | C7i | | C8g C7i-flex
- Memori dioptimalkan: R5 | R5a | R5ad | R5b | R5d | R5dn | R5n | R6a | R6g | R6gd | R6i | R6idn | R6in | R6id | R7a | R7g | R7gd | R7i | R7iz | R8g | | | | | | | | | | | | | | | | X1 | X1e | x2Gd | X2IDn | x2iEDn | x2iEZn | x8g | z1d U-3tb1 U-6tb1 U-9tb1 U-12tb1 U-18tb1 U-24tb1 U7i-6tb U7i-8tb U7i-12tb U7in-16tb U7in-24tb U7in-32tb
- Penyimpanan dioptimalkan: D3 | D3en | I3 | i3EN | i4G | i4i | i7iE | i8G | iM4GN | IS4Gen
- Komputasi yang dipercepat: DL1 | DL2q | F1 | F2 | G4ad | G4dn | G5 | G5g | G6 | G6e | Gr6 | Inf1 | Inf2 | P3 | P3dN | P4d | P5 | P5e | P5en | Trn1 | Trn1n | Trn2 | VT1
- High-performance komputasi: HPC6a | HPC6id | HPC7a | HPC7g
- Generasi sebelumnya: A1

Untuk membuat armada kapasitas cadangan dengan jenis instans tertentu:

- Di CodeBuild konsol, di halaman konfigurasi armada Compute, navigasikan ke bagian Konfigur asi kapasitas. Dalam mode pemilihan Komput asi, pilih Input manual dan di Jenis instance Compute pilih salah satu jenis instance dari menu drop-down. Untuk informasi selengkapnya, lihat Buat armada kapasitas yang dicadangkan.

- Untuk AWS CLI, jalankan `update-fleet` perintah `create-fleet` or, tentukan nilai `computeType` ke `CUSTOM_INSTANCE_TYPE` dan `ComputeConfiguration` `instanceType` ke jenis instance yang ditentukan. Untuk informasi selengkapnya, lihat create-fleet atau update-fleet.
- Untuk AWS SDK, panggil `UpdateFleet` operasi yang setara dengan `CreateFleet` atau untuk bahasa pemrograman target Anda, tentukan nilai `computeType` to `CUSTOM_INSTANCE_TYPE` dan ComputeConfiguration `instanceType` ke jenis instance yang ditentukan. Untuk informasi selengkapnya, lihat AWS Referensi SDK dan alat.

Tentang jenis lingkungan sesuai permintaan

AWS CodeBuild menyediakan lingkungan build dengan memori, vCPU, dan ruang disk yang tersedia berikut untuk mode komputasi EC2:

Jenis komputasi	Nilai ComputeType lingkungan	Nilai jenis lingkungan	Memori	vCPUs	Ruang disk
ARM Kecil [1]	`BUILD_GENERAL1_SMALL`	`ARM_CONTAINER` `ARM_EC2`	4 GiB	2	64GB
ARM Sedang [1]	`BUILD_GENERAL1_MEDIUM`	`ARM_CONTAINER` `ARM_EC2`	8 GiB	4	128GB
ARM Besar [1]	`BUILD_GENERAL1_LARGE`	`ARM_CONTAINER` `ARM_EC2`	16 GiB	8	128GB
ARM XLarge[1]	`BUILD_GENERAL1_XLARGE`	`ARM_CONTAINER`	64 GiB	32	256GB

Jenis komputasi	Nilai ComputeType lingkungan	Nilai jenis lingkungan	Memori	vCPUs	Ruang disk
ARM 2xLarge[1]	`BUILD_GENERAL1_2XLARGE`	`ARM_CONTAINER`	96 GiB	48	824GB
Linux Kecil [1]	`BUILD_GENERAL1_SMALL`	`LINUX_CONTAINER` `LINUX_EC2`	4 GiB	2	64GB
Linux Sedang [1]	`BUILD_GENERAL1_MEDIUM`	`LINUX_CONTAINER` `LINUX_EC2`	8 GiB	4	128GB
Linux Besar [1]	`BUILD_GENERAL1_LARGE`	`LINUX_CONTAINER` `LINUX_EC2`	16 GiB	8	128GB
Linux XLarge[1]	`BUILD_GENERAL1_XLARGE`	`LINUX_CONTAINER`	72 GiB	36	256GB
Linux 2xBesar	`BUILD_GENERAL1_2XLARGE`	`LINUX_CONTAINER`	144 GiB	72	824GB (SSD)
Linux GPU Kecil	`BUILD_GENERAL1_SMALL`	`LINUX_GPU_CONTAINER`	16 GiB	4	235GB (SSD)
GPU Linux Besar	`BUILD_GENERAL1_LARGE`	`LINUX_GPU_CONTAINER`	255 GiB	32	50 GB

Jenis komputasi	Nilai ComputeType lingkungan	Nilai jenis lingkungan	Memori	vCPUs	Ruang disk

Windows Sedang [1]	`BUILD_GENERAL1_MEDIUM`	`WINDOWS_SERVER_2019_CONTAINER`
		`WINDOWS_SERVER_2022_CONTAINER`
		`WINDOWS_EC2`
			8 GiB	4	128GB

Windows Besar [1]	`BUILD_GENERAL1_LARGE`	`WINDOWS_SERVER_2019_CONTAINER`
		`WINDOWS_SERVER_2022_CONTAINER`
		`WINDOWS_EC2`
			16 GiB	8	128GB

Windows XLarge [1]	`BUILD_GENERAL1_XLARGE`	`WINDOWS_SERVER_2022_CONTAINER`	72 GiB	36	256GB

Jenis komputasi	Nilai ComputeType lingkungan	Nilai jenis lingkungan	Memori	vCPUs	Ruang disk
Windows 2xLarge[1]	`BUILD_GENERAL1_2XLARGE`	`WINDOWS_SERVER_2022_CONTAINER`	144 GiB	72	824GB

[1] Versi terbaru dari jenis gambar ini di-cache. Jika Anda menentukan versi yang lebih spesifik, maka menyediakan CodeBuild versi itu alih-alih versi cache. Ini dapat menghasilkan waktu pembuatan yang lebih lama. Misalnya, untuk mendapatkan manfaat dari caching, tentukan al `aws/codebuild/amazonlinux-x86_64-standard:5.0` ih-alih versi yang lebih granular, sepertia`ws/codebuild/amazonlinux-x86_64-standard:5.0-1.0.0`.

AWS CodeBuild menyediakan lingkungan build dengan memori dan ruang disk yang tersedia berikut untuk mode AWS Lambda komputasi:

Jenis komputasi	Nilai ComputeType lingkungan	Nilai jenis lingkungan	Memori	Ruang disk
ARM Lambda 1GB	`BUILD_LAMBDA_1GB`	`ARM_LAMBDA_CONTAINER`	1 GiB	10 GB
ARM Lambda 2GB	`BUILD_LAMBDA_2GB`	`ARM_LAMBDA_CONTAINER`	2 GiB	10 GB
ARM Lambda 4GB	`BUILD_LAMBDA_4GB`	`ARM_LAMBDA_CONTAINER`	4 GiB	10 GB
ARM Lambda 8GB	`BUILD_LAMBDA_8GB`	`ARM_LAMBDA_CONTAINER`	8 GiB	10 GB

Jenis komputasi	Nilai ComputeType lingkungan	Nilai jenis lingkungan	Memori	Ruang disk
ARM Lambda 10GB	`BUILD_LAMBDA_10GB`	`ARM_LAMBDA_CONTAINER`	10 GiB	10 GB
Linux Lambda 1GB	`BUILD_LAMBDA_1GB`	`LINUX_LAMBDA_CONTAINER`	1 GiB	10 GB
Linux Lambda 2GB	`BUILD_LAMBDA_2GB`	`LINUX_LAMBDA_CONTAINER`	2 GiB	10 GB
Linux Lambda 4GB	`BUILD_LAMBDA_4GB`	`LINUX_LAMBDA_CONTAINER`	4 GiB	10 GB
Linux Lambda 8 GB	`BUILD_LAMBDA_8GB`	`LINUX_LAMBDA_CONTAINER`	8 GiB	10 GB
Linux Lambda 10GB	`BUILD_LAMBDA_10GB`	`LINUX_LAMBDA_CONTAINER`	10 GiB	10 GB

Saat menggunakan jenis lingkungan lain, disarankan agar Anda menggunakan gambar yang di-cache untuk mengurangi waktu pembuatan.

Ruang disk yang terdaftar untuk setiap lingkungan build hanya tersedia di direktori yang ditentukan oleh variabel `CODEBUILD_SRC_DIR` lingkungan.

Untuk memilih jenis komputasi:

- Di CodeBuild konsol, di panduan Buat proyek build atau halaman Edit Build Project, di Environment memperluas Konfigurasi tambahan, lalu pilih salah satu opsi dari Jenis Komputasi.

Untuk informasi selengkapnya, lihat Buat proyek build (konsol) atau Mengubah setelan proyek build (konsol).

- Untuk AWS CLI, jalankan `update-project` perintah `create-project` atau, menentukan `computeType` nilai `environment` objek. Untuk informasi selengkapnya, lihat Buat proyek build (AWS CLI) atau Mengubah setelan proyek build (AWS CLI).
- Untuk AWS SDK, panggil `UpdateProject` operasi yang setara dengan `CreateProject` atau untuk bahasa pemrograman target Anda, tentukan setara `computeType` nilai `environment` objek. Untuk informasi selengkapnya, lihat AWS Referensi SDK dan alat.

Beberapa jenis lingkungan dan komputasi memiliki keterbatasan ketersediaan Wilayah:

- Jenis komputasi Linux GPU Small (`LINUX_GPU_CONTAINER`) hanya tersedia di Wilayah ini:
  - AS Timur (Virginia Utara)
  - AS Barat (Oregon)
  - Asia Pasifik (Tokyo)
  - Kanada (Pusat)
  - Eropa (Frankfurt)
  - Eropa (Irlandia)
  - Eropa (London)
- Jenis komputasi Linux GPU Large (`LINUX_GPU_CONTAINER`) hanya tersedia di Wilayah ini:
  - AS Timur (Ohio)
  - AS Timur (Virginia Utara)
  - AS Barat (Oregon)
  - Asia Pasifik (Seoul)
  - Asia Pasifik (Sydney)
  - Asia Pasifik (Tokyo)
  - Kanada (Pusat)
  - Tiongkok (Beijing)
  - Tiongkok (Ningxia)
  - Eropa (Frankfurt)
  - Eropa (Irlandia)
  - Eropa (London)

- Jenis komputasi hanya `BUILD_GENERAL1_2XLARGE` tersedia di Wilayah ini:
  - AS Timur (Ohio)
  - AS Timur (Virginia Utara)
  - AS Barat (California Utara)
  - AS Barat (Oregon)
  - Asia Pasifik (Hyderabad)
  - Asia Pasifik (Hong Kong)
  - Asia Pasifik (Jakarta)
  - Asia Pasifik (Melbourne)
  - Asia Pasifik (Mumbai)
  - Asia Pasifik (Seoul)
  - Asia Pasifik (Singapura)
  - Asia Pasifik (Sydney)
  - Asia Pasifik (Tokyo)
  - Kanada (Pusat)
  - Tiongkok (Beijing)
  - Tiongkok (Ningxia)
  - Eropa (Frankfurt)
  - Eropa (Irlandia)
  - Eropa (London)
  - Eropa (Paris)
  - Eropa (Spanyol)
  - Eropa (Stockholm)
  - Eropa (Zürich)
  - Israel (Tel Aviv)
  - Timur Tengah (Bahrain)
  - Timur Tengah (UEA)
  - Amerika Selatan (Sao Paulo)
- Jenis lingkungan hanya `ARM_CONTAINER` tersedia di Wilayah ini:
  - AS Timur (Ohio)

- AS Timur (Virginia Utara)
- AS Barat (California Utara)
- AS Barat (Oregon)
- Asia Pasifik (Hong Kong)
- Asia Pasifik (Jakarta)
- Asia Pasifik (Hyderabad)
- Asia Pasifik (Malaysia)
- Asia Pasifik (Mumbai)
- Asia Pasifik (Selandia Baru)
- Asia Pasifik (Osaka)
- Asia Pasifik (Seoul)
- Asia Pasifik (Singapura)
- Asia Pasifik (Sydney)
- Asia Pasifik (Thailand)
- Asia Pasifik (Tokyo)
- Kanada (Pusat)
- Tiongkok (Beijing)
- Tiongkok (Ningxia)
- Eropa (Frankfurt)
- Eropa (Irlandia)
- Eropa (London)
- Eropa (Milan)
- Eropa (Paris)
- Eropa (Spanyol)
- Eropa (Stockholm)
- Israel (Tel Aviv)
- Timur Tengah (Bahrain)
- Timur Tengah (UEA)
- Amerika Selatan (Sao Paulo)

- Jenis lingkungan hanya `WINDOWS_SERVER_2022_CONTAINER` tersedia di Wilayah ini:

- AS Timur (Ohio)
- AS Timur (Virginia Utara)
- US West (Oregon)
- Asia Pasifik (Sydney)
- Asia Pasifik (Tokyo)
- Eropa (Frankfurt)
- Europe (Ireland)
- Amerika Selatan (Sao Paulo)

- Jenis lingkungan `LINUX_EC2` (`BUILD_GENERAL1_SMALL`,`BUILD_GENERAL1_MEDIUM`,`BUILD_GENERAL1_LARGE`) hanya tersedia di Wilayah ini:
  - AS Timur (Ohio)
  - AS Timur (Virginia Utara)
  - AS Barat (California Utara)
  - AS Barat (Oregon)
  - Afrika (Cape Town)
  - Asia Pasifik (Hong Kong)
  - Asia Pasifik (Jakarta)
  - Asia Pacific (Melbourne)
  - Asia Pasifik (Hyderabad)
  - Asia Pasifik (Malaysia)
  - Asia Pasifik (Mumbai)
  - Asia Pasifik (Selandia Baru)
  - Asia Pasifik (Osaka)
  - Asia Pasifik (Seoul)
  - Asia Pasifik (Singapura)
  - Asia Pasifik (Sydney)
  - Asia Pasifik (Thailand)
  - Asia Pasifik (Tokyo)
  - Kanada (Pusat)
  - Tiongkok (Beijing)

- Tiongkok (Ningxia)
- Eropa (Frankfurt)
- Eropa (Irlandia)
- Eropa (London)
- Eropa (Milan)
- Eropa (Paris)
- Eropa (Spanyol)
- Eropa (Stockholm)
- Eropa (Zürich)
- Israel (Tel Aviv)
- Timur Tengah (Bahrain)
- Timur Tengah (UEA)
- Amerika Selatan (Sao Paulo)
- AWS GovCloud (US-West)
- AWS GovCloud (US-East)

- Jenis lingkungan `ARM_EC2` (`BUILD_GENERAL1_SMALL`,`BUILD_GENERAL1_MEDIUM`,`BUILD_GENERAL1_LARGE`) hanya tersedia di Wilayah ini:
  - AS Timur (Ohio)
  - AS Timur (Virginia Utara)
  - AS Barat (California Utara)
  - AS Barat (Oregon)
  - Asia Pasifik (Hong Kong)
  - Asia Pasifik (Jakarta)
  - Asia Pasifik (Hyderabad)
  - Asia Pasifik (Malaysia)
  - Asia Pasifik (Mumbai)
  - Asia Pasifik (Selandia Baru)
  - Asia Pasifik (Osaka)
  - Asia Pasifik (Seoul)
  - Asia Pasifik (Thailand)

- Asia Pasifik (Singapura)
- Asia Pasifik (Sydney)
- Asia Pasifik (Tokyo)
- Kanada (Pusat)
- Tiongkok (Beijing)
- Tiongkok (Ningxia)
- Eropa (Frankfurt)
- Eropa (Irlandia)
- Eropa (London)
- Eropa (Milan)
- Eropa (Paris)
- Eropa (Spanyol)
- Eropa (Stockholm)
- Eropa (Zürich)
- Israel (Tel Aviv)
- Timur Tengah (Bahrain)
- Amerika Selatan (Sao Paulo)
- AWS GovCloud (US-West)
- AWS GovCloud (US-East)

- Jenis lingkungan `WINDOWS_EC2` (`BUILD_GENERAL1_MEDIUM`,`BUILD_GENERAL1_LARGE`) hanya tersedia di Wilayah ini:
  - AS Timur (Ohio)
  - AS Timur (Virginia Utara)
  - US West (Oregon)
  - Asia Pasifik (Sydney)
  - Asia Pasifik (Tokyo)
  - Eropa (Frankfurt)
  - Europe (Ireland)
  - Amerika Selatan (Sao Paulo)
- Mode komputasi AWS Lambda (`ARM_LAMBDA_CONTAINER`dan`LINUX_LAMBDA_CONTAINER`) hanya tersedia di Wilayah ini:

- Timur AS (N. Virginia)
- AS Timur (Ohio)
- AS Barat (California Utara)
- AS Barat (Oregon)
- Asia Pasifik (Mumbai)
- Asia Pasifik (Singapura)
- Asia Pasifik (Sydney)
- Asia Pasifik (Tokyo)
- Eropa (Frankfurt)
- Europe (Ireland)
- Amerika Selatan (Sao Paulo)

- Mode komputasi hanya `MAC_ARM` tersedia di Wilayah ini:
  - Timur AS (N. Virginia)
  - AS Timur (Ohio)
  - AS Barat (Oregon)
  - Asia Pacific (Sydney)
  - Eropa (Frankfurt)

Untuk tipe komputasi`BUILD_GENERAL1_2XLARGE`, gambar Docker hingga 100 GB tanpa kompresi didukung.

> **Note**
>
> Untuk gambar lingkungan build khusus, CodeBuild mendukung gambar Docker hingga 50 GB tanpa kompresi di Linux dan Windows, terlepas dari jenis komputasi. Untuk memeriksa ukuran image build Anda, gunakan Docker untuk menjalankan `docker images` *`REPOSITORY`*`:`*`TAG`* perintah.

Anda dapat menggunakan Amazon EFS untuk mengakses lebih banyak ruang di wadah build Anda. Untuk informasi selengkapnya, lihat Contoh Sistem File Amazon Elastic untuk AWS CodeBuild. Jika Anda ingin memanipulasi ruang disk kontainer selama build, maka build harus berjalan dalam mode istimewa.

> **Note**
>
> Secara default, daemon Docker diaktifkan untuk build non-VPC. Jika Anda ingin menggunakan wadah Docker untuk build VPC, lihat Runtime Privilege dan Kapabilitas Linux di situs Docker Docker dan aktifkan mode istimewa. Selain itu, Windows tidak mendukung mode istimewa.

Shell dan perintah di lingkungan build

Anda menyediakan serangkaian perintah AWS CodeBuild untuk dijalankan di lingkungan build selama siklus hidup build (misalnya, menginstal dependensi build dan menguji serta mengompilasi kode sumber Anda). Ada beberapa cara untuk menentukan perintah ini:

- Buat file spesifikasi build dan sertakan dengan kode sumber Anda. Dalam file ini, tentukan perintah yang ingin Anda jalankan di setiap fase siklus hidup build. Untuk informasi selengkapnya, lihat Buat referensi spesifikasi untuk CodeBuild.
- Gunakan CodeBuild konsol untuk membuat proyek build. Dalam Insert perintah build, untuk perintah Build, masukkan perintah yang ingin Anda jalankan dalam `build` fase. Untuk informasi selengkapnya, lihat Buat proyek build (konsol).
- Gunakan CodeBuild konsol untuk mengubah pengaturan proyek build. Dalam Insert perintah build, untuk perintah Build, masukkan perintah yang ingin Anda jalankan dalam `build` fase. Untuk informasi selengkapnya, lihat Mengubah setelan proyek build (konsol).
- Gunakan AWS CLI atau AWS SDK untuk membuat proyek build atau mengubah setelan proyek build. Referensikan kode sumber yang berisi file buildspec dengan perintah Anda, atau tentukan satu string yang menyertakan konten file buildspec yang setara. Untuk informasi selengkapnya, lihat Membuat proyek build atau Ubah pengaturan proyek build.
- Gunakan AWS CLI atau AWS SDK untuk memulai build, menentukan file buildspec atau string tunggal yang menyertakan konten file buildspec yang setara. Untuk informasi selengkapnya, lihat deskripsi untuk `buildspecOverride` nilai diJalankan build secara manual.

Anda dapat menentukan perintah Shell Command Language (sh). Di buildspec versi 0.1, CodeBuild jalankan setiap perintah Shell dalam instance terpisah di lingkungan build. Ini berarti bahwa setiap perintah berjalan secara terpisah dari semua perintah lainnya. Oleh karena itu, secara default, Anda tidak dapat menjalankan satu perintah yang bergantung pada status perintah sebelumnya (misalnya, mengubah direktori atau mengatur variabel lingkungan). Untuk mengatasi batasan ini, kami sarankan

Anda menggunakan versi 0.2, yang memecahkan masalah ini. Jika Anda harus menggunakan versi 0.1, kami merekomendasikan pendekatan berikut:

- Sertakan skrip shell dalam kode sumber Anda yang berisi perintah yang ingin Anda jalankan dalam satu contoh shell default. Misalnya, Anda dapat menyertakan file bernama `my-script.sh` dalam kode sumber Anda yang berisi perintah seperti`cd MyDir; mkdir -p mySubDir; cd mySubDir; pwd;`. Kemudian, di file buildspec Anda, tentukan perintahnya. `./my-script.sh`
- Di file buildspec Anda atau pada pengaturan perintah Build untuk `build` fase saja, masukkan satu perintah yang menyertakan semua perintah yang ingin Anda jalankan dalam satu instance shell default (misalnya,). `cd MyDir && mkdir -p mySubDir && cd mySubDir && pwd`

Jika CodeBuild menemukan kesalahan, kesalahan mungkin lebih sulit untuk memecahkan masalah dibandingkan dengan menjalankan satu perintah dalam instance shell defaultnya sendiri.

Perintah yang dijalankan dalam gambar Windows Server Core menggunakan PowerShell shell.

Variabel lingkungan di lingkungan build

AWS CodeBuild menyediakan beberapa variabel lingkungan yang dapat Anda gunakan dalam perintah build Anda:

AWS_DEFAULT_REGION

Wil AWS ayah tempat build berjalan (misalnya,`us-east-1`). Variabel lingkungan ini digunakan terutama oleh AWS CLI.

AWS_REGION

Wil AWS ayah tempat build berjalan (misalnya,`us-east-1`). Variabel lingkungan ini digunakan terutama oleh AWS SDK.

CODEBUILD_BATCH_BUILD_IDENTIFIER

Pengidentifikasi build dalam build batch. Ini ditentukan dalam batch buildspec. Untuk informasi selengkapnya, lihat the section called “Referensi buildspec batch”.

CODEBUILD_BUILD_ARN

Nama Sumber Daya Amazon (ARN) dari build (misalnya,`arn:aws:codebuild:`*`region-ID`*`:`*`account-ID`*`:build/codebuild-demo-project:b1e6661e-e4f2-4156-9ab9-82a19EXAMPLE`).

CODEBUILD_BUILD_ID

CodeBuild ID build (misalnya,`codebuild-demo-project:b1e6661e-e4f2-4156-9ab9-82a19EXAMPLE`).

CODEBUILD_BUILD_IMAGE

Pen CodeBuild gidentifikasi gambar build (misalnya,`aws/codebuild/standard:2.0`).

CODEBUILD_BUILD_NUMBER

Nomor build saat ini untuk proyek.

CODEBUILD_BUILD_SUCCESS

Apakah pembangunan saat ini berhasil. Setel ke `0` jika build gagal, atau `1` jika build berhasil.

CODEBUILD_INITIATOR

Entitas yang memulai pembangunan. Jika CodePipeline memulai pembangunan, ini adalah nama pipeline (misalnya,`codepipeline/my-demo-pipeline`). Jika pengguna memulai build, ini adalah nama pengguna (misalnya,`MyUserName`). Jika plugin Jenkins untuk CodeBuild memulai pembangunan, ini adalah string`CodeBuild-Jenkins-Plugin`.

CODEBUILD_KMS_KEY_ID

Pengidentifikasi AWS KMS kunci yang CodeBuild digunakan untuk mengenkripsi artefak keluaran build (misalnya, `arn:aws:kms:`*`region-ID`*`:`*`account-ID`*`:key/`*`key-ID`* atau`alias/`*`key-alias`*).

CODEBUILD_PROJECT_ARN

Nama Sumber Daya Amazon (ARN) dari proyek (misalnya,`arn:aws:codebuild:`*`region-ID`*`:`*`account-ID`*`:project/`*`project-name`*).

CODEBUILD_PUBLIC_BUILD_URL

URL hasil build untuk build ini di situs web build publik. Variabel ini hanya disetel jika proyek build mengaktifkan build publik. Untuk informasi selengkapnya, lihat Dapatkan URL proyek build publik.

CODEBUILD_TERSELESAIKAN_VERSI_SUMBER

Pengidentifikasi versi dari kode sumber build. Isinya tergantung pada repositori kode sumber: CodeCommit, GitHub, Server GitHub Perusahaan, dan Bitbucket

Variabel ini berisi Commit ID.

CodePipeline

Variabel ini berisi revisi sumber yang disediakan oleh CodePipeline.

Jika CodePipeline tidak dapat menyelesaikan revisi sumber, seperti ketika sumbernya adalah bucket Amazon S3 yang tidak mengaktifkan pembuatan versi, variabel lingkungan ini tidak disetel.

Amazon S3

Variabel ini tidak diatur.

Jika berlaku, `CODEBUILD_RESOLVED_SOURCE_VERSION` variabel hanya tersedia setelah `DOWNLOAD_SOURCE` fase.

CODEBUILD_SOURCE_REPO_URL

URL ke artefak input atau repositori kode sumber. Untuk Amazon S3, ini `s3://` diikuti oleh nama bucket dan jalur ke artefak input. Untuk CodeCommit dan GitHub, ini adalah URL klon repositori. Jika build berasal dari CodePipeline, variabel lingkungan ini mungkin kosong.

Untuk sumber sekunder, variabel lingkungan untuk URL repositori sumber sekunder adalah`CODEBUILD_SOURCE_REPO_URL_<sourceIdentifier>`, di mana pen *`<sourceIdentifier>`* gidentifikasi sumber yang Anda buat.

KODEBUILD_VERSI_SUMBER

Format nilai tergantung pada repositori sumber.

- Untuk Amazon S3, ini adalah ID versi yang terkait dengan artefak input.
- Untuk CodeCommit, ini adalah ID komit atau nama cabang yang terkait dengan versi kode sumber yang akan dibangun.
- Untuk GitHub, GitHub Enterprise Server, dan Bitbucket itu adalah ID commit, nama cabang, atau nama tag yang terkait dengan versi kode sumber yang akan dibangun.

> **Note**
>
> Untuk build Server GitHub Perusahaan GitHub atau yang dipicu oleh peristiwa permintaan tarik webhook, memang `pr/`*`pull-request-number`* demikian.

Untuk sumber sekunder, variabel lingkungan untuk versi sumber sekunder adalah`CODEBUILD_SOURCE_VERSION_<sourceIdentifier>`, di mana pen

<sourceIdentifier> gidentifikasi sumber yang Anda buat. Untuk informasi selengkapnya, lihat Beberapa sumber input dan sampel artefak keluaran.

CODEBUILD_SRC_DIR

Jalur direktori CodeBuild yang digunakan untuk build (misalnya,`/tmp/src123456789/src`).

Untuk sumber sekunder, variabel lingkungan untuk jalur direktori sumber sekunder adalah`CODEBUILD_SRC_DIR_`*<sourceIdentifier>*, di mana pen *<sourceIdentifier>* gidentifikasi sumber yang Anda buat. Untuk informasi selengkapnya, lihat Beberapa sumber input dan sampel artefak keluaran.

KODEBUILD_WAKTU_MULAI

Waktu mulai build ditentukan sebagai stempel waktu Unix dalam milidetik.

CODEBUILD_WEBHOOK_ACTOR_AKUN_ID

ID akun pengguna yang memicu peristiwa webhook.

CODEBUILD_WEBHOOK_BASE_REF

Nama referensi dasar dari peristiwa webhook yang memicu build saat ini. Untuk permintaan tarik, ini adalah referensi cabang.

CODEBUILD_WEBHOOK_EVENT

Peristiwa webhook yang memicu build saat ini.

CODEBUILD_WEBHOOK_MERGE_COMMIT

Pengidentifikasi komit gabungan yang digunakan untuk build. Variabel ini diatur ketika permintaan tarik Bitbucket digabungkan dengan strategi squash dan cabang permintaan tarik ditutup. Dalam hal ini, komit permintaan tarik asli tidak ada lagi, jadi variabel lingkungan ini berisi pengenal komit gabungan yang dihancurkan.

CODEBUILD_WEBHOOK_PREV_COMMIT

ID komit terbaru sebelum acara push webhook yang memicu build saat ini.

CODEBUILD_WEBHOOK_HEAD_REF

Nama referensi kepala dari peristiwa webhook yang memicu build saat ini. Ini bisa berupa referensi cabang atau referensi tag.

CODEBUILD_WEBHOOK_TRIGGER

Menunjukkan peristiwa webhook yang memicu build. Variabel ini hanya tersedia untuk build yang dipicu oleh webhook. Nilai diurai dari payload yang dikirim CodeBuild oleh GitHub, GitHub Enterprise Server, atau Bitbucket. Format nilai tergantung pada jenis peristiwa apa yang memicu build.

- Untuk build yang dipicu oleh permintaan tarik, memang demikian`pr/`*`pull-request-number`*.
- Untuk build yang dipicu dengan membuat cabang baru atau mendorong komit ke cabang, memang demikian`branch/`*`branch-name`*.
- Untuk build yang dipicu oleh mendorong tag ke repositori, memang demikian`tag/`*`tag-name`*.

RUMAH

Variabel lingkungan ini selalu disetel ke`/root`.

AWS CodeBuild juga mendukung satu set variabel lingkungan untuk build pelari yang di-host sendiri. Untuk mempelajari lebih lanjut tentang pelari CodeBuild yang dihosting sendiri, lihatTutorial: Konfigurasikan pelari CodeBuild-hosted GitHub Tindakan.

CODEBUILD_RUNNER_OWNER

Pemilik repositori yang memicu build runner yang di-host sendiri.

CODEBUILD_RUNNER_REPO

Nama repositori yang memicu build runner yang di-host sendiri.

CODEBUILD_RUNNER_REPO_DOMAIN

Domain repositori yang memicu build runner yang di-host sendiri. Hanya build GitHub Enterprise yang ditentukan.

CODEBUILD_WEBHOOK_LABEL

Label yang digunakan untuk mengonfigurasi penggantian build dan pelari yang di-host sendiri selama pembuatan.

CODEBUILD_WEBHOOK_RUN_ID

ID run alur kerja yang terkait dengan build.

CODEBUILD_WEBHOOK_JOB_ID

ID pekerjaan dari pekerjaan yang terkait dengan build.

CODEBUILD_WEBHOOK_NAMA_ALUR KERJA

Nama alur kerja yang terkait dengan build jika ada di payload permintaan webhook.

CODEBUILD_RUNNER_DENGAN_BUILDSPEC

Jika penggantian buildspec dikonfigurasi di label permintaan pelari yang di-host sendiri, ini disetel ke. `true`

Anda juga dapat menyediakan lingkungan build dengan variabel lingkungan Anda sendiri. Untuk informasi selengkapnya, lihat topik berikut:

- Gunakan CodeBuild dengan CodePipeline
- Membuat proyek build
- Ubah pengaturan proyek build
- Jalankan build secara manual
- Referensi Buildspec

Untuk membuat daftar semua variabel lingkungan yang tersedia di lingkungan build, Anda dapat menjalankan `printenv` perintah (untuk lingkungan Linux-based build) atau `"Get-ChildItem Env:"` (untuk lingkungan Windows-based build) selama build. Kecuali yang terdaftar sebelumnya, variabel lingkungan yang dimulai dengan `CODEBUILD_` adalah untuk penggunaan CodeBuild internal. Mereka tidak boleh digunakan dalam perintah build Anda.

Important

Kami sangat tidak menyarankan penggunaan variabel lingkungan untuk menyimpan nilai sensitif, terutama ID kunci AWS akses. Variabel lingkungan dapat ditampilkan dalam teks biasa menggunakan alat seperti CodeBuild konsol dan AWS CLI.
Kami sarankan Anda menyimpan nilai sensitif di Amazon EC2 Systems Manager Parameter Store dan kemudian mengambilnya dari buildspec Anda. Untuk menyimpan nilai sensitif, lihat Peny impanan Parameter dan Panduan Manajer Sistem: Membuat dan menguji parameter String (konsol) di Panduan Pengguna Amazon EC2 Systems Manager. Untuk mengambilnya, lihat pem `parameter-store` etaan diSintaks Buildspec.

CODEBUILD_BUILD_URL

URL hasil build untuk build ini.

Tugas latar belakang di lingkungan build

Anda dapat menjalankan tugas latar belakang di lingkungan build. Untuk melakukan ini, di buildspec Anda, gunakan nohup perintah untuk menjalankan perintah sebagai tugas di latar belakang, bahkan jika proses build keluar dari shell. Gunakan disown perintah untuk menghentikan tugas latar belakang yang berjalan secara paksa.

Contoh:

- Mulai proses latar belakang dan tunggu sampai selesai nanti:

```
|
nohup sleep 30 & echo $! > pidfile
…
wait $(cat pidfile)
```

- Mulai proses latar belakang dan jangan menunggu sampai selesai:

```
|
nohup sleep 30 & disown $!
```

- Mulai proses latar belakang dan matikan nanti:

```
|
nohup sleep 30 & echo $! > pidfile
…
kill $(cat pidfile)
```

Proyek build

Proyek build mencakup informasi tentang cara menjalankan build, termasuk tempat mendapatkan kode sumber, lingkungan build mana yang akan digunakan, perintah build mana yang akan dijalankan, dan tempat menyimpan output build.

Anda dapat melakukan tugas-tugas ini saat bekerja dengan proyek build:

Topik

- Membuat proyek pembangunan di AWS CodeBuild
- Membuat aturan notifikasi
- Ubah pengaturan proyek build di AWS CodeBuild
- Beberapa token akses di CodeBuild
- Hapus proyek build di AWS CodeBuild
- Dapatkan URL proyek build publik
- Bagikan proyek pembangunan
- Tag membangun proyek
- Gunakan pelari dengan AWS CodeBuild
- Gunakan webhook dengan AWS CodeBuild
- Lihat detail proyek build di AWS CodeBuild
- Lihat nama proyek build di AWS CodeBuild

Membuat proyek pembangunan di AWS CodeBuild

Anda dapat menggunakan AWS CodeBuild konsol AWS CLI, atau AWS SDK untuk membuat proyek build.

Topik

- Prasyarat
- Buat proyek build (konsol)
- Buat proyek build (AWS CLI)
- Buat proyek build (AWS SDK)
- Buat proyek build (CloudFormation)

Prasyarat

Sebelum membuat proyek build, jawab pertanyaan diMerencanakan build.

Buat proyek build (konsol)

Buka AWS CodeBuild konsol di https://console.aws.amazon.com/codesuite/codebuild/home.

Jika halaman CodeBuild informasi ditampilkan, pilih Buat proyek build. Jika tidak, pada panel navigasi, perluas Build, pilih Build projects, lalu pilih Create build project.

Pilih Buat proyek build.

Isi bagian berikut. Setelah selesai, pilih Buat proyek build di bagian bawah halaman.

Bagian:

- Konfigurasi proyek
- Sumber
- Lingkungan
- Buildspec
- Konfigurasi batch
- Artifacts
- Beberapa catatan

Konfigurasi proyek

Nama proyek

Masukkan nama untuk proyek build ini. Membangun nama proyek harus unik di setiap AWS akun.

Deskripsi

Masukkan deskripsi opsional proyek build untuk membantu pengguna lain memahami untuk apa proyek ini digunakan.

Membangun lencana

(Opsional) Pilih Aktifkan lencana build untuk membuat status build proyek Anda terlihat dan dapat disematkan. Untuk informasi selengkapnya, lihat Bangun contoh lencana.

> **Note**
>
> Build badge tidak berlaku jika penyedia sumber Anda adalah Amazon S3.

Aktifkan batas build bersamaan

(Opsional) Jika Anda ingin membatasi jumlah build bersamaan untuk proyek ini, lakukan langkah-langkah berikut:

1. Pilih Batasi jumlah build bersamaan yang dapat dimulai proyek ini.
2. Dalam batas build bersamaan, masukkan jumlah maksimum build bersamaan yang diizinkan untuk project ini. Batas ini tidak boleh lebih besar dari batas build bersamaan yang ditetapkan untuk akun. Jika Anda mencoba memasukkan angka yang lebih besar dari batas akun, pesan kesalahan akan ditampilkan.

Pembuatan baru hanya dimulai jika jumlah pembuatan saat ini kurang dari atau sama dengan batas ini. Jika jumlah pembuatan saat ini memenuhi batas ini, pembuatan baru ditutup dan tidak dijalankan.

Informasi tambahan

(Opsional) Untuk Tag, masukkan nama dan nilai tag apa pun yang Anda inginkan untuk digunakan oleh AWS layanan pendukung. Gunakan Tambah baris untuk menambahkan tag. Anda dapat menambahkan hingga 50 tag.

Sumber

Penyedia sumber

Pilih jenis penyedia kode sumber. Gunakan daftar berikut untuk membuat pilihan yang sesuai untuk penyedia sumber Anda:

> **Note**
>
> CodeBuild tidak mendukung Bitbucket Server.

Amazon S3

Bucket

Pilih nama bucket input yang berisi kode sumber.

Kunci objek S3 atau folder S3

Masukkan nama file ZIP atau path ke folder yang berisi kode sumber. Masukkan garis miring (/) untuk mengunduh semua yang ada di bucket S3.

Versi sumber

Masukkan ID versi objek yang mewakili build file input Anda. Untuk informasi selengkapnya, lihatContoh versi sumber dengan AWS CodeBuild.

CodeCommit

Repositori

Pilih repositori yang ingin Anda gunakan.

Jenis referensi

Pilih Cabang, tag Git, atau ID Komit untuk menentukan versi kode sumber Anda. Untuk informasi selengkapnya, lihat Contoh versi sumber dengan AWS CodeBuild.

> **Note**
>
> Kami menyarankan Anda memilih nama cabang Git yang tidak terlihat seperti ID komit, seperti `811dd1ba1aba14473856cee38308caed7190c0d` atau`5392f7`. Ini membantu Anda menghindari tabrakan checkout Git dengan commit yang sebenarnya.

Kedalaman klon Git

Pilih untuk membuat klon dangkal dengan riwayat terpotong ke jumlah komit yang ditentukan. Jika Anda ingin klon penuh, pilih Penuh.

Submodul Git

Pilih Gunakan submodul Git jika Anda ingin menyertakan submodul Git dalam repositori Anda.

Bitbucket

Kredensi

Pilih Default source credential atau Custom source credential dan ikuti petunjuk untuk mengelola kredensi sumber default atau menyesuaikan kredensi sumber.

Jenis koneksi

Pilih AWS CodeConnections, OAuth, Kata sandi aplikasi, atau Token akses pribadi untuk terhubung. CodeBuild

Koneksi

Pilih koneksi Bitbucket atau rahasia Secrets Manager untuk terhubung melalui jenis koneksi yang Anda tentukan.

Repositori

Pilih Repositori di akun Bitbucket saya atau repositori Publik dan masukkan URL repositori.

Versi sumber

Masukkan cabang, ID komit, tag, atau referensi dan ID komit. Untuk informasi selengkapnya, lihat Contoh versi sumber dengan AWS CodeBuild

> **Note**
>
> Kami menyarankan Anda memilih nama cabang Git yang tidak terlihat seperti ID komit, seperti `811dd1ba1aba14473856cee38308caed7190c0d` atau`5392f7`. Ini membantu Anda menghindari tabrakan checkout Git dengan commit yang sebenarnya.

Kedalaman klon Git

Pilih kedalaman klon Git untuk membuat klon dangkal dengan riwayat terpotong ke jumlah komit yang ditentukan. Jika Anda ingin klon penuh, pilih Penuh.

Submodul Git

Pilih Gunakan submodul Git jika Anda ingin menyertakan submodul Git dalam repositori Anda.

Membangun status

Pilih Laporkan status build ke penyedia sumber saat build dimulai dan selesai jika Anda ingin status awal dan penyelesaian build dilaporkan ke penyedia sumber.

Untuk dapat melaporkan status build ke penyedia sumber, pengguna yang terkait dengan penyedia sumber harus memiliki akses tulis ke repo. Jika pengguna tidak memiliki akses tulis, status build tidak dapat diperbarui. Untuk informasi selengkapnya, lihat Akses penyedia sumber.

Untuk konteks Status, masukkan nilai yang akan digunakan untuk `name` parameter dalam status komit Bitbucket. Untuk informasi selengkapnya, lihat build dalam dokumentasi Bitbucket API.

Untuk URL Target, masukkan nilai yang akan digunakan untuk `url` parameter dalam status komit Bitbucket. Untuk informasi selengkapnya, lihat build dalam dokumentasi Bitbucket API.

Status build yang dipicu oleh webhook selalu dilaporkan ke penyedia sumber. Agar status build yang dimulai dari konsol atau panggilan API dilaporkan ke penyedia sumber, Anda harus memilih setelan ini.

Jika build proyek Anda dipicu oleh webhook, Anda harus mendorong komit baru ke repo agar perubahan pada pengaturan ini diterapkan.

Dalam peristiwa webhook sumber utama, pilih Rebuild setiap kali perubahan kode didorong ke repositori ini jika Anda CodeBuild ingin membangun kode sumber setiap kali perubahan kode didorong ke repositori ini. Untuk informasi selengkapnya tentang webhook dan grup filter, lihat. Acara webhook Bitbucket

GitHub

Kredensi

Pilih Default source credential atau Custom source credential dan ikuti petunjuk untuk mengelola kredensi sumber default atau menyesuaikan kredensi sumber.

Jenis koneksi

Pilih GitHub Aplikasi, OAuth, atau Token akses pribadi untuk terhubung. CodeBuild

Koneksi

Pilih GitHub koneksi atau rahasia Secrets Manager untuk terhubung melalui jenis koneksi yang Anda tentukan.

Repositori

Pilih Repositori di GitHub akun saya, Repositori publik, atau webhook GitHub tercakup dan masukkan URL repositori.

Versi sumber

Masukkan cabang, ID komit, tag, atau referensi dan ID komit. Untuk informasi selengkapnya, lihat Contoh versi sumber dengan AWS CodeBuild

> **Note**
>
> Kami menyarankan Anda memilih nama cabang Git yang tidak terlihat seperti ID komit, seperti `811dd1ba1aba14473856cee38308caed7190c0d` atau`5392f7`. Ini membantu Anda menghindari tabrakan checkout Git dengan commit yang sebenarnya.

Kedalaman klon Git

Pilih kedalaman klon Git untuk membuat klon dangkal dengan riwayat terpotong ke jumlah komit yang ditentukan. Jika Anda ingin klon penuh, pilih Penuh.

Submodul Git

Pilih Gunakan submodul Git jika Anda ingin menyertakan submodul Git dalam repositori Anda.

Membangun status

Pilih Laporkan status build ke penyedia sumber saat build dimulai dan selesai jika Anda ingin status awal dan penyelesaian build dilaporkan ke penyedia sumber.

Untuk dapat melaporkan status build ke penyedia sumber, pengguna yang terkait dengan penyedia sumber harus memiliki akses tulis ke repo. Jika pengguna tidak memiliki akses tulis, status build tidak dapat diperbarui. Untuk informasi selengkapnya, lihat Akses penyedia sumber.

Untuk konteks Status, masukkan nilai yang akan digunakan untuk `context` parameter dalam status GitHub komit. Untuk informasi selengkapnya, lihat Membuat status komit di panduan GitHub pengembang.

Untuk URL Target, masukkan nilai yang akan digunakan untuk `target_url` parameter dalam status GitHub komit. Untuk informasi selengkapnya, lihat Membuat status komit di panduan GitHub pengembang.

Status build yang dipicu oleh webhook selalu dilaporkan ke penyedia sumber. Agar status build yang dimulai dari konsol atau panggilan API dilaporkan ke penyedia sumber, Anda harus memilih setelan ini.

Jika build proyek Anda dipicu oleh webhook, Anda harus mendorong komit baru ke repo agar perubahan pada pengaturan ini diterapkan.

Dalam peristiwa webhook sumber utama, pilih Rebuild setiap kali perubahan kode didorong ke repositori ini jika Anda CodeBuild ingin membangun kode sumber setiap kali perubahan kode didorong ke repositori ini. Untuk informasi selengkapnya tentang webhook dan grup filter, lihat. GitHub acara webhook

GitHub Enterprise Server

Kredensi

Pilih Default source credential atau Custom source credential dan ikuti petunjuk untuk mengelola kredensi sumber default atau menyesuaikan kredensi sumber.

Jenis koneksi

Pilih AWS CodeConnectionsatau Token akses pribadi untuk terhubung CodeBuild.

Koneksi

Pilih koneksi GitHub Enterprise atau rahasia Secrets Manager untuk terhubung melalui jenis koneksi yang Anda tentukan.

Repositori

Pilih Repositori di akun GitHub Enterprise saya atau webhook cakupan GitHub Enterprise dan masukkan URL repositori.

Versi sumber

Masukkan permintaan tarik, cabang, ID komit, tag, atau referensi, dan ID komit. Untuk informasi selengkapnya, lihat Contoh versi sumber dengan AWS CodeBuild.

> **Note**
>
> Kami menyarankan Anda memilih nama cabang Git yang tidak terlihat seperti ID komit, seperti `811dd1ba1aba14473856cee38308caed7190c0d` atau`5392f7`. Ini membantu Anda menghindari tabrakan checkout Git dengan commit yang sebenarnya.

Kedalaman klon Git

Pilih kedalaman klon Git untuk membuat klon dangkal dengan riwayat terpotong ke jumlah komit yang ditentukan. Jika Anda ingin klon penuh, pilih Penuh.

Submodul Git

Pilih Gunakan submodul Git jika Anda ingin menyertakan submodul Git dalam repositori Anda.

Membangun status

Pilih Laporkan status build ke penyedia sumber saat build dimulai dan selesai jika Anda ingin status awal dan penyelesaian build dilaporkan ke penyedia sumber.

Untuk dapat melaporkan status build ke penyedia sumber, pengguna yang terkait dengan penyedia sumber harus memiliki akses tulis ke repo. Jika pengguna tidak memiliki akses tulis, status build tidak dapat diperbarui. Untuk informasi selengkapnya, lihat Akses penyedia sumber.

Untuk konteks Status, masukkan nilai yang akan digunakan untuk `context` parameter dalam status GitHub komit. Untuk informasi selengkapnya, lihat Membuat status komit di panduan GitHub pengembang.

Untuk URL Target, masukkan nilai yang akan digunakan untuk `target_url` parameter dalam status GitHub komit. Untuk informasi selengkapnya, lihat Membuat status komit di panduan GitHub pengembang.

Status build yang dipicu oleh webhook selalu dilaporkan ke penyedia sumber. Agar status build yang dimulai dari konsol atau panggilan API dilaporkan ke penyedia sumber, Anda harus memilih setelan ini.

Jika build proyek Anda dipicu oleh webhook, Anda harus mendorong komit baru ke repo agar perubahan pada pengaturan ini diterapkan.

SSL tidak aman

Pilih Aktifkan SSL yang tidak aman untuk mengabaikan peringatan SSL saat menghubungkan ke repositori proyek Enterprise Anda GitHub .

Dalam peristiwa webhook sumber utama, pilih Rebuild setiap kali perubahan kode didorong ke repositori ini jika Anda CodeBuild ingin membangun kode sumber setiap kali perubahan kode

didorong ke repositori ini. Untuk informasi selengkapnya tentang webhook dan grup filter, lihat. GitHub acara webhook

GitLab

Kredensi

Pilih Default source credential atau Custom source credential dan ikuti petunjuk untuk mengelola kredensi sumber default atau menyesuaikan kredensi sumber.

Jenis koneksi

AWS CodeConnectionsdigunakan untuk terhubung GitLab ke CodeBuild.

Koneksi

Pilih GitLab koneksi untuk terhubung AWS CodeConnections.

Repositori

Pilih repositori yang ingin Anda gunakan.

Versi sumber

Masukkan ID permintaan tarik, cabang, ID komit, tag, atau referensi, dan ID komit. Untuk informasi selengkapnya, lihat Contoh versi sumber dengan AWS CodeBuild.

> **Note**
>
> Kami menyarankan Anda memilih nama cabang Git yang tidak terlihat seperti ID komit, seperti `811dd1ba1aba14473856cee38308caed7190c0d` atau`5392f7`. Ini membantu Anda menghindari tabrakan checkout Git dengan commit yang sebenarnya.

Kedalaman klon Git

Pilih kedalaman klon Git untuk membuat klon dangkal dengan riwayat terpotong ke jumlah komit yang ditentukan. Jika Anda ingin klon penuh, pilih Penuh.

Membangun status

Pilih Laporkan status build ke penyedia sumber saat build dimulai dan selesai jika Anda ingin status awal dan penyelesaian build dilaporkan ke penyedia sumber.

Untuk dapat melaporkan status build ke penyedia sumber, pengguna yang terkait dengan penyedia sumber harus memiliki akses tulis ke repo. Jika pengguna tidak memiliki akses

tulis, status build tidak dapat diperbarui. Untuk informasi selengkapnya, lihat Akses penyedia sumber.

GitLab Self Managed

Kredensi

Pilih Default source credential atau Custom source credential dan ikuti petunjuk untuk mengelola kredensi sumber default atau menyesuaikan kredensi sumber.

Jenis koneksi

AWS CodeConnectionsdigunakan untuk menghubungkan GitLab Self Managed ke CodeBuild.

Koneksi

Pilih koneksi GitLab Self Managed untuk terhubung AWS CodeConnections.

Repositori

Pilih repositori yang ingin Anda gunakan.

Versi sumber

Masukkan ID permintaan tarik, cabang, ID komit, tag, atau referensi, dan ID komit. Untuk informasi selengkapnya, lihat Contoh versi sumber dengan AWS CodeBuild.

> **Note**
>
> Kami menyarankan Anda memilih nama cabang Git yang tidak terlihat seperti ID komit, seperti `811dd1ba1aba14473856cee38308caed7190c0d` atau`5392f7`. Ini membantu Anda menghindari tabrakan checkout Git dengan commit yang sebenarnya.

Kedalaman klon Git

Pilih kedalaman klon Git untuk membuat klon dangkal dengan riwayat terpotong ke jumlah komit yang ditentukan. Jika Anda ingin klon penuh, pilih Penuh.

Membangun status

Pilih Laporkan status build ke penyedia sumber saat build dimulai dan selesai jika Anda ingin status awal dan penyelesaian build dilaporkan ke penyedia sumber.

Untuk dapat melaporkan status build ke penyedia sumber, pengguna yang terkait dengan penyedia sumber harus memiliki akses tulis ke repo. Jika pengguna tidak memiliki akses tulis, status build tidak dapat diperbarui. Untuk informasi selengkapnya, lihat Akses penyedia sumber.

Lingkungan

Model penyediaan

Lakukan salah satu tindakan berikut:

- Untuk menggunakan armada sesuai permintaan yang dikelola oleh AWS CodeBuild, pilih. On-demand Dengan armada sesuai permintaan, CodeBuild menyediakan komputasi untuk build Anda. Mesin-mesin dihancurkan saat build selesai. On-demand armada dikelola sepenuhnya, dan mencakup kemampuan penskalaan otomatis untuk menangani lonjakan permintaan.
- Untuk menggunakan armada kapasitas cadangan yang dikelola oleh AWS CodeBuild, pilih Kapasitas cadangan, lalu pilih nama Armada. Dengan armada kapasitas cadangan, Anda mengonfigurasi satu set instans khusus untuk lingkungan build Anda. Mesin ini tetap menganggur, siap untuk memproses build atau pengujian segera, dan mengurangi durasi pembuatan. Dengan armada kapasitas cadangan, mesin Anda selalu berjalan dan akan terus mengeluarkan biaya selama disediakan.

Untuk informasi, lihat Jalankan build pada armada kapasitas yang dicadangkan.

Citra lingkungan

Lakukan salah satu tindakan berikut:

- Untuk menggunakan image Docker yang dikelola oleh AWS CodeBuild, pilih Managed image, lalu buat pilihan dari Sistem operasi, Runtime, Image, dan versi Image. Buat pilihan dari jenis Lingkungan jika tersedia.
- Untuk menggunakan gambar Docker lain, pilih Custom image. Untuk jenis Lingkungan, pilih ARM, Linux, GPU Linux, atau Windows. Jika Anda memilih Registri lain, untuk URL registri eksternal, masukkan nama dan tag gambar Docker di Docker Hub, menggunakan format. *docker repository*/*docker image name* Jika Anda memilih Amazon ECR, gunakan repositori Amazon ECR dan gambar Amazon ECR untuk memilih gambar Docker di akun Anda. AWS
- Untuk menggunakan image Docker pribadi, pilih Custom image. Untuk jenis Lingkungan, pilih ARM, Linux, GPU Linux, atau Windows. Untuk registri Gambar, pilih Registri lain, lalu

masukkan ARN kredensil untuk gambar Docker pribadi Anda. Kredensialnya harus dibuat oleh Secrets Manager. Untuk informasi lebih lanjut, lihat Apa itu AWS Secrets Manager? di Panduan Pengguna AWS Secrets Manager .

> **Note**
>
> CodeBuild mengganti `ENTRYPOINT` untuk gambar Docker khusus.

Hitung

Lakukan salah satu tindakan berikut:

- Untuk menggunakan komputasi EC2, pilih EC2. Komputasi EC2 menawarkan fleksibilitas yang dioptimalkan selama aksi berjalan.
- Untuk menggunakan komputasi Lambda, pilih Lambda. Lambda compute menawarkan kecepatan start-up yang dioptimalkan untuk build Anda. Lambda mendukung build yang lebih cepat karena latensi start-up yang lebih rendah. Lambda juga secara otomatis menskalakan, jadi build tidak menunggu dalam antrian untuk dijalankan. Untuk informasi selengkapnya, lihat Jalankan dibangun di AWS Lambda komputasi.

Peran layanan

Lakukan salah satu tindakan berikut:

- Jika Anda tidak memiliki peran CodeBuild layanan, pilih Peran layanan baru. Di Nama peran, masukkan nama untuk peran baru.
- Jika Anda memiliki peran CodeBuild layanan, pilih Peran layanan yang ada. Di Peran ARN, pilih peran layanan.

> **Note**
>
> Saat menggunakan konsol untuk membuat proyek build, Anda dapat membuat peran CodeBuild layanan secara bersamaan. Secara default, peran hanya berfungsi dengan proyek build tersebut. Jika Anda menggunakan konsol untuk mengaitkan peran layanan ini dengan proyek build lain, peran akan diperbarui agar berfungsi dengan proyek build lainnya. Peran layanan dapat bekerja dengan hingga 10 proyek pembangunan.

Konfigurasi tambahan

Auto-retry batasi

Tentukan jumlah percobaan ulang otomatis tambahan setelah build gagal. Misalnya, jika batas coba ulang otomatis disetel ke 2, CodeBuild akan memanggil `RetryBuild` API untuk mencoba ulang build Anda secara otomatis hingga 2 kali tambahan.

Batas waktu

Tentukan nilai, antara 5 menit dan 36 jam, setelah itu CodeBuild menghentikan build jika tidak selesai. Jika jam dan menit dibiarkan kosong, nilai default 60 menit digunakan.

Diistimewakan

(Opsional) Pilih Aktifkan bendera ini jika Anda ingin membuat gambar Docker atau ingin build Anda mendapatkan hak istimewa yang ditingkatkan hanya jika Anda berencana menggunakan proyek build ini untuk membuat gambar Docker. Jika tidak, semua build terkait yang mencoba berinteraksi dengan daemon Docker gagal. Anda juga harus memulai daemon Docker sehingga build Anda dapat berinteraksi dengannya. Salah satu cara untuk melakukannya adalah dengan menginisialisasi daemon Docker di `install` fase buildspec Anda dengan menjalankan perintah build berikut. Jangan jalankan perintah ini jika Anda memilih image lingkungan build yang disediakan oleh CodeBuild dengan dukungan Docker.

> **Note**
>
> Secara default, daemon Docker diaktifkan untuk build non-VPC. Jika Anda ingin menggunakan kontainer Docker untuk build VPC, lihat Runtime Privilege dan Kemampuan Linux di situs web Docker Docs dan aktifkan mode istimewa. Juga, Windows tidak mendukung mode istimewa.

```
- nohup /usr/local/bin/dockerd --host=unix:///var/run/docker.sock --
host=tcp://127.0.0.1:2375 --storage-driver=overlay2 &
- timeout 15 sh -c "until docker info; do echo .; sleep 1; done"
```

VPC

Jika Anda CodeBuild ingin bekerja dengan VPC Anda:

- Untuk VPC, pilih ID VPC yang digunakan. CodeBuild

- Untuk Subnet VPC, pilih subnet yang menyertakan sumber daya yang digunakan. CodeBuild
- Untuk grup Keamanan VPC, pilih grup keamanan yang CodeBuild digunakan untuk memungkinkan akses ke sumber daya di VPC.

Untuk informasi selengkapnya, lihat Gunakan AWS CodeBuild dengan Amazon Virtual Private Cloud.

Hitung

Pilih salah satu opsi yang tersedia.

Kredensi registri

Tentukan kredensi registri saat proyek dikonfigurasi dengan gambar registri non-pribadi.

> **Note**
>
> Kredensyal ini hanya akan digunakan jika gambar diganti dengan yang berasal dari pendaftar pribadi.

Variabel lingkungan

Masukkan nama dan nilai, lalu pilih jenis setiap variabel lingkungan untuk build yang akan digunakan.

> **Note**
>
> CodeBuild menetapkan variabel lingkungan untuk AWS Wilayah Anda secara otomatis. Anda harus menyetel variabel lingkungan berikut jika Anda belum menambahkannya ke buildspec.ymlmu:
>
> - AWS_ACCOUNT_ID
> - IMAGE_REPO_NAME
> - IMAGE_TAG

Konsol dan AWS CLI pengguna dapat melihat variabel lingkungan. Jika Anda tidak memiliki kekhawatiran tentang visibilitas variabel lingkungan Anda, atur bidang Nama dan Nilai, lalu setel Type ke Plaintext.

Kami menyarankan Anda menyimpan variabel lingkungan dengan nilai sensitif, seperti ID kunci AWS akses, kunci akses AWS rahasia, atau kata sandi sebagai parameter di Amazon EC2 Systems Manager Parameter Store AWS Secrets Manager atau.

Jika Anda menggunakan Amazon EC2 Systems Manager Parameter Store, lalu untuk Type, pilih Parameter. Untuk Nama, masukkan pengenal CodeBuild untuk referensi. Untuk Nilai, masukkan nama parameter seperti yang disimpan di Amazon EC2 Systems Manager Parameter Store. Menggunakan parameter bernama `/CodeBuild/dockerLoginPassword` sebagai contoh, untuk Type, pilih Parameter. Untuk Nama, masukkan `LOGIN_PASSWORD`. Untuk Nilai, masukkan `/CodeBuild/dockerLoginPassword`.

> **Important**
>
> Jika Anda menggunakan Amazon EC2 Systems Manager Parameter Store, sebaiknya Anda menyimpan parameter dengan nama parameter yang dimulai `/CodeBuild/` dengan (misalnya`/CodeBuild/dockerLoginPassword`,). Anda dapat menggunakan CodeBuild konsol untuk membuat parameter di Amazon EC2 Systems Manager. Pilih Buat parameter, lalu ikuti instruksi di kotak dialog. (Di kotak dialog itu, untuk kunci KMS, Anda dapat menentukan ARN kunci di AWS KMS akun Anda. Amazon EC2 Systems Manager menggunakan kunci ini untuk mengenkripsi nilai parameter selama penyimpanan dan mendekripsi selama pengambilan.) Jika Anda menggunakan CodeBuild konsol untuk membuat parameter, konsol memulai nama parameter dengan `/CodeBuild/` saat sedang disimpan. Untuk informasi selengkapnya, lihat Systems Manager Parameter Store dan Systems Manager Parameter Store Console Walkthrough di Panduan Pengguna Amazon EC2 Systems Manager.
>
> Jika project build Anda mengacu pada parameter yang disimpan di Amazon EC2 Systems Manager Parameter Store, peran layanan project build harus mengizinkan tindakan `ssm:GetParameters` tersebut. Jika Anda memilih Peran layanan baru lebih awal, CodeBuild sertakan tindakan ini dalam peran layanan default untuk proyek build Anda. Namun, jika Anda memilih Peran layanan yang ada, Anda harus menyertakan tindakan ini ke peran layanan Anda secara terpisah.
>
> Jika project build Anda merujuk ke parameter yang disimpan di Amazon EC2 Systems Manager Parameter Store dengan nama parameter yang tidak dimulai`/CodeBuild/`, dan Anda memilih Peran layanan baru, Anda harus memperbarui peran layanan tersebut untuk mengizinkan akses ke nama parameter yang tidak dimulai/

`CodeBuild/`. Ini karena peran layanan itu memungkinkan akses hanya ke nama parameter yang dimulai dengan`/CodeBuild/`.
Jika Anda memilih Peran layanan baru, peran layanan menyertakan izin untuk mendekripsi semua parameter di bawah `/CodeBuild/` namespace di Amazon EC2 Systems Manager Parameter Store.
Variabel lingkungan yang Anda tetapkan menggantikan variabel lingkungan yang ada. Misalnya, jika image Docker sudah berisi variabel lingkungan bernama `MY_VAR` dengan nilai`my_value`, dan Anda menetapkan variabel lingkungan bernama `MY_VAR` dengan nilai`other_value`, `my_value` maka diganti dengan`other_value`. Demikian pula, jika image Docker sudah berisi variabel lingkungan bernama `PATH` dengan nilai`/usr/local/sbin:/usr/local/bin`, dan Anda menetapkan variabel lingkungan bernama `PATH` dengan nilai`$PATH:/usr/share/ant/bin`, `/usr/local/sbin:/usr/local/bin` maka digantikan oleh nilai `$PATH:/usr/share/ant/bin` literal.
Jangan mengatur variabel lingkungan dengan nama yang dimulai dengan`CODEBUILD_`. Awalan ini dicadangkan untuk penggunaan internal .
Jika variabel lingkungan dengan nama yang sama didefinisikan di beberapa tempat, nilainya ditentukan sebagai berikut:

- Nilai dalam panggilan operasi start build lebih diutamakan.
- Nilai dalam definisi proyek build akan diutamakan berikutnya.
- Nilai dalam deklarasi buildspec diutamakan paling rendah.

Jika Anda menggunakan Secrets Manager, untuk Type, pilih Secrets Manager. Untuk Nama, masukkan pengenal CodeBuild untuk referensi. Untuk Nilai, masukkan `reference-key` menggunakan pola*`secret-id`*`:`*`json-key`*`:`*`version-stage`*`:`*`version-id`*. Untuk informasi, lihat Secrets Manager reference-key in the buildspec file.

Important

Jika Anda menggunakan Secrets Manager, kami sarankan Anda menyimpan rahasia dengan nama yang dimulai dengan `/CodeBuild/` (misalnya,`/CodeBuild/dockerLoginPassword`). Untuk informasi lebih lanjut, lihat Apa itu AWS Secrets Manager? di Panduan Pengguna AWS Secrets Manager .
Jika proyek build Anda mengacu pada rahasia yang disimpan di Secrets Manager, peran layanan proyek build harus mengizinkan `secretsmanager:GetSecretValue` tindakan tersebut. Jika Anda memilih Peran layanan baru lebih awal, CodeBuild

sertakan tindakan ini dalam peran layanan default untuk proyek build Anda. Namun, jika Anda memilih Peran layanan yang ada, Anda harus menyertakan tindakan ini ke peran layanan Anda secara terpisah.
Jika proyek build mengacu pada rahasia yang disimpan di Secrets Manager dengan nama rahasia yang tidak dimulai`/CodeBuild/`, dan Anda memilih Peran layanan baru, Anda harus memperbarui peran layanan untuk mengizinkan akses ke nama rahasia yang tidak dimulai`/CodeBuild/`. Ini karena peran layanan memungkinkan akses hanya ke nama rahasia yang dimulai dengan`/CodeBuild/`.
Jika Anda memilih Peran layanan baru, peran layanan menyertakan izin untuk mendekripsi semua rahasia di bawah `/CodeBuild/` namespace di Secrets Manager.

Buildspec

Membangun spesifikasi

Lakukan salah satu tindakan berikut:

- Jika kode sumber Anda menyertakan file buildspec, pilih Gunakan file buildspec. Secara default, CodeBuild mencari file bernama `buildspec.yml` di direktori root kode sumber. Jika file buildspec Anda menggunakan nama atau lokasi yang berbeda, masukkan jalurnya dari root sumber dalam nama Buildspec (misalnya, atau. `buildspec-two.yml` `configuration/buildspec.yml` Jika file buildspec ada di bucket S3, file tersebut harus berada di AWS Region yang sama dengan project build Anda. Tentukan file buildspec menggunakan ARN (misalnya,). `arn:aws:s3:::`*`<my-codebuild-sample2>`*`/buildspec.yml`
- Jika kode sumber Anda tidak menyertakan file buildspec, atau jika Anda ingin menjalankan perintah build yang berbeda dari yang ditentukan untuk `build` fase dalam `buildspec.yml` file di direktori root kode sumber, pilih Sisipkan perintah build. Untuk perintah Build, masukkan perintah yang ingin Anda jalankan dalam `build` fase. Untuk beberapa perintah, pisahkan setiap perintah dengan && (misalnya,`mvn test && mvn package`). Untuk menjalankan perintah dalam fase lain, atau jika Anda memiliki daftar panjang perintah untuk `build` fase tersebut, tambahkan `buildspec.yml` file ke direktori root kode sumber, tambahkan perintah ke file, lalu pilih Gunakan buildspec.ymldi direktori root kode sumber.

Untuk informasi selengkapnya, lihat Referensi Buildspec.

Konfigurasi batch

Anda dapat menjalankan grup build sebagai satu operasi. Untuk informasi selengkapnya, lihat Jalankan build dalam batch.

Tentukan konfigurasi batch

Pilih untuk mengizinkan pembuatan batch dalam proyek ini.

Peran layanan Batch

Menyediakan peran layanan untuk build batch.

Pilih salah satu cara berikut:

- Jika Anda tidak memiliki peran layanan batch, pilih Peran layanan baru. Di peran Layanan, masukkan nama untuk peran baru.
- Jika Anda memiliki peran layanan batch, pilih Peran layanan yang ada. Dalam peran Layanan, pilih peran layanan.

Build batch memperkenalkan peran keamanan baru dalam konfigurasi batch. Peran baru ini diperlukan karena CodeBuild harus dapat memanggil`StartBuild`,`StopBuild`, dan `RetryBuild` tindakan atas nama Anda untuk menjalankan build sebagai bagian dari batch. Pelanggan harus menggunakan peran baru, dan bukan peran yang sama yang mereka gunakan dalam build mereka, karena dua alasan:

- Memberikan izin `StartBuild`, `StopBuild`, dan `RetryBuild` kepada peran build akan memungkinkan satu build untuk memulai lebih banyak build melalui buildspec.
- CodeBuild build batch memberikan batasan yang membatasi jumlah build dan tipe komputasi yang dapat digunakan untuk build dalam batch. Jika peran build memiliki izin ini, build dapat melewati batasan ini sendiri.

Jenis komputasi yang diizinkan untuk batch

Pilih jenis komputasi yang diizinkan untuk batch. Pilih semua yang berlaku.

Armada yang diizinkan untuk batch

Pilih armada yang diizinkan untuk batch. Pilih semua yang berlaku.

Build maksimum yang diizinkan dalam batch

Masukkan jumlah maksimum build yang diizinkan dalam batch. Jika batch melebihi batas ini, batch akan gagal.

Batas waktu Batch

Masukkan jumlah waktu maksimum untuk menyelesaikan batch build.

Gabungkan artefak

Pilih Gabungkan semua artefak dari batch ke satu lokasi agar semua artefak dari batch digabungkan menjadi satu lokasi.

Mode laporan Batch

Pilih mode laporan status build yang diinginkan untuk build batch.

> **Note**
>
> Bidang ini hanya tersedia jika sumber proyek adalah Bitbucket, GitHub, atau GitHub Enterprise, dan Laporkan status build ke penyedia sumber saat build Anda mulai dan selesai dipilih di bawah Sumber.

Build agregat

Pilih agar status untuk semua build dalam batch digabungkan menjadi satu laporan status.

Membangun individu

Pilih agar status build untuk semua build dalam batch dilaporkan secara terpisah.

Artifacts

Jenis

Lakukan salah satu tindakan berikut:

- Jika Anda tidak ingin membuat artefak keluaran build apa pun, pilih Tidak ada artefak. Anda mungkin ingin melakukan ini jika Anda hanya menjalankan pengujian build atau Anda ingin mendorong image Docker ke repositori Amazon ECR.
- Untuk menyimpan output build dalam bucket S3, pilih Amazon S3, lalu lakukan hal berikut:
  - Jika Anda ingin menggunakan nama proyek Anda untuk file atau folder ZIP keluaran build, biarkan Nama kosong. Jika tidak, masukkan nama. (Jika Anda ingin mengeluarkan file ZIP, dan Anda ingin file ZIP memiliki ekstensi file, pastikan untuk memasukkannya setelah nama file ZIP.)

- Pilih Aktifkan versi semantik jika Anda ingin nama yang ditentukan dalam file buildspec untuk mengganti nama apa pun yang ditentukan di konsol. Nama dalam file buildspec dihitung pada waktu pembuatan dan menggunakan bahasa perintah Shell. Misalnya, Anda dapat menambahkan tanggal dan waktu untuk nama artefak Anda sehingga selalu unik. Nama artefak unik mencegah artefak ditimpa. Untuk informasi selengkapnya, lihat Sintaks Buildspec.
- Untuk nama Bucket, pilih nama bucket keluaran.
- Jika Anda memilih Sisipkan perintah build sebelumnya dalam prosedur ini, maka untuk file Output, masukkan lokasi file dari build yang ingin Anda masukkan ke dalam file atau folder ZIP keluaran build. Untuk beberapa lokasi, pisahkan setiap lokasi dengan koma (misalnya,`appspec.yml, target/my-app.jar`). Untuk informasi lebih lanjut, lihat deskripsi `files` diSintaks Buildspec.
- Jika Anda tidak ingin artefak build Anda dienkripsi, pilih Hapus enkripsi artefak.

Untuk setiap set artefak sekunder yang Anda inginkan:

1. Untuk pengenal Artifact, masukkan nilai yang kurang dari 128 karakter dan hanya berisi karakter alfanumerik dan garis bawah.
2. Pilih Tambahkan artefak.
3. Ikuti langkah-langkah sebelumnya untuk mengonfigurasi artefak sekunder Anda.
4. Pilih Simpan artefak.

Konfigurasi tambahan

Kunci enkripsi

(Opsional) Lakukan salah satu langkah berikut:

- Untuk menggunakan Amazon S3 di akun Anda untuk mengenkripsi artefak keluaran build, biarkan kunci Enkripsi kosong. Kunci yang dikelola AWS Ini adalah opsi default.
- Untuk menggunakan kunci yang dikelola pelanggan untuk mengenkripsi artefak keluaran build, di kunci Enkripsi, masukkan ARN kunci KMS. Gunakan format `arn:aws:kms:`*`region-ID`*`:`*`account-ID`*`:key/`*`key-ID`*.

Jenis cache

Untuk jenis Cache, pilih salah satu dari berikut ini:

- Jika Anda tidak ingin menggunakan cache, pilih Tidak ada cache.

- Jika Anda ingin menggunakan cache Amazon S3, pilih Amazon S3, lalu lakukan hal berikut:
  - Untuk Bucket, pilih nama bucket S3 tempat cache disimpan.
  - (Opsional) Untuk awalan jalur Cache, masukkan awalan jalur Amazon S3. Nilai awalan jalur Cache mirip dengan nama direktori. Itu memungkinkan Anda untuk menyimpan cache di bawah direktori yang sama dalam ember.

    Important

    Jangan menambahkan garis miring (/) ke ujung awalan jalur.
- Jika Anda ingin menggunakan cache lokal, pilih Lokal, lalu pilih satu atau beberapa mode cache lokal.

  Note

  Mode cache lapisan Docker hanya tersedia untuk Linux. Jika Anda memilihnya, proyek Anda harus berjalan dalam mode istimewa.

Menggunakan cache menghemat banyak waktu pembuatan karena bagian lingkungan build yang dapat digunakan kembali disimpan dalam cache dan digunakan di seluruh build. Untuk informasi tentang menentukan cache dalam file buildspec, lihat. Sintaks Buildspec Untuk informasi lebih lanjut tentang caching, lihatCache dibangun untuk meningkatkan kinerja.

Beberapa catatan

Pilih log yang ingin Anda buat. Anda dapat membuat CloudWatch Log Amazon, log Amazon S3, atau keduanya.

CloudWatch

Jika Anda ingin CloudWatch log Amazon Log:

CloudWatch log

Pilih CloudWatch log.

Nama grup

Masukkan nama grup CloudWatch log Amazon Logs Anda.

Nama Streaming

Masukkan nama aliran CloudWatch log Amazon Logs Anda.

S3

Jika Anda ingin log Amazon S3:

Log S3

Pilih log S3.

Bucket

Pilih nama bucket S3 untuk log Anda.

Awalan jalur

Masukkan awalan untuk log Anda.

Nonaktifkan enkripsi log S3

Pilih jika Anda tidak ingin log S3 Anda dienkripsi.

Buat proyek build (AWS CLI)

Untuk informasi lebih lanjut tentang menggunakan AWS CLI with CodeBuild, lihatReferensi baris perintah.

Untuk membuat proyek CodeBuild build menggunakan AWS CLI, Anda membuat struktur JSON-formatted Project, mengisi struktur, dan memanggil `create-project`perintah untuk membuat proyek.

Buat file JSON

Buat file JSON kerangka dengan `create-project`perintah, menggunakan opsi: `--generate-cli-skeleton`

```
aws codebuild create-project --generate-cli-skeleton > <json-file>
```

Ini membuat file JSON dengan path dan nama file yang ditentukan oleh*<json-file>*.

Isi file JSON

Ubah data JSON sebagai berikut dan simpan hasil Anda.

```
{
  "name": "<project-name>",
  "description": "<description>",
  "source": {
    "type": "CODECOMMIT" | "CODEPIPELINE" | "GITHUB" | "GITHUB_ENTERPRISE" | "GITLAB" |
 "GITLAB_SELF_MANAGED" | "BITBUCKET" | "S3" | "NO_SOURCE",
    "location": "<source-location>",
    "gitCloneDepth": "<git-clone-depth>",
    "buildspec": "<buildspec>",
    "InsecureSsl": "<insecure-ssl>",
    "reportBuildStatus": "<report-build-status>",
    "buildStatusConfig": {
      "context": "<context>",
      "targetUrl": "<target-url>"
    },
    "gitSubmodulesConfig": {
      "fetchSubmodules": "<fetch-submodules>"
    },
    "auth": {
      "type": "<auth-type>",
      "resource": "<auth-resource>"
    },
    "sourceIdentifier": "<source-identifier>"
  },
  "secondarySources": [
    {
        "type": "CODECOMMIT" | "CODEPIPELINE" | "GITHUB" | "GITHUB_ENTERPRISE" |
 "GITLAB" | "GITLAB_SELF_MANAGED" | "BITBUCKET" | "S3" | "NO_SOURCE",
        "location": "<source-location>",
        "gitCloneDepth": "<git-clone-depth>",
        "buildspec": "<buildspec>",
        "InsecureSsl": "<insecure-ssl>",
        "reportBuildStatus": "<report-build-status>",
        "auth": {
          "type": "<auth-type>",
          "resource": "<auth-resource>"
        },
        "sourceIdentifier": "<source-identifier>"
    }
  ],
  "secondarySourceVersions": [
    {
      "sourceIdentifier": "<secondary-source-identifier>",
```

```
      "sourceVersion": "<secondary-source-version>"
    }
  ],
  "sourceVersion": "<source-version>",
  "artifacts": {
    "type": "CODEPIPELINE" | "S3" | "NO_ARTIFACTS",
    "location": "<artifacts-location>",
    "path": "<artifacts-path>",
    "namespaceType": "<artifacts-namespacetype>",
    "name": "<artifacts-name>",
    "overrideArtifactName": "<override-artifact-name>",
    "packaging": "<artifacts-packaging>"
  },
  "secondaryArtifacts": [
    {
      "type": "CODEPIPELINE" | "S3" | "NO_ARTIFACTS",
      "location": "<secondary-artifact-location>",
      "path": "<secondary-artifact-path>",
      "namespaceType": "<secondary-artifact-namespaceType>",
      "name": "<secondary-artifact-name>",
      "packaging": "<secondary-artifact-packaging>",
      "artifactIdentifier": "<secondary-artifact-identifier>"
    }
  ],
  "cache": {
    "type": "<cache-type>",
    "location": "<cache-location>",
    "mode": [
      "<cache-mode>"
    ]
  },
  "environment": {
    "type": "LINUX_CONTAINER" | "LINUX_GPU_CONTAINER" | "ARM_CONTAINER" |
"WINDOWS_SERVER_2019_CONTAINER" | "WINDOWS_SERVER_2022_CONTAINER",
    "image": "<image>",
    "computeType": "BUILD_GENERAL1_SMALL" | "BUILD_GENERAL1_MEDIUM" |
"BUILD_GENERAL1_LARGE" | "BUILD_GENERAL1_2XLARGE",
    "certificate": "<certificate>",
    "environmentVariables": [
      {
        "name": "<environmentVariable-name>",
        "value": "<environmentVariable-value>",
        "type": "<environmentVariable-type>"
      }
```

```
    ],
    "registryCredential": [
      {
        "credential": "<credential-arn-or-name>",
        "credentialProvider": "<credential-provider>"
      }
    ],
    "imagePullCredentialsType": "CODEBUILD" | "SERVICE_ROLE",
    "privilegedMode": "<privileged-mode>"
  },
  "serviceRole": "<service-role>",
  "autoRetryLimit": <auto-retry-limit>,
  "timeoutInMinutes": <timeout>,
  "queuedTimeoutInMinutes": <queued-timeout>,
  "encryptionKey": "<encryption-key>",
  "tags": [
    {
      "key": "<tag-key>",
      "value": "<tag-value>"
    }
  ],
  "vpcConfig": {
    "securityGroupIds": [
        "<security-group-id>"
    ],
    "subnets": [
        "<subnet-id>"
    ],
    "vpcId": "<vpc-id>"
  },
  "badgeEnabled": "<badge-enabled>",
  "logsConfig": {
    "cloudWatchLogs": {
      "status": "<cloudwatch-logs-status>",
      "groupName": "<group-name>",
      "streamName": "<stream-name>"
    },
    "s3Logs": {
      "status": "<s3-logs-status>",
      "location": "<s3-logs-location>",
      "encryptionDisabled": "<s3-logs-encryption-disabled>"
    }
  },
  "fileSystemLocations": [
```

```
    {
      "type": "EFS",
      "location": "<EFS-DNS-name-1>:/<directory-path>",
      "mountPoint": "<mount-point>",
      "identifier": "<efs-identifier>",
      "mountOptions": "<efs-mount-options>"
    }
  ],
  "buildBatchConfig": {
    "serviceRole": "<batch-service-role>",
    "combineArtifacts": <combine-artifacts>,
    "restrictions": {
      "maximumBuildsAllowed": <max-builds>,
      "computeTypesAllowed": [
        "<compute-type>"
      ],
      "fleetsAllowed": [
        "<fleet-name>"
      ]
    },
    "timeoutInMins": <batch-timeout>,
    "batchReportMode": "REPORT_AGGREGATED_BATCH" | "REPORT_INDIVIDUAL_BUILDS"
  },
  "concurrentBuildLimit": <concurrent-build-limit>
}
```

Ganti yang berikut ini:

name

Wajib. Nama untuk proyek pembangunan ini. Nama ini harus unik di semua proyek build di AWS akun Anda.

deskripsi

Tidak wajib. Deskripsi untuk proyek pembangunan ini.

sumber

Wajib. ProjectSourceObjek yang berisi informasi tentang pengaturan kode sumber proyek build ini. Setelah Anda menambahkan `source` objek, Anda dapat menambahkan hingga 12 sumber lagi menggunakan file. Pengaturan ini meliputi yang berikut:

sumber/jenis

Wajib. Jenis repositori yang berisi kode sumber untuk membangun. Nilai yang valid meliputi:

- `CODECOMMIT`
- `CODEPIPELINE`
- `GITHUB`
- `GITHUB_ENTERPRISE`
- `GITLAB`
- `GITLAB_SELF_MANAGED`
- `BITBUCKET`
- `S3`
- `NO_SOURCE`

Jika Anda menggunakan`NO_SOURCE`, buildspec tidak dapat berupa file karena proyek tidak memiliki sumber. Sebagai gantinya, Anda harus menggunakan `buildspec` atribut untuk menentukan YAML-formatted string untuk buildspec Anda. Untuk informasi selengkapnya, lihat Membuat proyek build tanpa sumber.

sumber/lokasi

Diperlukan kecuali Anda mengatur *<source-type>* ke`CODEPIPELINE`. Lokasi kode sumber untuk jenis repositori yang ditentukan.

- Untuk CodeCommit, URL kloning HTTPS ke repositori yang berisi kode sumber dan file buildspec (misalnya,). `https://git-codecommit.`*<region-id>*`.amazonaws.com/v1/repos/`*<repo-name>*
- Untuk Amazon S3, nama bucket input build, diikuti dengan path dan nama file ZIP yang berisi kode sumber dan buildspec. Contoh:
  - Untuk file ZIP yang terletak di root bucket input:*<bucket-name>*`/`*<object-name>*`.zip`.
  - Untuk file ZIP yang terletak di subfolder di bucket input:*<bucket-name>*`/`*<subfoler-path>*`/`*<object-name>*`.zip`.
- Untuk GitHub, URL kloning HTTPS ke repositori yang berisi kode sumber dan file buildspec. URL harus berisi github.com. Anda harus menghubungkan AWS akun Anda ke GitHub akun Anda. Untuk melakukan ini, gunakan CodeBuild konsol untuk membuat proyek build.
  - Pilih Otorisasi aplikasi. (Setelah Anda terhubung ke GitHub akun Anda, Anda tidak perlu menyelesaikan pembuatan proyek build. Anda dapat menutup CodeBuild konsol.)

- Untuk GitHub Enterprise Server, HTTP atau HTTPS mengkloning URL ke repositori yang berisi kode sumber dan file buildspec. Anda juga harus menghubungkan AWS akun Anda ke akun GitHub Enterprise Server Anda. Untuk melakukan ini, gunakan CodeBuild konsol untuk membuat proyek build.

  1. Buat token akses pribadi di Server GitHub Perusahaan.
  2. Salin token ini ke clipboard Anda sehingga Anda dapat menggunakannya saat membuat proyek. CodeBuild Untuk informasi selengkapnya, lihat Membuat token akses pribadi untuk baris perintah di situs web GitHub Bantuan.
  3. Saat Anda menggunakan konsol untuk membuat CodeBuild proyek, di Sumber, untuk penyedia Sumber, pilih GitHubEnterprise.
  4. Untuk Token Akses Pribadi, tempel token yang disalin ke clipboard Anda. Pilih Simpan Token. CodeBuild Akun Anda sekarang terhubung ke akun GitHub Enterprise Server Anda.

- Untuk GitLab dan GitLab dikelola sendiri, HTTPS mengkloning URL ke repositori yang berisi kode sumber dan file buildspec. Perhatikan bahwa jika Anda menggunakan GitLab, URL harus berisi gitlab.com. Jika Anda menggunakan GitLab self-managed, URL tidak perlu berisi gitlab.com. Anda harus menghubungkan AWS akun Anda ke akun Anda GitLab atau yang GitLab dikelola sendiri. Untuk melakukan ini, gunakan CodeBuild konsol untuk membuat proyek build.

  - Di panel navigasi Alat Pengembang, pilih Pengaturan, Koneksi, lalu Buat koneksi. Di halaman ini, buat koneksi yang GitLab dikelola sendiri GitLab atau yang dikelola sendiri, lalu pilih Connect to GitLab.

- Untuk Bitbucket, URL kloning HTTPS ke repositori yang berisi kode sumber dan file buildspec. URL harus berisi bitbucket.org. Anda juga harus menghubungkan AWS akun Anda ke akun Bitbucket Anda. Untuk melakukan ini, gunakan CodeBuild konsol untuk membuat proyek build.

  1. Ketika Anda menggunakan konsol untuk menghubungkan (atau menghubungkan kembali) dengan Bitbucket, pada halaman Mengonfirmasi akses ke akun Anda di Bitbucket, pilih Berikan akses. (Setelah Anda terhubung ke akun Bitbucket Anda, Anda tidak perlu menyelesaikan pembuatan proyek build. Anda dapat menutup CodeBuild konsol.)

- Untuk AWS CodePipeline, jangan tentukan `location` nilai untuk`source`. CodePipeline mengabaikan nilai ini karena saat Anda membuat pipeline CodePipeline, Anda menentukan lokasi kode sumber di tahap Sumber pipeline.

sumber/git CloneDepth

Tidak wajib. Kedalaman riwayat yang akan diunduh. Nilai minimum adalah 0. Jika nilai ini adalah 0, lebih besar dari 25, atau tidak disediakan, maka riwayat penuh diunduh dengan setiap proyek build. Jika jenis sumber Anda adalah Amazon S3, nilai ini tidak didukung.

sumber/buildspec

Tidak wajib. Definisi spesifikasi build atau file yang akan digunakan. Jika nilai ini tidak disediakan atau diatur ke string kosong, kode sumber harus berisi `buildspec.yml` file dalam direktori root nya. Jika nilai ini disetel, itu bisa berupa definisi buildspec sebaris, jalur ke file buildspec alternatif relatif terhadap direktori root sumber utama Anda, atau jalur ke bucket S3. Bucket harus berada di AWS Wilayah yang sama dengan proyek pembangunan. Tentukan file buildspec menggunakan ARN (misalnya,). `arn:aws:s3:::<my-codebuild-sample2>/buildspec.yml` Untuk informasi selengkapnya, lihat Nama file buildspec dan lokasi penyimpanan.

sumber/autentikasi

Berisi informasi tentang pengaturan otorisasi CodeBuild untuk mengakses kode sumber yang akan dibangun.

source/auth/jenis

Wajib. Tipe otorisasi yang digunakan. Nilai yang valid adalah:

- `OAUTH`
- `CODECONNECTIONS`
- `SECRETS_MANAGER`

source/auth/sumber daya

Tidak wajib. Nilai sumber daya yang berlaku untuk jenis otorisasi yang ditentukan. Ini bisa berupa Secrets Manager ARN atau AWS CodeConnections ARN.

sumber/laporan BuildStatus

Menentukan apakah akan mengirim status awal dan penyelesaian build kepada penyedia sumber Anda. Jika Anda mengatur ini dengan penyedia sumber selain GitHub, GitHub Enterprise Server, atau Bitbucket, akan `invalidInputException` dilemparkan.

Untuk dapat melaporkan status build ke penyedia sumber, pengguna yang terkait dengan penyedia sumber harus memiliki akses tulis ke repo. Jika pengguna tidak memiliki akses tulis, status build tidak dapat diperbarui. Untuk informasi selengkapnya, lihat Akses penyedia sumber.

sumber/membangun StatusConfig

Berisi informasi yang menentukan cara proyek CodeBuild build melaporkan status build ke penyedia sumber. Opsi ini hanya digunakan ketika jenis sumbernya`GITHUB`,`GITHUB_ENTERPRISE`, atau`BITBUCKET`.

source/buildStatusConfig/konteks

Untuk sumber Bitbucket, parameter ini digunakan untuk `name` parameter dalam status komit Bitbucket. Untuk GitHub sumber, parameter ini digunakan untuk `context` parameter dalam status GitHub komit.

Misalnya, Anda dapat memiliki `context` berisi nomor build dan pemicu webhook menggunakan variabel CodeBuild lingkungan:

```
AWS CodeBuild sample-project Build #$CODEBUILD_BUILD_NUMBER -
 $CODEBUILD_WEBHOOK_TRIGGER
```

Ini menghasilkan konteks yang muncul seperti ini untuk build #24 yang dipicu oleh peristiwa permintaan tarik webhook:

```
AWS CodeBuild sample-project Build #24 - pr/8
```

source/buildStatusConfig/TargeTurl

Untuk sumber Bitbucket, parameter ini digunakan untuk `url` parameter dalam status komit Bitbucket. Untuk GitHub sumber, parameter ini digunakan untuk `target_url` parameter dalam status GitHub komit.

Misalnya, Anda dapat mengatur `targetUrl` ke `https://aws.amazon.com/codebuild/`*`<path to build>`* dan status komit akan ditautkan ke URL ini.

Anda juga dapat menyertakan variabel CodeBuild lingkungan `targetUrl` untuk menambahkan informasi tambahan ke URL. Misalnya, untuk menambahkan wilayah build ke URL, setel `targetUrl` ke:

```
"targetUrl": "https://aws.amazon.com/codebuild/<path to build>?region=
$AWS_REGION"
```

Jika wilayah build adalah`us-east-2`, ini akan berkembang menjadi:

```
https://aws.amazon.com/codebuild/<path to build>?region=us-east-2
```

sumber/git SubmodulesConfig

Tidak wajib. Informasi tentang konfigurasi submodul Git. Digunakan dengan CodeCommit, GitHub, Server GitHub Perusahaan, dan Bitbucket saja.

source/gitSubmodulesConfig/fetchSubmodules

Setel `fetchSubmodules` ke `true` jika Anda ingin menyertakan submodul Git dalam repositori Anda. Submodul Git yang disertakan harus dikonfigurasi sebagai HTTPS.

sumber/ InsecureSsl

Tidak wajib. Digunakan dengan GitHub Enterprise Server saja. Tetapkan nilai ini `true` untuk mengabaikan peringatan TLS saat menghubungkan ke repositori proyek GitHub Enterprise Server Anda. Nilai default adalah `false`. `InsecureSsl` harus digunakan hanya untuk tujuan pengujian. Ini tidak boleh digunakan di lingkungan produksi.

sumber/SourceIdentifier

Pengidentifikasi yang ditentukan pengguna untuk sumber proyek. Opsional untuk sumber utama. Diperlukan untuk sumber sekunder.

SECONDARYSOURCES

Tidak wajib. Array ProjectSourceobjek yang berisi informasi tentang sumber sekunder untuk proyek build. Anda dapat menambahkan hingga 12 sumber sekunder. `secondarySources`Objek menggunakan properti yang sama yang digunakan oleh objek. Dalam objek sumber sekunder, `sourceIdentifier` diperlukan.

sekunder SourceVersions

Tidak wajib. Susunan objek ProjectSourceVersion. Jika `secondarySourceVersions` ditentukan pada level build, maka mereka lebih diutamakan daripada ini.

SOURCEVERSION

Tidak wajib. Versi input build yang akan dibangun untuk proyek ini. Jika tidak ditentukan, versi terbaru digunakan. Jika ditentukan, itu harus berupa salah satu dari:

- Untuk CodeCommit, ID komit, cabang, atau tag Git yang akan digunakan.

- Untuk GitHub, ID komit, ID permintaan tarik, nama cabang, atau nama tag yang sesuai dengan versi kode sumber yang ingin Anda buat. Jika ID permintaan tarik ditentukan, itu harus menggunakan format `pr/pull-request-ID` (misalnya `pr/25`). Jika nama cabang ditentukan, ID melakukan HEAD cabang digunakan. Jika tidak ditentukan, ID melakukan HEAD cabang default digunakan.
- Untuk GitLab, ID komit, ID permintaan tarik, nama cabang, nama tag, atau referensi, dan ID komit. Untuk informasi selengkapnya, lihat Contoh versi sumber dengan AWS CodeBuild.
- Untuk Bitbucket, ID komit, nama cabang, atau nama tag yang sesuai dengan versi kode sumber yang ingin Anda buat. Jika nama cabang ditentukan, ID melakukan HEAD cabang digunakan. Jika tidak ditentukan, ID melakukan HEAD cabang default digunakan.
- Untuk Amazon S3, ID versi objek yang mewakili file ZIP input build yang akan digunakan.

Jika `sourceVersion` ditentukan pada tingkat pembuatan, maka versi tersebut akan mengambil alih `sourceVersion` ini (di tingkat proyek). Untuk informasi selengkapnya, lihat Contoh versi sumber dengan AWS CodeBuild.

artefak

Wajib. ProjectArtifactsObjek yang berisi informasi tentang pengaturan artefak keluaran proyek build ini. Setelah Anda menambahkan `artifacts` objek, Anda dapat menambahkan hingga 12 artefak lagi menggunakan. Pengaturan ini meliputi yang berikut:

artefak/jenis

Wajib. Jenis artefak output build. Nilai yang valid adalah:

- `CODEPIPELINE`
- `NO_ARTIFACTS`
- `S3`

artefak/lokasi

Hanya digunakan dengan tipe `S3` artefak. Tidak digunakan untuk jenis artefak lainnya.

Nama bucket keluaran yang Anda buat atau identifikasi dalam prasyarat.

artefak/jalur

Hanya digunakan dengan tipe `S3` artefak. Tidak digunakan untuk jenis artefak lainnya.

Jalur bucket keluaran untuk menempatkan file atau folder ZIP. Jika Anda tidak menentukan nilai untuk`path`, CodeBuild gunakan `namespaceType` (jika ditentukan) dan `name` untuk menentukan jalur dan nama file atau folder ZIP keluaran build. Misalnya, jika Anda menentukan `MyPath` untuk `path` dan `MyArtifact.zip` untuk`name`, jalur dan nama akan menjadi`MyPath/MyArtifact.zip`.

artefak/Namespacetype

Hanya digunakan dengan tipe `S3` artefak. Tidak digunakan untuk jenis artefak lainnya.

Namespace dari file atau folder ZIP keluaran build. Nilai yang valid mencakup `BUILD_ID` dan `NONE`. Gunakan `BUILD_ID` untuk menyisipkan ID build ke jalur file atau folder ZIP keluaran build. Jika tidak sesuai, gunakan `NONE`. Jika Anda tidak menentukan nilai untuk`namespaceType`, CodeBuild gunakan `path` (jika ditentukan) dan `name` untuk menentukan jalur dan nama file atau folder ZIP keluaran build. Misalnya, jika Anda menentukan `MyPath` untuk`path`, `BUILD_ID` untuk`namespaceType`, dan `MyArtifact.zip` untuk`name`, jalur dan nama akan menjadi`MyPath/`*`build-ID`*`/MyArtifact.zip`.

artefak/nama

Hanya digunakan dengan tipe `S3` artefak. Tidak digunakan untuk jenis artefak lainnya.

Nama file ZIP keluaran build atau folder di dalamnya`location`. Misalnya, jika Anda menentukan `MyPath` untuk `path` dan `MyArtifact.zip` untuk`name`, jalur dan nama akan menjadi`MyPath/MyArtifact.zip`.

artefak/timpa ArtifactName

Hanya digunakan dengan tipe artefak S3. Tidak digunakan untuk jenis artefak lainnya.

Tidak wajib. Jika disetel ke`true`, nama yang ditentukan dalam `artifacts` blok file buildspec akan ditimpa. `name` Untuk informasi selengkapnya, lihat Buat referensi spesifikasi untuk CodeBuild.

artefak/kemasan

Hanya digunakan dengan tipe `S3` artefak. Tidak digunakan untuk jenis artefak lainnya.

Tidak wajib. Menentukan bagaimana untuk paket artefak. Nilai yang diizinkan adalah:

NONE

Buat folder yang berisi artefak build. Ini adalah nilai default.

ZIP

Buat file ZIP yang berisi artefak build.

SECONDARYARTIFACTS

Tidak wajib. Array ProjectArtifactsobjek yang berisi informasi tentang pengaturan artefak sekunder untuk proyek build. Anda dapat menambahkan hingga 12 artefak sekunder. `secondaryArtifacts`Menggunakan banyak pengaturan yang sama yang digunakan oleh objek.

cache

Wajib. ProjectCacheObjek yang berisi informasi tentang pengaturan cache proyek build ini. Untuk informasi selengkapnya, lihat Membangun cache.

lingkungan

Wajib. ProjectEnvironmentObjek yang berisi informasi tentang pengaturan lingkungan build proyek ini. Pengaturan ini meliputi:

lingkungan/tipe

Wajib. Jenis lingkungan bangunan. Untuk informasi selengkapnya, lihat mengetik Referensi CodeBuild API.

lingkungan/gambar

Wajib. Pengidentifikasi gambar Docker yang digunakan oleh lingkungan build ini. Biasanya, pengidentifikasi ini dinyatakan sebagai*`image-name`*:*`tag`*. Misalnya, di repositori Docker yang CodeBuild digunakan untuk mengelola gambar Docker-nya, ini bisa jadi. `aws/codebuild/standard:5.0` Di Docker Hub,`maven:3.3.9-jdk-8`. Di Amazon ECR,*`account-id`*`.dkr.ecr.`*`region-id`*`.amazonaws.com/`*`your-Amazon-ECR-repo-name`*`:`*`tag`*. Untuk informasi selengkapnya, lihat Gambar Docker disediakan oleh CodeBuild.

lingkungan/ComputeType

Wajib. Menentukan sumber daya komputasi yang digunakan oleh lingkungan build ini. Untuk informasi selengkapnya, lihat ComputeType di Referensi API. CodeBuild

lingkungan/sertifikat

Tidak wajib. ARN bucket Amazon S3, awalan jalur, dan kunci objek yang berisi sertifikat. PEM-encoded Kunci objek dapat berupa file.pem atau file.zip yang berisi sertifikat. PEM-encoded

Misalnya, jika nama bucket Amazon S3 Anda, awalan jalur Anda adalah*<my-bucket>*, dan nama kunci objek Anda *<cert><certificate.pem>*, maka format yang dapat diterima untuk `certificate` are atau. *<my-bucket/cert/certificate.pem>* `arn:aws:s3:::`*<my-bucket/cert/certificate.pem>*

lingkungan/lingkunganVariabel

Tidak wajib. Array EnvironmentVariableobjek yang berisi variabel lingkungan yang ingin Anda tentukan untuk lingkungan build ini. Setiap variabel lingkungan dinyatakan sebagai objek yang berisi`name`,`value`, dan `type` dari`name`,`value`, dan`type`.

Konsol dan AWS CLI pengguna dapat melihat semua variabel lingkungan. Jika Anda tidak memiliki kekhawatiran tentang visibilitas variabel lingkungan Anda, atur `name` dan`value`, dan atur `type` ke`PLAINTEXT`.

Kami menyarankan Anda menyimpan variabel lingkungan dengan nilai sensitif, seperti ID kunci AWS akses, kunci akses AWS rahasia, atau kata sandi, sebagai parameter di Amazon EC2 Systems Manager Parameter Store AWS Secrets Manager atau. Untuk`name`, untuk parameter yang disimpan itu, tetapkan pengenal CodeBuild untuk referensi.

Jika Anda menggunakan Amazon EC2 Systems Manager Parameter Store, `value` untuk, tetapkan nama parameter sebagai disimpan di Parameter Store. Atur `type` ke `PARAMETER_STORE`. Menggunakan parameter bernama `/CodeBuild/dockerLoginPassword` sebagai contoh, atur `name` ke`LOGIN_PASSWORD`. Atur `value` ke `/CodeBuild/dockerLoginPassword`. Atur `type` ke `PARAMETER_STORE`.

> **Important**
>
> Jika Anda menggunakan Amazon EC2 Systems Manager Parameter Store, sebaiknya Anda menyimpan parameter dengan nama parameter yang dimulai `/CodeBuild/` dengan (misalnya`/CodeBuild/dockerLoginPassword`,). Anda dapat menggunakan CodeBuild konsol untuk membuat parameter di Amazon EC2 Systems Manager. Pilih Buat parameter, lalu ikuti instruksi di kotak dialog. (Di kotak dialog itu, untuk kunci KMS, Anda dapat menentukan ARN kunci di AWS KMS akun Anda. Amazon EC2 Systems Manager menggunakan kunci ini untuk mengenkripsi nilai parameter selama penyimpanan dan mendekripsi selama pengambilan.) Jika Anda menggunakan CodeBuild konsol untuk membuat parameter, konsol memulai nama parameter dengan `/CodeBuild/` saat sedang disimpan. Untuk informasi selengkapnya, lihat Systems Manager Parameter Store dan Systems Manager Parameter Store Console Walkthrough di Panduan Pengguna Amazon EC2 Systems Manager.

Jika project build Anda mengacu pada parameter yang disimpan di Amazon EC2 Systems Manager Parameter Store, peran layanan project build harus mengizinkan tindakan `ssm:GetParameters` tersebut. Jika Anda memilih Peran layanan baru lebih awal, CodeBuild sertakan tindakan ini dalam peran layanan default untuk proyek build Anda. Namun, jika Anda memilih Peran layanan yang ada, Anda harus menyertakan tindakan ini ke peran layanan Anda secara terpisah.
Jika project build Anda merujuk ke parameter yang disimpan di Amazon EC2 Systems Manager Parameter Store dengan nama parameter yang tidak dimulai`/CodeBuild/`, dan Anda memilih Peran layanan baru, Anda harus memperbarui peran layanan tersebut untuk mengizinkan akses ke nama parameter yang tidak dimulai`/CodeBuild/`. Ini karena peran layanan itu memungkinkan akses hanya ke nama parameter yang dimulai dengan`/CodeBuild/`.
Jika Anda memilih Peran layanan baru, peran layanan menyertakan izin untuk mendekripsi semua parameter di bawah `/CodeBuild/` namespace di Amazon EC2 Systems Manager Parameter Store.
Variabel lingkungan yang Anda tetapkan menggantikan variabel lingkungan yang ada. Misalnya, jika image Docker sudah berisi variabel lingkungan bernama `MY_VAR` dengan nilai`my_value`, dan Anda menetapkan variabel lingkungan bernama `MY_VAR` dengan nilai`other_value`, `my_value` maka diganti dengan`other_value`. Demikian pula, jika image Docker sudah berisi variabel lingkungan bernama `PATH` dengan nilai`/usr/local/sbin:/usr/local/bin`, dan Anda menetapkan variabel lingkungan bernama `PATH` dengan nilai`$PATH:/usr/share/ant/bin`, `/usr/local/sbin:/usr/local/bin` maka digantikan oleh nilai `$PATH:/usr/share/ant/bin` literal.
Jangan mengatur variabel lingkungan dengan nama yang dimulai dengan`CODEBUILD_`. Awalan ini dicadangkan untuk penggunaan internal .
Jika variabel lingkungan dengan nama yang sama didefinisikan di beberapa tempat, nilainya ditentukan sebagai berikut:

- Nilai dalam panggilan operasi start build lebih diutamakan.
- Nilai dalam definisi proyek build akan diutamakan berikutnya.
- Nilai dalam deklarasi buildspec diutamakan paling rendah.

Jika Anda menggunakan Secrets Manager, untuk`value`, tetapkan nama parameter sebagai disimpan di Secrets Manager. Atur `type` ke `SECRETS_MANAGER`. Menggunakan rahasia bernama `/CodeBuild/dockerLoginPassword` sebagai contoh, atur `name` ke`LOGIN_PASSWORD`. Atur `value` ke `/CodeBuild/dockerLoginPassword`. Atur `type` ke `SECRETS_MANAGER`.

> **Important**
>
> Jika Anda menggunakan Secrets Manager, kami sarankan Anda menyimpan rahasia dengan nama yang dimulai dengan `/CodeBuild/` (misalnya,`/CodeBuild/dockerLoginPassword`). Untuk informasi lebih lanjut, lihat Apa itu AWS Secrets Manager? di Panduan Pengguna AWS Secrets Manager .
> Jika proyek build Anda mengacu pada rahasia yang disimpan di Secrets Manager, peran layanan proyek build harus mengizinkan `secretsmanager:GetSecretValue` tindakan tersebut. Jika Anda memilih Peran layanan baru lebih awal, CodeBuild sertakan tindakan ini dalam peran layanan default untuk proyek build Anda. Namun, jika Anda memilih Peran layanan yang ada, Anda harus menyertakan tindakan ini ke peran layanan Anda secara terpisah.
> Jika proyek build mengacu pada rahasia yang disimpan di Secrets Manager dengan nama rahasia yang tidak dimulai`/CodeBuild/`, dan Anda memilih Peran layanan baru, Anda harus memperbarui peran layanan untuk mengizinkan akses ke nama rahasia yang tidak dimulai`/CodeBuild/`. Ini karena peran layanan memungkinkan akses hanya ke nama rahasia yang dimulai dengan`/CodeBuild/`.
> Jika Anda memilih Peran layanan baru, peran layanan menyertakan izin untuk mendekripsi semua rahasia di bawah `/CodeBuild/` namespace di Secrets Manager.

lingkungan/registryCredential

Tidak wajib. RegistryCredentialObjek yang menentukan kredensyal yang menyediakan akses ke registri Docker pribadi.

environment/registryCredential/kredensi

Menentukan ARN atau nama kredensil dibuat menggunakan. AWS Managed Services Anda dapat menggunakan nama kredensialnya hanya jika ada di Wilayah Anda saat ini.

environment/registryCredential/CredentialProvider

Satu-satunya nilai yang valid adalah `SECRETS_MANAGER`.

Ketika ini diatur:

- `imagePullCredentials` harus diatur ke `SERVICE_ROLE`.
- Gambar tidak dapat berupa gambar yang dikuratori atau gambar Amazon ECR.

lingkungan/gambar PullCredentialsType

Tidak wajib. Jenis kredensil yang CodeBuild digunakan untuk menarik gambar di build Anda. Ada dua nilai yang valid:

CODEBUILD

`CODEBUILD`menentukan yang CodeBuild menggunakan kredensialnya sendiri. Anda harus mengedit kebijakan repositori Amazon ECR Anda untuk mempercayai prinsip layanan. CodeBuild

SERVICE_ROLE

Menentukan yang CodeBuild menggunakan peran layanan proyek build Anda.

Saat Anda menggunakan gambar registri lintas akun atau pribadi, Anda harus menggunakan `SERVICE_ROLE` kredensyal. Saat Anda menggunakan gambar yang CodeBuild dikuratori, Anda harus menggunakan `CODEBUILD` kredensil.

lingkungan/PrivilegedMode

Setel ke `true` hanya jika Anda berencana menggunakan proyek build ini untuk membangun image Docker. Jika tidak, semua build terkait yang mencoba berinteraksi dengan daemon Docker gagal. Anda juga harus memulai daemon Docker sehingga build Anda dapat berinteraksi dengannya. Salah satu cara untuk melakukannya adalah dengan menginisialisasi daemon Docker di `install` fase file buildspec Anda dengan menjalankan perintah build berikut. Jangan jalankan perintah ini jika Anda menentukan image lingkungan build yang disediakan oleh CodeBuild dengan dukungan Docker.

> **Note**
>
> Secara default, daemon Docker diaktifkan untuk build non-VPC. Jika Anda ingin menggunakan kontainer Docker untuk build VPC, lihat Runtime Privilege dan Kemampuan Linux di situs web Docker Docs dan aktifkan mode istimewa. Juga, Windows tidak mendukung mode istimewa.

```
- nohup /usr/local/bin/dockerd --host=unix:///var/run/docker.sock --
host=tcp://127.0.0.1:2375 --storage-driver=overlay2 &
- timeout 15 sh -c "until docker info; do echo .; sleep 1; done"
```

SERVICEROLE

Wajib. ARN dari peran layanan CodeBuild digunakan untuk berinteraksi dengan layanan atas nama pengguna (misalnya,`arn:aws:iam::`*`account-id`*`:role/`*`role-name`*).

auto RetryLimit

Tidak wajib. Jumlah percobaan ulang otomatis tambahan setelah build gagal. Misalnya, jika batas coba ulang otomatis disetel ke 2, CodeBuild akan memanggil `RetryBuild` API untuk mencoba ulang build Anda secara otomatis hingga 2 kali tambahan.

batas waktu InMinutes

Tidak wajib. Jumlah menit, antara 5 hingga 2160 (36 jam), setelah itu CodeBuild menghentikan build jika tidak selesai. Jika tidak ditentukan, default 60 digunakan. Untuk menentukan apakah dan kapan CodeBuild menghentikan build karena batas waktu, jalankan `batch-get-builds` perintah. Untuk menentukan apakah build telah berhenti, lihat di output untuk `buildStatus` nilai`FAILED`. Untuk menentukan kapan waktu build habis, lihat di output untuk `endTime` nilai yang terkait dengan `phaseStatus` nilai. `TIMED_OUT`

mengantri TimeoutInMinutes

Tidak wajib. Jumlah menit, antara 5 hingga 480 (8 jam), setelah itu CodeBuild menghentikan build jika masih antri. Jika tidak ditentukan, default 60 digunakan.

ENCRYPTIONKEY

Tidak wajib. Alias atau ARN yang digunakan CodeBuild oleh untuk mengenkripsi output build. AWS KMS key Jika Anda menentukan alias, gunakan format `arn:aws:kms:`*`region-ID`*`:`*`account-ID`*`:key/`*`key-ID`* atau, jika ada alias, gunakan formatnya. `alias/`*`key-alias`* Jika tidak ditentukan, kunci KMS yang AWS dikelola untuk Amazon S3 digunakan.

tag

Tidak wajib. Array objek Tag yang menyediakan tag yang ingin Anda kaitkan dengan proyek build ini. Anda dapat menentukan hingga 50 tag. Tag ini dapat digunakan oleh AWS layanan apa pun yang mendukung tag proyek CodeBuild build. Setiap tag dinyatakan sebagai objek dengan a `key` dan `avalue`.

VPCCONFIG

Tidak wajib. VpcConfigObjek yang berisi informasi tentang konfigurasi VPC untuk proyek Anda. Untuk informasi selengkapnya, lihat Gunakan AWS CodeBuild dengan Amazon Virtual Private Cloud.

Properti ini meliputi:

VPCID

Wajib. ID VPC yang menggunakan CodeBuild Jalankan perintah ini untuk mendapatkan daftar semua ID VPC di Wilayah Anda:

```
aws ec2 describe-vpcs --region <region-ID>
```

subnet

Wajib. Array ID subnet yang menyertakan sumber daya yang digunakan oleh CodeBuild. Jalankan perintah ini untuk mendapatkan ID ini:

```
aws ec2 describe-subnets --filters "Name=vpc-id,Values=<vpc-id>" --region <region-ID>
```

keamanan GroupIds

Wajib. Array ID grup keamanan yang digunakan oleh CodeBuild untuk memungkinkan akses ke sumber daya di VPC. Jalankan perintah ini untuk mendapatkan ID ini:

```
aws ec2 describe-security-groups --filters "Name=vpc-id,Values=<vpc-id>" --<region-ID>
```

BADGEENABLED

Tidak wajib. Menentukan apakah akan menyertakan lencana build dengan proyek Anda CodeBuild . Setel `true` untuk mengaktifkan lencana build, atau `false` lainnya. Untuk informasi selengkapnya, lihat Buat sampel lencana dengan CodeBuild.

LogsConfig

LogsConfigObjek yang berisi informasi tentang lokasi log build ini.

LogsConfig/awan WatchLogs

CloudWatchLogsConfigObjek yang berisi informasi tentang mendorong log ke CloudWatch Log.

LogsConfig/ S3logs

S3LogsConfigObjek yang berisi informasi tentang mendorong log ke Amazon S3.

berkas SystemLocations

Tidak wajib. Larik ProjectFileSystemsLocationobjek yang berisi informasi tentang konfigurasi Amazon EFS Anda.

membangun BatchConfig

Tidak wajib. `buildBatchConfig`Objek adalah ProjectBuildBatchConfigstruktur yang berisi informasi konfigurasi build batch untuk proyek.

BatchConfigbangun/ServiceRole

Peran layanan ARN untuk proyek pembuatan batch.

BatchConfigbangun/combineArtifacts

Nilai Boolean yang menentukan apakah akan menggabungkan artefak build untuk batch build menjadi satu lokasi artefak.

buildBatchConfig/restrictions/maksimum BuildsAllowed

Jumlah maksimum build yang diizinkan.

buildBatchConfig/restrictions/menghitung TypesAllowed

Array string yang menentukan jenis komputasi yang diizinkan untuk pembangunan batch. Lihat Membangun tipe komputasi lingkungan untuk nilai ini.

buildBatchConfig/restrictions/FleetsAllowed

Array string yang menentukan armada yang diizinkan untuk pembuatan batch. Lihat Menjalankan build pada armada kapasitas cadangan untuk informasi selengkapnya.

BatchConfigbangun/batas waktu InMinutes

Jumlah waktu maksimum, dalam hitungan menit, pembuatan batch harus diselesaikan.

BatchConfigbangun/batch ReportMode

Menentukan cara laporan status build dikirim ke penyedia sumber untuk build batch. Nilai yang valid meliputi:

`REPORT_AGGREGATED_BATCH`

(Default) Gabungkan semua status build ke dalam satu laporan status.

`REPORT_INDIVIDUAL_BUILDS`

Kirim laporan status terpisah untuk setiap build individu.

bersamaan BuildLimit

Jumlah maksimum membangun bersamaan yang diizinkan untuk proyek ini.

Pembuatan baru hanya dimulai jika jumlah pembuatan saat ini kurang dari atau sama dengan batas ini. Jika jumlah pembuatan saat ini memenuhi batas ini, pembuatan baru ditutup dan tidak dijalankan.

Buat proyek

Untuk membuat proyek, jalankan **`create-project`** perintah lagi, meneruskan file JSON Anda:

```
aws codebuild create-project --cli-input-json file://<json-file>
```

Jika berhasil, representasi JSON dari objek Project muncul di output konsol. Lihat Sintaks CreateProject Respons untuk contoh data ini.

Kecuali untuk nama proyek build, Anda dapat mengubah pengaturan proyek build apa pun nanti. Untuk informasi selengkapnya, lihat Mengubah setelan proyek build (AWS CLI).

Untuk mulai menjalankan build, lihatJalankan build (AWS CLI).

Jika kode sumber Anda disimpan dalam GitHub repositori, dan Anda CodeBuild ingin membangun kembali kode sumber setiap kali perubahan kode didorong ke repositori, lihat. Mulai menjalankan build secara otomatis ()AWS CLI

Buat proyek build (AWS SDK)

Untuk informasi tentang penggunaan AWS CodeBuild dengan AWS SDK, lihat. AWS Referensi SDK dan alat

Buat proyek build (CloudFormation)

Untuk informasi tentang menggunakan AWS CodeBuild with CloudFormation, lihat CloudFormation template untuk CodeBuild di Panduan AWS CloudFormation Pengguna.

Membuat aturan notifikasi

Anda dapat menggunakan aturan notifikasi untuk memberi tahu pengguna saat terjadi perubahan penting, seperti keberhasilan dan kegagalan build. Aturan notifikasi menentukan peristiwa dan topik Amazon SNS yang digunakan untuk mengirim notifikasi. Untuk informasi selengkapnya, lihat Apa itu notifikasi?

Anda dapat menggunakan konsol atau AWS CLI untuk membuat aturan notifikasi untuk AWS CodeBuild.

Untuk membuat aturan notifikasi (konsol)

1. Masuk ke Konsol Manajemen AWS dan buka CodeBuild konsol di https://console.aws.amazon.com/codebuild/.
2. Pilih Build, pilih Build projects, lalu pilih build project tempat Anda ingin menambahkan notifikasi.
3. Pada halaman proyek build, pilih Beri tahu, lalu pilih Buat aturan notifikasi. Anda juga dapat membuka halaman Pengaturan untuk proyek build dan memilih Create notification rule.
4. Di Nama notifikasi, masukkan nama untuk aturan.
5. Di Jenis detail, pilih Dasar jika Anda hanya menginginkan informasi yang diberikan ke Amazon yang EventBridge disertakan dalam notifikasi. Pilih Lengkap jika Anda ingin menyertakan informasi yang diberikan ke Amazon EventBridge dan informasi yang mungkin diberikan oleh CodeBuild atau pengelola notifikasi.

   Untuk informasi selengkapnya, lihat Memahami Konten Notifikasi dan Keamanan.
6. Di Peristiwa yang memicu notifikasi, pilih peristiwa yang ingin Anda kirimkan notifikasi. Untuk informasi selengkapnya, lihat Peristiwa untuk Aturan Pemberitahuan tentang Proyek Pembuatan.
7. Di Target, lakukan salah satu langkah berikut:

   - Jika Anda telah mengonfigurasi sumber daya untuk digunakan dengan notifikasi, di Pilih jenis target, pilih Pengembang Amazon Q di aplikasi obrolan (Slack) atau topik SNS. Di Pilih target, pilih nama klien (untuk klien Slack yang dikonfigurasi di Amazon Q Developer

dalam aplikasi obrolan) atau Nama Sumber Daya Amazon (ARN) dari topik Amazon SNS (untuk topik Amazon SNS yang sudah dikonfigurasi dengan kebijakan yang diperlukan untuk pemberitahuan).

- Jika Anda belum mengonfigurasi sumber daya untuk digunakan dengan notifikasi, pilih Buat target, lalu pilih Topik SNS. Berikan nama untuk topik setelah codestar-notifications-, lalu pilih Buat.

> **Note**
>
> - Jika Anda membuat topik Amazon SNS sebagai bagian dari membuat aturan notifikasi, kebijakan yang memungkinkan fitur notifikasi untuk mempublikasikan peristiwa untuk topik diterapkan untuk Anda. Menggunakan topik yang dibuat untuk aturan notifikasi membantu memastikan bahwa Anda hanya berlangganan pengguna yang ingin Anda terima notifikasinya tentang sumber daya ini.
> - Anda tidak dapat membuat Pengembang Amazon Q di klien aplikasi obrolan sebagai bagian dari pembuatan aturan notifikasi. Jika Anda memilih Pengembang Amazon Q dalam aplikasi obrolan (Slack), Anda akan melihat tombol yang mengarahkan Anda untuk mengonfigurasi klien di Amazon Q Developer dalam aplikasi obrolan. Memilih opsi itu membuka Pengembang Amazon Q di konsol aplikasi obrolan. Untuk informasi selengkapnya, lihat Mengonfigurasi Integrasi Antara Pemberitahuan dan Pengembang Amazon Q di aplikasi obrolan.
> - Jika ingin menggunakan topik Amazon SNS yang ada sebagai target, Anda harus menambahkan kebijakan yang diperlukan untuk AWS CodeStar Pemberitahuan selain kebijakan lain yang mungkin ada untuk topik tersebut. Untuk informasi selengkapnya, lihat Konfigurasi Topik Amazon SNS untuk Notifikasi dan Memahami Konten Notifikasi dan Keamanan.

8. Untuk menyelesaikan pembuatan aturan, pilih Kirim.
9. Anda harus membuat pengguna berlangganan ke topik Amazon SNS untuk aturan tersebut sebelum mereka dapat menerima notifikasi. Untuk informasi selengkapnya, lihat Buat Pengguna Berlangganan ke Topik Amazon SNS yang Merupakan Target. Anda juga dapat mengatur integrasi antara notifikasi dan Pengembang Amazon Q di aplikasi obrolan untuk mengirim pemberitahuan ke ruang obrolan Amazon Chime. Untuk informasi selengkapnya, lihat Mengkonfigurasi Integrasi Antara Pemberitahuan dan Pengembang Amazon Q di aplikasi obrolan.

Membuat aturan notifikasi (AWS CLI)

1. Di terminal atau prompt perintah, jalankan perintah create-notification rule untuk menghasilkan kerangka JSON:

```
aws codestarnotifications create-notification-rule --generate-cli-skeleton
 > rule.json
```

   Anda dapat memberi nama file apa pun yang Anda inginkan. Dalam contoh ini, file dinamai *rule.json*.

2. Buka file JSON dalam editor plaintext dan edit untuk memasukkan sumber daya, jenis peristiwa, dan target yang Anda inginkan untuk aturan tersebut. Contoh berikut menunjukkan aturan notifikasi bernama **MyNotificationRule** untuk proyek build bernama *MyBuildProject* AWS acccount dengan ID*123456789012*. Pemberitahuan dikirim dengan tipe detail lengkap ke topik Amazon SNS yang diberi nama *codestar-notifications-MyNotificationTopic* saat build berhasil:

```
{
    "Name": "MyNotificationRule",
    "EventTypeIds": [
        "codebuild-project-build-state-succeeded"
    ],
    "Resource": "arn:aws:codebuild:us-east-2:123456789012:MyBuildProject",
    "Targets": [
        {
            "TargetType": "SNS",
            "TargetAddress": "arn:aws:sns:us-east-2:123456789012:codestar-
notifications-MyNotificationTopic"
        }
    ],
    "Status": "ENABLED",
    "DetailType": "FULL"
}
```

   Simpan file tersebut.

3. Menggunakan file yang baru saja Anda edit, di terminal atau baris perintah, jalankan create-notification-rule perintah lagi untuk membuat aturan notifikasi:

```
aws codestarnotifications create-notification-rule --cli-input-json
 file://rule.json
```

4. Jika berhasil, perintah tersebut mengembalikan ARN aturan notifikasi, yang serupa dengan berikut ini:

```
{
    "Arn": "arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/
dc82df7a-EXAMPLE"
}
```

Ubah pengaturan proyek build di AWS CodeBuild

Anda dapat menggunakan AWS CodeBuild konsol AWS CLI, atau AWS SDK untuk mengubah setelan proyek build.

Jika Anda menambahkan pelaporan pengujian ke proyek build, pastikan peran IAM Anda memiliki izin yang dijelaskan. Izin laporan uji

Topik

- Mengubah setelan proyek build (konsol)
- Mengubah setelan proyek build (AWS CLI)
- Mengubah setelan proyek build (AWS SDK)

Mengubah setelan proyek build (konsol)

Untuk mengubah pengaturan untuk proyek build, lakukan prosedur berikut:

1. Buka AWS CodeBuild konsol di https://console.aws.amazon.com/codesuite/codebuild/home.
2. Di panel navigasi, pilih Membangun proyek.
3. Lakukan salah satu tindakan berikut:
   - Pilih tautan untuk proyek build yang ingin Anda ubah, lalu pilih Detail build.
   - Pilih tombol di samping proyek build yang ingin diubah, pilih Lihat detail, lalu pilih Build details.

Anda dapat memodifikasi bagian berikut:

Bagian-bagian

- Konfigurasi proyek
- Sumber
- Lingkungan
- Buildspec
- Konfigurasi batch
- Artifacts
- Beberapa catatan

Konfigurasi proyek

Di bagian Konfigurasi proyek, pilih Edit. Ketika perubahan Anda selesai, pilih Perbarui konfigurasi untuk menyimpan konfigurasi baru.

Anda dapat memodifikasi properti berikut.

Deskripsi

Masukkan deskripsi opsional proyek build untuk membantu pengguna lain memahami untuk apa proyek ini digunakan.

Membangun lencana

Pilih Aktifkan lencana build untuk membuat status build project Anda terlihat dan dapat disematkan. Untuk informasi selengkapnya, lihat Bangun contoh lencana.

> **Note**
>
> Build badge tidak berlaku jika penyedia sumber Anda adalah Amazon S3.

Aktifkan batas build bersamaan

Jika Anda ingin membatasi jumlah build bersamaan untuk proyek ini, lakukan langkah-langkah berikut:

1. Pilih Batasi jumlah build bersamaan yang dapat dimulai proyek ini.

2. Dalam batas build bersamaan, masukkan jumlah maksimum build bersamaan yang diizinkan untuk project ini. Batas ini tidak boleh lebih besar dari batas build bersamaan yang ditetapkan untuk akun. Jika Anda mencoba memasukkan angka yang lebih besar dari batas akun, pesan kesalahan akan ditampilkan.

Pembuatan baru hanya dimulai jika jumlah pembuatan saat ini kurang dari atau sama dengan batas ini. Jika jumlah pembuatan saat ini memenuhi batas ini, pembuatan baru ditutup dan tidak dijalankan.

Aktifkan akses build publik

Untuk membuat hasil build project Anda tersedia untuk umum, termasuk pengguna tanpa akses ke AWS akun, pilih Aktifkan akses build publik dan konfirmasikan bahwa Anda ingin membuat hasil build menjadi publik. Properti berikut digunakan untuk proyek pembangunan publik:

Peran layanan pembangunan publik

Pilih Peran layanan baru jika Anda ingin CodeBuild membuat peran layanan baru untuk Anda, atau Peran layanan yang ada jika Anda ingin menggunakan peran layanan yang ada.

Peran layanan build publik memungkinkan CodeBuild untuk membaca CloudWatch Log dan mengunduh artefak Amazon S3 untuk build proyek. Ini diperlukan untuk membuat log dan artefak bangunan proyek tersedia untuk umum.

Peran layanan

Masukkan nama peran layanan baru atau peran layanan yang ada.

Untuk membuat hasil build proyek Anda pribadi, hapus Aktifkan akses build publik.

Untuk informasi selengkapnya, lihat Dapatkan URL proyek build publik.

> **Warning**
>
> Hal-hal berikut harus diingat saat membuat hasil build proyek Anda menjadi publik:
>
> - Semua hasil build proyek, log, dan artefak, termasuk build yang dijalankan saat proyek bersifat pribadi, tersedia untuk umum.
> - Semua log build dan artefak tersedia untuk umum. Variabel lingkungan, kode sumber, dan informasi sensitif lainnya mungkin telah dihasilkan ke log dan artefak build. Anda harus berhati-hati tentang informasi apa yang dihasilkan ke log build. Beberapa praktik terbaik adalah:

- Jangan menyimpan nilai sensitif, terutama ID kunci AWS akses dan kunci akses rahasia, dalam variabel lingkungan. Sebaiknya gunakan Amazon EC2 Systems Manager Parameter Store AWS Secrets Manager atau untuk menyimpan nilai sensitif.
- Ikuti Praktik terbaik untuk menggunakan webhook untuk membatasi entitas mana yang dapat memicu build, dan jangan menyimpan buildspec dalam proyek itu sendiri, untuk memastikan bahwa webhook Anda seaman mungkin.

- Pengguna jahat dapat menggunakan build publik untuk mendistribusikan artefak berbahaya. Kami menyarankan agar administrator proyek meninjau semua permintaan tarik untuk memverifikasi bahwa permintaan tarik adalah perubahan yang sah. Kami juga menyarankan Anda memvalidasi artefak apa pun dengan checksum mereka untuk memastikan bahwa artefak yang benar sedang diunduh.

Informasi tambahan

Untuk Tag, masukkan nama dan nilai tag apa pun yang ingin digunakan AWS layanan pendukung. Gunakan Tambah baris untuk menambahkan tag. Anda dapat menambahkan hingga 50 tag.

Sumber

Di bagian Sumber, pilih Edit. Ketika perubahan Anda selesai, pilih Perbarui konfigurasi untuk menyimpan konfigurasi baru.

Anda dapat memodifikasi properti berikut:

Penyedia sumber

Pilih jenis penyedia kode sumber. Gunakan daftar berikut untuk membuat pilihan yang sesuai untuk penyedia sumber Anda:

> **Note**
>
> CodeBuild tidak mendukung Bitbucket Server.

Amazon S3

Bucket

Pilih nama bucket input yang berisi kode sumber.

Kunci objek S3 atau folder S3

Masukkan nama file ZIP atau path ke folder yang berisi kode sumber. Masukkan garis miring (/) untuk mengunduh semua yang ada di bucket S3.

Versi sumber

Masukkan ID versi objek yang mewakili build file input Anda. Untuk informasi selengkapnya, lihatContoh versi sumber dengan AWS CodeBuild.

CodeCommit

Repositori

Pilih repositori yang ingin Anda gunakan.

Jenis referensi

Pilih Cabang, tag Git, atau ID Komit untuk menentukan versi kode sumber Anda. Untuk informasi selengkapnya, lihat Contoh versi sumber dengan AWS CodeBuild.

> **Note**
>
> Kami menyarankan Anda memilih nama cabang Git yang tidak terlihat seperti ID komit, seperti `811dd1ba1aba14473856cee38308caed7190c0d` atau`5392f7`. Ini membantu Anda menghindari tabrakan checkout Git dengan commit yang sebenarnya.

Kedalaman klon Git

Pilih untuk membuat klon dangkal dengan riwayat terpotong ke jumlah komit yang ditentukan. Jika Anda ingin klon penuh, pilih Penuh.

Submodul Git

Pilih Gunakan submodul Git jika Anda ingin menyertakan submodul Git dalam repositori Anda.

Bitbucket

Kredensi

Pilih Default source credential atau Custom source credential dan ikuti petunjuk untuk mengelola kredensi sumber default atau menyesuaikan kredensi sumber.

Jenis koneksi

Pilih AWS CodeConnections, OAuth, Kata sandi aplikasi, atau Token akses pribadi untuk terhubung. CodeBuild

Koneksi

Pilih koneksi Bitbucket atau rahasia Secrets Manager untuk terhubung melalui jenis koneksi yang Anda tentukan.

Repositori

Pilih Repositori di akun Bitbucket saya atau repositori Publik dan masukkan URL repositori.

Versi sumber

Masukkan cabang, ID komit, tag, atau referensi dan ID komit. Untuk informasi selengkapnya, lihat Contoh versi sumber dengan AWS CodeBuild

> **Note**
>
> Kami menyarankan Anda memilih nama cabang Git yang tidak terlihat seperti ID komit, seperti `811dd1ba1aba14473856cee38308caed7190c0d` atau`5392f7`. Ini membantu Anda menghindari tabrakan checkout Git dengan commit yang sebenarnya.

Kedalaman klon Git

Pilih kedalaman klon Git untuk membuat klon dangkal dengan riwayat terpotong ke jumlah komit yang ditentukan. Jika Anda ingin klon penuh, pilih Penuh.

Submodul Git

Pilih Gunakan submodul Git jika Anda ingin menyertakan submodul Git dalam repositori Anda.

Membangun status

Pilih Laporkan status build ke penyedia sumber saat build dimulai dan selesai jika Anda ingin status awal dan penyelesaian build dilaporkan ke penyedia sumber.

Untuk dapat melaporkan status build ke penyedia sumber, pengguna yang terkait dengan penyedia sumber harus memiliki akses tulis ke repo. Jika pengguna tidak memiliki akses tulis, status build tidak dapat diperbarui. Untuk informasi selengkapnya, lihat Akses penyedia sumber.

Untuk konteks Status, masukkan nilai yang akan digunakan untuk `name` parameter dalam status komit Bitbucket. Untuk informasi selengkapnya, lihat build dalam dokumentasi Bitbucket API.

Untuk URL Target, masukkan nilai yang akan digunakan untuk `url` parameter dalam status komit Bitbucket. Untuk informasi selengkapnya, lihat build dalam dokumentasi Bitbucket API.

Status build yang dipicu oleh webhook selalu dilaporkan ke penyedia sumber. Agar status build yang dimulai dari konsol atau panggilan API dilaporkan ke penyedia sumber, Anda harus memilih setelan ini.

Jika build proyek Anda dipicu oleh webhook, Anda harus mendorong komit baru ke repo agar perubahan pada pengaturan ini diterapkan.

Dalam peristiwa webhook sumber utama, pilih Rebuild setiap kali perubahan kode didorong ke repositori ini jika Anda CodeBuild ingin membangun kode sumber setiap kali perubahan kode didorong ke repositori ini. Untuk informasi selengkapnya tentang webhook dan grup filter, lihat. Acara webhook Bitbucket

GitHub

Kredensi

Pilih Default source credential atau Custom source credential dan ikuti petunjuk untuk mengelola kredensi sumber default atau menyesuaikan kredensi sumber.

Jenis koneksi

Pilih GitHub Aplikasi, OAuth, atau Token akses pribadi untuk terhubung. CodeBuild

Koneksi

Pilih GitHub koneksi atau rahasia Secrets Manager untuk terhubung melalui jenis koneksi yang Anda tentukan.

Repositori

Pilih Repositori di GitHub akun saya, Repositori publik, atau webhook GitHub tercakup dan masukkan URL repositori.

Versi sumber

Masukkan cabang, ID komit, tag, atau referensi dan ID komit. Untuk informasi selengkapnya, lihat Contoh versi sumber dengan AWS CodeBuild

> **Note**
>
> Kami menyarankan Anda memilih nama cabang Git yang tidak terlihat seperti ID komit, seperti `811dd1ba1aba14473856cee38308caed7190c0d` atau`5392f7`. Ini membantu Anda menghindari tabrakan checkout Git dengan commit yang sebenarnya.

Kedalaman klon Git

Pilih kedalaman klon Git untuk membuat klon dangkal dengan riwayat terpotong ke jumlah komit yang ditentukan. Jika Anda ingin klon penuh, pilih Penuh.

Submodul Git

Pilih Gunakan submodul Git jika Anda ingin menyertakan submodul Git dalam repositori Anda.

Membangun status

Pilih Laporkan status build ke penyedia sumber saat build dimulai dan selesai jika Anda ingin status awal dan penyelesaian build dilaporkan ke penyedia sumber.

Untuk dapat melaporkan status build ke penyedia sumber, pengguna yang terkait dengan penyedia sumber harus memiliki akses tulis ke repo. Jika pengguna tidak memiliki akses tulis, status build tidak dapat diperbarui. Untuk informasi selengkapnya, lihat Akses penyedia sumber.

Untuk konteks Status, masukkan nilai yang akan digunakan untuk `context` parameter dalam status GitHub komit. Untuk informasi selengkapnya, lihat Membuat status komit di panduan GitHub pengembang.

Untuk URL Target, masukkan nilai yang akan digunakan untuk `target_url` parameter dalam status GitHub komit. Untuk informasi selengkapnya, lihat Membuat status komit di panduan GitHub pengembang.

Status build yang dipicu oleh webhook selalu dilaporkan ke penyedia sumber. Agar status build yang dimulai dari konsol atau panggilan API dilaporkan ke penyedia sumber, Anda harus memilih setelan ini.

Jika build proyek Anda dipicu oleh webhook, Anda harus mendorong komit baru ke repo agar perubahan pada pengaturan ini diterapkan.

Dalam peristiwa webhook sumber utama, pilih Rebuild setiap kali perubahan kode didorong ke repositori ini jika Anda CodeBuild ingin membangun kode sumber setiap kali perubahan kode didorong ke repositori ini. Untuk informasi selengkapnya tentang webhook dan grup filter, lihat. GitHub acara webhook

GitHub Enterprise Server

Kredensi

Pilih Default source credential atau Custom source credential dan ikuti petunjuk untuk mengelola kredensi sumber default atau menyesuaikan kredensi sumber.

Jenis koneksi

Pilih AWS CodeConnectionsatau Token akses pribadi untuk terhubung CodeBuild.

Koneksi

Pilih koneksi GitHub Enterprise atau rahasia Secrets Manager untuk terhubung melalui jenis koneksi yang Anda tentukan.

Repositori

Pilih Repositori di akun GitHub Enterprise saya atau webhook cakupan GitHub Enterprise dan masukkan URL repositori.

Versi sumber

Masukkan permintaan tarik, cabang, ID komit, tag, atau referensi, dan ID komit. Untuk informasi selengkapnya, lihat Contoh versi sumber dengan AWS CodeBuild.

> **Note**
>
> Kami menyarankan Anda memilih nama cabang Git yang tidak terlihat seperti ID komit, seperti `811dd1ba1aba14473856cee38308caed7190c0d` atau`5392f7`. Ini membantu Anda menghindari tabrakan checkout Git dengan commit yang sebenarnya.

Kedalaman klon Git

Pilih kedalaman klon Git untuk membuat klon dangkal dengan riwayat terpotong ke jumlah komit yang ditentukan. Jika Anda ingin klon penuh, pilih Penuh.

Submodul Git

Pilih Gunakan submodul Git jika Anda ingin menyertakan submodul Git dalam repositori Anda.

Membangun status

Pilih Laporkan status build ke penyedia sumber saat build dimulai dan selesai jika Anda ingin status awal dan penyelesaian build dilaporkan ke penyedia sumber.

Untuk dapat melaporkan status build ke penyedia sumber, pengguna yang terkait dengan penyedia sumber harus memiliki akses tulis ke repo. Jika pengguna tidak memiliki akses tulis, status build tidak dapat diperbarui. Untuk informasi selengkapnya, lihat Akses penyedia sumber.

Untuk konteks Status, masukkan nilai yang akan digunakan untuk `context` parameter dalam status GitHub komit. Untuk informasi selengkapnya, lihat Membuat status komit di panduan GitHub pengembang.

Untuk URL Target, masukkan nilai yang akan digunakan untuk `target_url` parameter dalam status GitHub komit. Untuk informasi selengkapnya, lihat Membuat status komit di panduan GitHub pengembang.

Status build yang dipicu oleh webhook selalu dilaporkan ke penyedia sumber. Agar status build yang dimulai dari konsol atau panggilan API dilaporkan ke penyedia sumber, Anda harus memilih setelan ini.

Jika build proyek Anda dipicu oleh webhook, Anda harus mendorong komit baru ke repo agar perubahan pada pengaturan ini diterapkan.

SSL tidak aman

Pilih Aktifkan SSL yang tidak aman untuk mengabaikan peringatan SSL saat menghubungkan ke repositori proyek Enterprise Anda GitHub .

Dalam peristiwa webhook sumber utama, pilih Rebuild setiap kali perubahan kode didorong ke repositori ini jika Anda CodeBuild ingin membangun kode sumber setiap kali perubahan kode didorong ke repositori ini. Untuk informasi selengkapnya tentang webhook dan grup filter, lihat. GitHub acara webhook

GitLab

Kredensi

Pilih Default source credential atau Custom source credential dan ikuti petunjuk untuk mengelola kredensi sumber default atau menyesuaikan kredensi sumber.

Jenis koneksi

AWS CodeConnectionsdigunakan untuk terhubung GitLab ke CodeBuild.

Koneksi

Pilih GitLab koneksi untuk terhubung AWS CodeConnections.

Repositori

Pilih repositori yang ingin Anda gunakan.

Versi sumber

Masukkan ID permintaan tarik, cabang, ID komit, tag, atau referensi, dan ID komit. Untuk informasi selengkapnya, lihat Contoh versi sumber dengan AWS CodeBuild.

> **Note**
>
> Kami menyarankan Anda memilih nama cabang Git yang tidak terlihat seperti ID komit, seperti `811dd1ba1aba14473856cee38308caed7190c0d` atau`5392f7`. Ini membantu Anda menghindari tabrakan checkout Git dengan commit yang sebenarnya.

Kedalaman klon Git

Pilih kedalaman klon Git untuk membuat klon dangkal dengan riwayat terpotong ke jumlah komit yang ditentukan. Jika Anda ingin klon penuh, pilih Penuh.

Membangun status

Pilih Laporkan status build ke penyedia sumber saat build dimulai dan selesai jika Anda ingin status awal dan penyelesaian build dilaporkan ke penyedia sumber.

Untuk dapat melaporkan status build ke penyedia sumber, pengguna yang terkait dengan penyedia sumber harus memiliki akses tulis ke repo. Jika pengguna tidak memiliki akses tulis, status build tidak dapat diperbarui. Untuk informasi selengkapnya, lihat Akses penyedia sumber.

GitLab Self Managed

Kredensi

Pilih Default source credential atau Custom source credential dan ikuti petunjuk untuk mengelola kredensi sumber default atau menyesuaikan kredensi sumber.

Jenis koneksi

AWS CodeConnectionsdigunakan untuk menghubungkan GitLab Self Managed ke CodeBuild.

Koneksi

Pilih koneksi GitLab Self Managed untuk terhubung AWS CodeConnections.

Repositori

Pilih repositori yang ingin Anda gunakan.

Versi sumber

Masukkan ID permintaan tarik, cabang, ID komit, tag, atau referensi, dan ID komit. Untuk informasi selengkapnya, lihat Contoh versi sumber dengan AWS CodeBuild.

> **Note**
>
> Kami menyarankan Anda memilih nama cabang Git yang tidak terlihat seperti ID komit, seperti `811dd1ba1aba14473856cee38308caed7190c0d` atau`5392f7`. Ini membantu Anda menghindari tabrakan checkout Git dengan commit yang sebenarnya.

Kedalaman klon Git

Pilih kedalaman klon Git untuk membuat klon dangkal dengan riwayat terpotong ke jumlah komit yang ditentukan. Jika Anda ingin klon penuh, pilih Penuh.

Membangun status

Pilih Laporkan status build ke penyedia sumber saat build dimulai dan selesai jika Anda ingin status awal dan penyelesaian build dilaporkan ke penyedia sumber.

Untuk dapat melaporkan status build ke penyedia sumber, pengguna yang terkait dengan penyedia sumber harus memiliki akses tulis ke repo. Jika pengguna tidak memiliki akses tulis, status build tidak dapat diperbarui. Untuk informasi selengkapnya, lihat Akses penyedia sumber.

Lingkungan

Di bagian Lingkungan, pilih Edit. Ketika perubahan Anda selesai, pilih Perbarui konfigurasi untuk menyimpan konfigurasi baru.

Anda dapat memodifikasi properti berikut:

Model penyediaan

Untuk mengubah model penyediaan, pilih Ubah model penyediaan dan lakukan salah satu hal berikut:

- Untuk menggunakan armada sesuai permintaan yang dikelola oleh AWS CodeBuild, pilih. On-demand Dengan armada sesuai permintaan, CodeBuild menyediakan komputasi untuk build Anda. Mesin-mesin dihancurkan saat build selesai. On-demand armada dikelola sepenuhnya, dan mencakup kemampuan penskalaan otomatis untuk menangani lonjakan permintaan.
- Untuk menggunakan armada kapasitas cadangan yang dikelola oleh AWS CodeBuild, pilih Kapasitas cadangan, lalu pilih nama Armada. Dengan armada kapasitas cadangan, Anda mengonfigurasi satu set instans khusus untuk lingkungan build Anda. Mesin ini tetap menganggur, siap untuk memproses build atau pengujian segera, dan mengurangi durasi pembuatan. Dengan armada kapasitas cadangan, mesin Anda selalu berjalan dan akan terus mengeluarkan biaya selama disediakan.

Untuk informasi, lihat Jalankan build pada armada kapasitas yang dicadangkan.

Citra lingkungan

Untuk mengubah image build, pilih Override image dan lakukan salah satu hal berikut:

- Untuk menggunakan image Docker yang dikelola oleh AWS CodeBuild, pilih Managed image, lalu buat pilihan dari Sistem operasi, Runtime, Image, dan versi Image. Buat pilihan dari jenis Lingkungan jika tersedia.
- Untuk menggunakan gambar Docker lain, pilih Custom image. Untuk jenis Lingkungan, pilih ARM, Linux, GPU Linux, atau Windows. Jika Anda memilih Registri lain, untuk URL registri eksternal, masukkan nama dan tag gambar Docker di Docker Hub, menggunakan format. *docker repository*/*docker image name* Jika Anda memilih Amazon ECR, gunakan repositori Amazon ECR dan gambar Amazon ECR untuk memilih gambar Docker di akun Anda. AWS
- Untuk menggunakan image Docker pribadi, pilih Custom image. Untuk jenis Lingkungan, pilih ARM, Linux, GPU Linux, atau Windows. Untuk registri Gambar, pilih Registri lain, lalu masukkan ARN kredensi untuk gambar Docker pribadi Anda. Kredensialnya harus dibuat oleh

Secrets Manager. Untuk informasi lebih lanjut, lihat Apa itu AWS Secrets Manager? di Panduan Pengguna AWS Secrets Manager .

> **Note**
>
> CodeBuild mengganti `ENTRYPOINT` untuk gambar Docker khusus.

Peran layanan

Lakukan salah satu tindakan berikut:

- Jika Anda tidak memiliki peran CodeBuild layanan, pilih Peran layanan baru. Di Nama peran, masukkan nama untuk peran baru.
- Jika Anda memiliki peran CodeBuild layanan, pilih Peran layanan yang ada. Di Peran ARN, pilih peran layanan.

> **Note**
>
> Saat menggunakan konsol untuk membuat proyek build, Anda dapat membuat peran CodeBuild layanan secara bersamaan. Secara default, peran hanya berfungsi dengan proyek build tersebut. Jika Anda menggunakan konsol untuk mengaitkan peran layanan ini dengan proyek build lain, peran akan diperbarui agar berfungsi dengan proyek build lainnya. Peran layanan dapat bekerja dengan hingga 10 proyek pembangunan.

Konfigurasi tambahan

Batas waktu

Tentukan nilai, antara 5 menit dan 36 jam, setelah itu CodeBuild menghentikan build jika tidak selesai. Jika jam dan menit dibiarkan kosong, nilai default 60 menit digunakan.

Diistimewakan

Pilih Aktifkan bendera ini jika Anda ingin membuat gambar Docker atau ingin build Anda mendapatkan hak istimewa yang lebih tinggi. hanya jika Anda berencana menggunakan proyek build ini untuk membangun gambar Docker. Jika tidak, semua build terkait yang mencoba berinteraksi dengan daemon Docker gagal. Anda juga harus memulai daemon Docker sehingga build Anda dapat berinteraksi dengannya. Salah satu cara untuk melakukannya adalah dengan menginisialisasi daemon Docker di `install` fase buildspec

Anda dengan menjalankan perintah build berikut. Jangan jalankan perintah ini jika Anda memilih image lingkungan build yang disediakan oleh CodeBuild dengan dukungan Docker.

> **Note**
>
> Secara default, daemon Docker diaktifkan untuk build non-VPC. Jika Anda ingin menggunakan kontainer Docker untuk build VPC, lihat Runtime Privilege dan Kemampuan Linux di situs web Docker Docs dan aktifkan mode istimewa. Juga, Windows tidak mendukung mode istimewa.

```
- nohup /usr/local/bin/dockerd --host=unix:///var/run/docker.sock --
host=tcp://127.0.0.1:2375 --storage-driver=overlay2 &
- timeout 15 sh -c "until docker info; do echo .; sleep 1; done"
```

VPC

Jika Anda CodeBuild ingin bekerja dengan VPC Anda:

- Untuk VPC, pilih ID VPC yang digunakan. CodeBuild
- Untuk Subnet VPC, pilih subnet yang menyertakan sumber daya yang digunakan. CodeBuild
- Untuk grup Keamanan VPC, pilih grup keamanan yang CodeBuild digunakan untuk memungkinkan akses ke sumber daya di VPC.

Untuk informasi selengkapnya, lihat Gunakan AWS CodeBuild dengan Amazon Virtual Private Cloud.

Hitung

Pilih salah satu opsi yang tersedia.

Kredensi registri

Tentukan kredensi registri saat proyek dikonfigurasi dengan gambar registri non-pribadi.

> **Note**
>
> Kredensi ini hanya akan digunakan jika gambar diganti dengan yang berasal dari pendaftar pribadi.

Variabel lingkungan

Masukkan nama dan nilai, lalu pilih jenis setiap variabel lingkungan untuk build yang akan digunakan.

> **Note**
>
> CodeBuild menetapkan variabel lingkungan untuk AWS Wilayah Anda secara otomatis. Anda harus menyetel variabel lingkungan berikut jika Anda belum menambahkannya ke buildspec.ymlmu:
>
> - AWS_ACCOUNT_ID
> - IMAGE_REPO_NAME
> - IMAGE_TAG

Konsol dan AWS CLI pengguna dapat melihat variabel lingkungan. Jika Anda tidak memiliki kekhawatiran tentang visibilitas variabel lingkungan Anda, atur bidang Nama dan Nilai, lalu setel Type ke Plaintext.

Kami menyarankan Anda menyimpan variabel lingkungan dengan nilai sensitif, seperti ID kunci AWS akses, kunci akses AWS rahasia, atau kata sandi sebagai parameter di Amazon EC2 Systems Manager Parameter Store AWS Secrets Manager atau.

Jika Anda menggunakan Amazon EC2 Systems Manager Parameter Store, lalu untuk Type, pilih Parameter. Untuk Nama, masukkan pengenal CodeBuild untuk referensi. Untuk Nilai, masukkan nama parameter seperti yang disimpan di Amazon EC2 Systems Manager Parameter Store. Menggunakan parameter bernama `/CodeBuild/dockerLoginPassword` sebagai contoh, untuk Type, pilih Parameter. Untuk Nama, masukkan `LOGIN_PASSWORD`. Untuk Nilai, masukkan `/CodeBuild/dockerLoginPassword`.

> **Important**
>
> Jika Anda menggunakan Amazon EC2 Systems Manager Parameter Store, sebaiknya Anda menyimpan parameter dengan nama parameter yang dimulai `/CodeBuild/` dengan (misalnya`/CodeBuild/dockerLoginPassword`,). Anda dapat menggunakan CodeBuild konsol untuk membuat parameter di Amazon EC2 Systems Manager. Pilih Buat parameter, lalu ikuti instruksi di kotak dialog. (Di kotak dialog itu, untuk kunci KMS, Anda dapat menentukan ARN kunci di AWS KMS akun

Anda. Amazon EC2 Systems Manager menggunakan kunci ini untuk mengenkripsi nilai parameter selama penyimpanan dan mendekripsi selama pengambilan.) Jika Anda menggunakan CodeBuild konsol untuk membuat parameter, konsol memulai nama parameter dengan `/CodeBuild/` saat sedang disimpan. Untuk informasi selengkapnya, lihat Systems Manager Parameter Store dan Systems Manager Parameter Store Console Walkthrough di Panduan Pengguna Amazon EC2 Systems Manager.

Jika project build Anda mengacu pada parameter yang disimpan di Amazon EC2 Systems Manager Parameter Store, peran layanan project build harus mengizinkan tindakan `ssm:GetParameters` tersebut. Jika Anda memilih Peran layanan baru lebih awal, CodeBuild sertakan tindakan ini dalam peran layanan default untuk proyek build Anda. Namun, jika Anda memilih Peran layanan yang ada, Anda harus menyertakan tindakan ini ke peran layanan secara terpisah.

Jika project build Anda merujuk ke parameter yang disimpan di Amazon EC2 Systems Manager Parameter Store dengan nama parameter yang tidak dimulai/ `CodeBuild/`, dan Anda memilih Peran layanan baru, Anda harus memperbarui peran layanan tersebut untuk mengizinkan akses ke nama parameter yang tidak dimulai/ `CodeBuild/`. Ini karena peran layanan itu memungkinkan akses hanya ke nama parameter yang dimulai dengan`/CodeBuild/`.

Jika Anda memilih Peran layanan baru, peran layanan menyertakan izin untuk mendekripsi semua parameter di bawah `/CodeBuild/` namespace di Amazon EC2 Systems Manager Parameter Store.

Variabel lingkungan yang Anda tetapkan menggantikan variabel lingkungan yang ada. Misalnya, jika image Docker sudah berisi variabel lingkungan bernama `MY_VAR` dengan nilai`my_value`, dan Anda menetapkan variabel lingkungan bernama `MY_VAR` dengan nilai`other_value`, `my_value` maka diganti dengan`other_value`. Demikian pula, jika image Docker sudah berisi variabel lingkungan bernama `PATH` dengan nilai/ `usr/local/sbin:/usr/local/bin`, dan Anda menetapkan variabel lingkungan bernama `PATH` dengan nilai`$PATH:/usr/share/ant/bin`, `/usr/local/sbin:/ usr/local/bin` maka digantikan oleh nilai `$PATH:/usr/share/ant/bin` literal.

Jangan mengatur variabel lingkungan apa pun dengan nama yang dimulai dengan`CODEBUILD_`. Awalan ini dicadangkan untuk penggunaan internal .

Jika variabel lingkungan dengan nama yang sama didefinisikan di beberapa tempat, nilainya ditentukan sebagai berikut:

- Nilai dalam panggilan operasi start build lebih diutamakan.
- Nilai dalam definisi proyek build akan diutamakan berikutnya.

- Nilai dalam deklarasi buildspec diutamakan paling rendah.

Jika Anda menggunakan Secrets Manager, untuk Type, pilih Secrets Manager. Untuk Nama, masukkan pengenal CodeBuild untuk referensi. Untuk Nilai, masukkan `reference-key` menggunakan pola*`secret-id`*`:`*`json-key`*`:`*`version-stage`*`:`*`version-id`*. Untuk informasi, lihat Secrets Manager reference-key in the buildspec file.

> **Important**
>
> Jika Anda menggunakan Secrets Manager, kami sarankan Anda menyimpan rahasia dengan nama yang dimulai dengan `/CodeBuild/` (misalnya,`/CodeBuild/dockerLoginPassword`). Untuk informasi lebih lanjut, lihat Apa itu AWS Secrets Manager? di Panduan Pengguna AWS Secrets Manager .
> Jika proyek build Anda mengacu pada rahasia yang disimpan di Secrets Manager, peran layanan proyek build harus mengizinkan `secretsmanager:GetSecretValue` tindakan tersebut. Jika Anda memilih Peran layanan baru lebih awal, CodeBuild sertakan tindakan ini dalam peran layanan default untuk proyek build Anda. Namun, jika Anda memilih Peran layanan yang ada, Anda harus menyertakan tindakan ini ke peran layanan secara terpisah.
> Jika proyek build mengacu pada rahasia yang disimpan di Secrets Manager dengan nama rahasia yang tidak dimulai`/CodeBuild/`, dan Anda memilih Peran layanan baru, Anda harus memperbarui peran layanan untuk mengizinkan akses ke nama rahasia yang tidak dimulai`/CodeBuild/`. Ini karena peran layanan memungkinkan akses hanya ke nama rahasia yang dimulai dengan`/CodeBuild/`.
> Jika Anda memilih Peran layanan baru, peran layanan menyertakan izin untuk mendekripsi semua rahasia di bawah `/CodeBuild/` namespace di Secrets Manager.

Buildspec

Di bagian Buildspec, pilih Edit. Ketika perubahan Anda selesai, pilih Perbarui konfigurasi untuk menyimpan konfigurasi baru.

Anda dapat memodifikasi properti berikut:

Membangun spesifikasi

Lakukan salah satu tindakan berikut:

- Jika kode sumber Anda menyertakan file buildspec, pilih Gunakan file buildspec. Secara default, CodeBuild mencari file bernama `buildspec.yml` di direktori root kode sumber. Jika file buildspec Anda menggunakan nama atau lokasi yang berbeda, masukkan jalurnya dari root sumber dalam nama Buildspec (misalnya, atau. `buildspec-two.yml configuration/buildspec.yml` Jika file buildspec ada di bucket S3, file tersebut harus berada di AWS Region yang sama dengan project build Anda. Tentukan file buildspec menggunakan ARN (misalnya,). `arn:aws:s3:::<my-codebuild-sample2>/buildspec.yml`
- Jika kode sumber Anda tidak menyertakan file buildspec, atau jika Anda ingin menjalankan perintah build yang berbeda dari yang ditentukan untuk `build` fase dalam `buildspec.yml` file di direktori root kode sumber, pilih Sisipkan perintah build. Untuk perintah Build, masukkan perintah yang ingin Anda jalankan dalam `build` fase. Untuk beberapa perintah, pisahkan setiap perintah dengan && (misalnya,`mvn test && mvn package`). Untuk menjalankan perintah di fase lain, atau jika Anda memiliki daftar panjang perintah untuk `build` fase tersebut, tambahkan `buildspec.yml` file ke direktori root kode sumber, tambahkan perintah ke file, lalu pilih Gunakan buildspec.ymldi direktori root kode sumber.

Untuk informasi selengkapnya, lihat Referensi Buildspec.

Konfigurasi batch

Di bagian konfigurasi Batch, pilih Edit. Ketika perubahan Anda selesai, pilih Perbarui konfigurasi untuk menyimpan konfigurasi baru. Untuk informasi selengkapnya, lihat Jalankan build dalam batch.

Anda dapat memodifikasi properti berikut:

Peran layanan Batch

Menyediakan peran layanan untuk build batch.

Pilih salah satu cara berikut:

- Jika Anda tidak memiliki peran layanan batch, pilih Peran layanan baru. Di peran Layanan, masukkan nama untuk peran baru.
- Jika Anda memiliki peran layanan batch, pilih Peran layanan yang ada. Dalam peran Layanan, pilih peran layanan.

Build batch memperkenalkan peran keamanan baru dalam konfigurasi batch. Peran baru ini diperlukan karena CodeBuild harus dapat memanggil`StartBuild`,`StopBuild`, dan `RetryBuild` tindakan atas nama Anda untuk menjalankan build sebagai bagian dari batch. Pelanggan harus menggunakan peran baru, dan bukan peran yang sama yang mereka gunakan dalam build mereka, karena dua alasan:

- Memberikan izin `StartBuild`, `StopBuild`, dan `RetryBuild` kepada peran build akan memungkinkan satu build untuk memulai lebih banyak build melalui buildspec.
- CodeBuild build batch memberikan batasan yang membatasi jumlah build dan tipe komputasi yang dapat digunakan untuk build dalam batch. Jika peran build memiliki izin ini, build dapat melewati batasan ini sendiri.

Jenis komputasi yang diizinkan untuk batch

Pilih jenis komputasi yang diizinkan untuk batch. Pilih semua yang berlaku.

Armada yang diizinkan untuk batch

Pilih armada yang diizinkan untuk batch. Pilih semua yang berlaku.

Build maksimum yang diizinkan dalam batch

Masukkan jumlah maksimum build yang diizinkan dalam batch. Jika batch melebihi batas ini, batch akan gagal.

Batas waktu Batch

Masukkan jumlah waktu maksimum untuk menyelesaikan batch build.

Gabungkan artefak

Pilih Gabungkan semua artefak dari batch ke satu lokasi agar semua artefak dari batch digabungkan menjadi satu lokasi.

Mode laporan Batch

Pilih mode laporan status build yang diinginkan untuk build batch.

> **Note**
>
> Bidang ini hanya tersedia jika sumber proyek adalah Bitbucket, GitHub, atau GitHub Enterprise, dan Laporkan status build ke penyedia sumber saat build Anda mulai dan selesai dipilih di bawah Sumber.

Build agregat

Pilih agar status untuk semua build dalam batch digabungkan menjadi satu laporan status.

Membangun individu

Pilih agar status build untuk semua build dalam batch dilaporkan secara terpisah.

Artifacts

Di bagian Artefak, pilih Edit. Ketika perubahan Anda selesai, pilih Perbarui konfigurasi untuk menyimpan konfigurasi baru.

Anda dapat memodifikasi properti berikut:

Jenis

Lakukan salah satu tindakan berikut:

- Jika Anda tidak ingin membuat artefak keluaran build apa pun, pilih Tidak ada artefak. Anda mungkin ingin melakukan ini jika Anda hanya menjalankan pengujian build atau Anda ingin mendorong image Docker ke repositori Amazon ECR.
- Untuk menyimpan output build dalam bucket S3, pilih Amazon S3, lalu lakukan hal berikut:
  - Jika Anda ingin menggunakan nama proyek Anda untuk file atau folder ZIP keluaran build, biarkan Nama kosong. Jika tidak, masukkan nama. (Jika Anda ingin mengeluarkan file ZIP, dan Anda ingin file ZIP memiliki ekstensi file, pastikan untuk memasukkannya setelah nama file ZIP.)
  - Pilih Aktifkan versi semantik jika Anda ingin nama yang ditentukan dalam file buildspec untuk mengganti nama apa pun yang ditentukan di konsol. Nama dalam file buildspec dihitung pada waktu pembuatan dan menggunakan bahasa perintah Shell. Misalnya, Anda dapat menambahkan tanggal dan waktu untuk nama artefak Anda sehingga selalu unik. Nama artefak unik mencegah artefak ditimpa. Untuk informasi selengkapnya, lihat Sintaks Buildspec.
  - Untuk nama Bucket, pilih nama bucket keluaran.
  - Jika Anda memilih Sisipkan perintah build sebelumnya dalam prosedur ini, maka untuk file Output, masukkan lokasi file dari build yang ingin Anda masukkan ke dalam file atau folder ZIP keluaran build. Untuk beberapa lokasi, pisahkan setiap lokasi dengan koma (misalnya,`appspec.yml, target/my-app.jar`). Untuk informasi lebih lanjut, lihat deskripsi `files` diSintaks Buildspec.

- Jika Anda tidak ingin artefak build Anda dienkripsi, pilih Hapus enkripsi artefak.

Untuk setiap set artefak sekunder yang Anda inginkan:

1. Untuk pengenal Artifact, masukkan nilai yang kurang dari 128 karakter dan hanya berisi karakter alfanumerik dan garis bawah.
2. Pilih Tambahkan artefak.
3. Ikuti langkah-langkah sebelumnya untuk mengonfigurasi artefak sekunder Anda.
4. Pilih Simpan artefak.

Konfigurasi tambahan

Kunci enkripsi

Lakukan salah satu tindakan berikut:

- Untuk menggunakan Kunci yang dikelola AWS Amazon S3 di akun Anda untuk mengenkripsi artefak keluaran build, biarkan kunci Enkripsi kosong. Ini adalah opsi default.
- Untuk menggunakan kunci yang dikelola pelanggan untuk mengenkripsi artefak keluaran build, di kunci Enkripsi, masukkan ARN kunci yang dikelola pelanggan. Gunakan format `arn:aws:kms:`*`region-ID`*`:`*`account-ID`*`:key/`*`key-ID`*.

Jenis cache

Untuk jenis Cache, pilih salah satu dari berikut ini:

- Jika Anda tidak ingin menggunakan cache, pilih Tidak ada cache.
- Jika Anda ingin menggunakan cache Amazon S3, pilih Amazon S3, lalu lakukan hal berikut:
  - Untuk Bucket, pilih nama bucket S3 tempat cache disimpan.
  - (Opsional) Untuk awalan jalur Cache, masukkan awalan jalur Amazon S3. Nilai awalan jalur Cache mirip dengan nama direktori. Itu memungkinkan Anda untuk menyimpan cache di bawah direktori yang sama dalam ember.

    > **Important**
    >
    > Jangan menambahkan garis miring (/) ke ujung awalan jalur.

- Jika Anda ingin menggunakan cache lokal, pilih Lokal, lalu pilih satu atau beberapa mode cache lokal.

> **Note**
>
> Mode cache lapisan Docker hanya tersedia untuk Linux. Jika Anda memilihnya, proyek Anda harus berjalan dalam mode istimewa.

Menggunakan cache menghemat banyak waktu pembuatan karena bagian lingkungan build yang dapat digunakan kembali disimpan dalam cache dan digunakan di seluruh build. Untuk informasi tentang menentukan cache dalam file buildspec, lihat. Sintaks Buildspec Untuk informasi lebih lanjut tentang caching, lihatCache dibangun untuk meningkatkan kinerja.

Beberapa catatan

Di bagian Log, pilih Edit. Ketika perubahan Anda selesai, pilih Perbarui konfigurasi untuk menyimpan konfigurasi baru.

Anda dapat memodifikasi properti berikut:

Pilih log yang ingin Anda buat. Anda dapat membuat CloudWatch Log Amazon, log Amazon S3, atau keduanya.

CloudWatch

Jika Anda ingin CloudWatch log Amazon Log:

CloudWatch log

Pilih CloudWatch log.

Nama grup

Masukkan nama grup CloudWatch log Amazon Logs Anda.

Nama aliran

Masukkan nama aliran CloudWatch log Amazon Logs Anda.

S3

Jika Anda ingin log Amazon S3:

Log S3

Pilih log S3.

Bucket

Pilih nama bucket S3 untuk log Anda.

Awalan jalur

Masukkan awalan untuk log Anda.

Nonaktifkan enkripsi log S3

Pilih jika Anda tidak ingin log S3 Anda dienkripsi.

Mengubah setelan proyek build (AWS CLI)

Untuk informasi tentang menggunakan AWS CLI with AWS CodeBuild, lihatReferensi baris perintah.

Untuk memperbarui CodeBuild proyek dengan AWS CLI, Anda membuat file JSON dengan properti yang diperbarui dan meneruskan file itu ke `update-project`perintah. Properti apa pun yang tidak terkandung dalam file pembaruan tetap tidak berubah.

Dalam file JSON pembaruan, hanya `name` properti dan properti yang dimodifikasi yang diperlukan. `name`Properti mengidentifikasi proyek untuk dimodifikasi. Untuk setiap struktur yang dimodifikasi, parameter yang diperlukan untuk struktur tersebut juga harus disertakan. Misalnya, untuk memodifikasi lingkungan proyek, `environment/computeType` properti `environment/type` dan diperlukan. Berikut adalah contoh yang memperbarui citra lingkungan:

```
{
  "name": "<project-name>",
  "environment": {
    "type": "LINUX_CONTAINER",
    "computeType": "BUILD_GENERAL1_SMALL",
    "image": "aws/codebuild/amazonlinux-x86_64-standard:4.0"
  }
}
```

Jika Anda perlu mendapatkan nilai properti saat ini untuk sebuah proyek, gunakan batch-get-projectsperintah untuk mendapatkan properti saat ini dari proyek yang Anda modifikasi, dan tulis output ke file.

```
aws codebuild batch-get-projects --names "<project-name>" > project-info.json
```

*project-info.json*File berisi array proyek, sehingga tidak dapat digunakan secara langsung untuk memperbarui proyek. Namun, Anda dapat menyalin properti yang ingin Anda modifikasi dari *project-info.json* file dan menempelkannya ke file pembaruan Anda sebagai dasar untuk properti yang ingin Anda modifikasi. Untuk informasi selengkapnya, lihat Lihat detail proyek build (AWS CLI).

Ubah file JSON pembaruan seperti yang dijelaskan dalamBuat proyek build (AWS CLI), dan simpan hasil Anda. Setelah Anda selesai memodifikasi file JSON pembaruan, jalankan `update-project`perintah, melewati file JSON pembaruan.

```
aws codebuild update-project --cli-input-json file://<update-project-file>
```

Jika berhasil, proyek JSON yang diperbarui muncul di output. Jika ada parameter yang diperlukan yang hilang, pesan kesalahan ditampilkan dalam output yang mengidentifikasi parameter yang hilang. Misalnya, ini adalah pesan kesalahan yang ditampilkan jika `environment/type` parameter tidak ada:

```
aws codebuild update-project --cli-input-json file://update-project.json

Parameter validation failed:
Missing required parameter in environment: "type"
```

Mengubah setelan proyek build (AWS SDK)

Untuk informasi tentang penggunaan AWS CodeBuild dengan AWS SDK, lihat. AWS Referensi SDK dan alat

Beberapa token akses di CodeBuild

CodeBuild mendukung sumber token akses ke penyedia pihak ketiga dari rahasia Anda di dalam AWS Secrets Manager atau melalui AWS CodeConnections koneksi. Anda dapat menetapkan rahasia atau koneksi Anda sebagai kredensi default untuk interaksi dengan penyedia pihak ketiga tertentu seperti GitHub, GitHub Enterprise, atau Bitbucket.

Anda dapat mengatur kredensi sumber Anda di tiga tingkatan yang berbeda:

1. Kredensial tingkat akun untuk semua proyek: Ini adalah kredensi default untuk semua proyek dalam akun. AWS Mereka akan digunakan pada proyek ketika tidak ada kredenal tingkat proyek atau sumber yang ditentukan.

2. Kredensial tingkat sumber untuk repositori tertentu: Ini adalah saat rahasia atau AWS CodeConnections koneksi Secrets Manager didefinisikan pada sumber proyek. Kredensi ini hanya akan digunakan untuk operasi pada repositori sumber yang ditentukan. Ini memungkinkan Anda untuk mengatur beberapa token akses dengan cakupan izin yang berbeda dalam proyek yang sama, dan tidak menggunakan kredenal tingkat akun default.
3. Kredensial fallback tingkat proyek: Anda dapat mengatur kredensi fallback tingkat proyek dengan menggunakan `NO_SOURCE` sebagai tipe sumber utama dan menentukan rahasia atau koneksi di dalamnya. Gunakan ini ketika Anda memiliki beberapa sumber pada proyek tetapi ingin menggunakan kredenal yang sama untuk mereka, atau ketika Anda tidak ingin menggunakan kredenal tingkat akun default untuk proyek Anda.

Topik

- Langkah 1: Buat rahasia Secrets Manager atau AWS CodeConnections koneksi
- Langkah 2: Berikan akses peran IAM CodeBuild proyek ke rahasia Secrets Manager
- Langkah 3: Konfigurasikan Secrets Manager atau AWS CodeConnections token
- Opsi penyiapan tambahan

Langkah 1: Buat rahasia Secrets Manager atau AWS CodeConnections koneksi

Gunakan petunjuk berikut untuk membuat rahasia Secrets Manager atau AWS CodeConnections koneksi:

- Buat dan simpan token dalam rahasia Secrets Manager.
- Buat koneksi ke GitHub
- Membuat koneksi ke GitHub Enterprise Server
- Buat koneksi ke Bitbucket

Langkah 2: Berikan akses peran IAM CodeBuild proyek ke rahasia Secrets Manager

> **Note**
>
> Sebelum melanjutkan, Anda harus memiliki akses ke token yang dibuat di Secrets Manager atau AWS CodeConnections.

Untuk memberikan akses peran IAM CodeBuild proyek ke Secrets Manager atau AWS CodeConnections, Anda harus menambahkan kebijakan IAM berikut.

Untuk memberikan CodeBuild akses peran IAM proyek

1. Buat peran IAM untuk CodeBuild proyek Anda dengan mengikuti petunjuk Memungkinkan CodeBuild untuk berinteraksi dengan orang lain AWS layanan untuk CodeBuild proyek Anda.
2. Lakukan salah satu tindakan berikut:
   - Tambahkan kebijakan IAM berikut ke peran CodeBuild proyek Anda untuk memberikan akses ke rahasia Anda.

     JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue"
            ],
            "Resource": [
                "arn:aws:iam::*:role/Service*"
            ]
        }
    ]
}
```

(Opsional) Jika Anda menggunakan kunci terkelola AWS KMS pelanggan untuk mengenkripsi rahasia Secrets Manager, Anda dapat menambahkan pernyataan kebijakan berikut untuk memberikan akses.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": "arn:aws:iam::*:role/Service*",
            "Condition": {
                "StringEquals": {
                    "kms:EncryptionContext:SecretARN":
 "arn:aws:iam::*:role/Service*"
                }
            }
        }
    ]
}
```

- Tambahkan kebijakan IAM berikut ke peran CodeBuild proyek Anda untuk memberikan akses ke koneksi Anda.

  JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "codeconnections:GetConnectionToken",
                "codeconnections:GetConnection"
            ],
```

```
            "Resource": [
                "arn:aws:iam::*:role/Service*"
            ]
        }
    ]
}
```

Langkah 3: Konfigurasikan Secrets Manager atau AWS CodeConnections token

Anda dapat mengatur kredensi sumber Anda di tiga level berbeda dengan Secrets Manager atau AWS CodeConnections token.

Konfigurasikan Secrets Manager atau AWS CodeConnections token sebagai kredensi tingkat akun

Anda dapat mengonfigurasi rahasia Secrets Manager atau AWS CodeConnections koneksi sebagai kredensi tingkat akun dan menggunakannya dalam proyek.

Konsol Manajemen AWS

Untuk mengonfigurasi koneksi sebagai kredensi tingkat akun di Konsol Manajemen AWS

1. Untuk penyedia Sumber, pilih Bitbucket, GitHub, atau GitHub Enterprise.
2. Untuk Credential, lakukan salah satu hal berikut:
   - Pilih Kredensi sumber default untuk menggunakan kredensi sumber default akun Anda untuk diterapkan ke semua proyek.

     a. Jika Anda tidak tersambung ke penyedia sumber, pilih Kelola kredensi sumber default.
     b. Untuk tipe Credential, pilih jenis kredensi.
     c. Jika Anda memilih AWS CodeConnections, pilih untuk menggunakan koneksi yang ada atau buat koneksi baru.

     Jika Anda memilih jenis kredensi yang berbeda, untuk Layanan pilih layanan mana yang ingin Anda gunakan untuk menyimpan token Anda dan lakukan hal berikut:
     - Jika Anda memilih untuk menggunakan Secrets Manager, Anda dapat memilih untuk menggunakan koneksi rahasia yang ada atau membuat rahasia baru dan

memilih Simpan. Untuk informasi selengkapnya cara membuat rahasia baru, lihatBuat dan simpan token dalam rahasia Secrets Manager.

- Jika Anda memilih untuk menggunakan CodeBuild, masukkan token atau nama pengguna dan kata sandi aplikasi Anda, dan pilih Simpan. Untuk token API Bitbucket, masukkan alamat email akun Atlassian Anda dan token API sebagai gantinya.

- Pilih Kredensi sumber kustom untuk menggunakan kredensi sumber kustom untuk mengganti setelan default akun Anda.

  a. Untuk tipe Credential, pilih jenis kredensi.

  b. Di Koneksi, pilih untuk menggunakan koneksi yang ada atau buat koneksi baru.

AWS CLI

Untuk mengonfigurasi koneksi sebagai kredensi tingkat akun di AWS CLI

- Buka terminal (Linux, macOS, atau Unix) atau command prompt (Windows). Gunakan AWS CLI untuk menjalankan import-source-credentials perintah.

  Gunakan perintah berikut untuk mengonfigurasi rahasia Secrets Manager:

```
aws codebuild import-source-credentials \
    --token "<secret-arn>" \
    --server-type <source-provider> \
    --auth-type SECRETS_MANAGER \
    --region <aws-region>
```

  Gunakan perintah berikut untuk mengkonfigurasi AWS CodeConnections koneksi:

```
aws codebuild import-source-credentials \
    --token "<connection-arn>" \
    --server-type <source-provider> \
    --auth-type CODECONNECTIONS \
    --region <aws-region>
```

  Perintah ini memungkinkan Anda untuk mengimpor token sebagai kredenal sumber default tingkat akun. Saat Anda mengimpor kredensi menggunakan ImportSourceCredentialsAPI, CodeBuild akan menggunakan token untuk semua interaksi dengan penyedia sumber,

seperti webhook, pelaporan status build, dan operasi git clone kecuali kumpulan kredensional yang lebih spesifik telah dikonfigurasi dalam proyek.

Anda sekarang dapat menggunakan token dalam proyek build Anda dan menjalankannya. Untuk informasi selengkapnya, lihat Membuat proyek pembangunan di AWS CodeBuild dan Jalankan AWS CodeBuild build secara manual.

Konfigurasikan beberapa token sebagai kredenal tingkat sumber

Untuk menggunakan rahasia atau AWS CodeConnections koneksi Secrets Manager sebagai kredenal tingkat sumber, langsung rujuk token dalam CodeBuild proyek, dan mulai build.

Konsol Manajemen AWS

Untuk mengonfigurasi beberapa token sebagai kredensi tingkat sumber di Konsol Manajemen AWS

1. Untuk penyedia Sumber, pilih GitHub.
2. Untuk Credential, lakukan salah satu hal berikut:
   - Pilih Kredensi sumber default untuk menggunakan kredensi sumber default akun Anda untuk diterapkan ke semua proyek.
     a. Jika Anda tidak tersambung GitHub, pilih Kelola kredensi sumber default.
     b. Untuk jenis Kredenal, pilih GitHub Aplikasi.
     c. Di Koneksi, pilih untuk menggunakan koneksi yang ada atau buat koneksi baru.
   - Pilih Kredensi sumber kustom untuk menggunakan kredensi sumber kustom untuk mengganti setelan default akun Anda.
     a. Untuk jenis Kredenal, pilih GitHub Aplikasi.
     b. Di Koneksi, pilih untuk menggunakan koneksi yang ada atau buat koneksi baru.
3. Pilih Tambahkan sumber dan ulangi proses memilih penyedia sumber dan kredensialnya.

AWS CLI

Untuk mengonfigurasi beberapa token sebagai kredensi tingkat sumber di AWS CLI

- Buka terminal (Linux, macOS, atau Unix) atau command prompt (Windows). Gunakan AWS CLI untuk menjalankan create-project perintah.

Gunakan perintah berikut ini.

```
aws codebuild create-project --region <aws-region> \
    --name <project-name> \
    --artifacts type=NO_ARTIFACTS \
    --environment "type=LINUX_CONTAINER,
                   computeType=BUILD_GENERAL1_SMALL,
                   image=aws/codebuild/amazonlinux-x86_64-standard:5.0" \
    --service-role <service-role-name> \
    --source "type=GITHUB,
              location=<github-repository-1>,
              auth={type=SECRETS_MANAGER,resource=<secret-or-connection-arn-1>}"
 \
    --secondary-sources "type=GITHUB,
              location=<github-repository-2>,
              auth={type=SECRETS_MANAGER,resource=<secret-or-connection-arn-2>},
              sourceIdentifier=secondary"

aws codebuild start-build --region <aws-region> --project-name <project-name>
```

Tetapkan fallback kredenal sumber tingkat proyek

Untuk menyiapkan fallback kredenal sumber tingkat proyek, gunakan `NO_SOURCE` untuk sumber utama proyek Anda dan referensi token.

```
aws codebuild create-project \
    --name <project-name> \
    --service-role <service-role-name> \
    --artifacts type=NO_ARTIFACTS \
    --environment "type=LINUX_CONTAINER,
                   computeType=BUILD_GENERAL1_SMALL,
                   image=aws/codebuild/amazonlinux-x86_64-standard:5.0" \
    --service-role <service-role-name> \
    --source "type=NO_SOURCE,
              auth={type=SECRETS_MANAGER,resource=<secret-or-connection-arn>},
              buildspec=<buildspec>"
    --secondary-sources "type=GITHUB,
                         location=<github-repository>,
                         sourceIdentifier=secondary"
```

```
aws codebuild start-build --region <aws-region> --project-name <project_name>
```

Saat menggunakanNO_SOURCE, buildspec biasanya disediakan dalam model sumber karena tidak dikonfigurasi secara langsung untuk menggunakan sumber eksternal untuk mengambil buildspec. Umumnya, NO_SOURCE sumber akan menangani kloning semua repositori yang relevan dari dalam buildspec. Untuk memastikan kredenal yang dikonfigurasi tersedia untuk operasi tersebut, Anda dapat mengaktifkan `git-credential-helper` opsi di buildspec.

```
env:
  git-credential-helper: yes
```

Selama pembuatan, kemudian CodeBuild akan membaca `AuthServer` bidang dari token yang dikonfigurasi dan menggunakan kredenal token untuk semua permintaan git ke penyedia sumber pihak ketiga tertentu.

Opsi penyiapan tambahan

Anda dapat mengonfigurasi kredensi tingkat akun Secrets Manager dengan menggunakan CloudFormation templat. Anda dapat menggunakan CloudFormation template berikut untuk menetapkan kredensi tingkat akun:

```
Parameters:
  GitHubToken:
    Type: String
    NoEcho: true
    Default: placeholder
Resources:
  CodeBuildAuthTokenSecret:
    Type: AWS::SecretsManager::Secret
    Properties:
      Description: CodeBuild auth token
      Name: codebuild-auth-token
      SecretString:
        !Join
          - ''
          - - '{"ServerType":"GITHUB","AuthType":"PERSONAL_ACCESS_TOKEN","Token":"'
            - !Ref GitHubToken
            - '"}'
      Tags:
        - Key: codebuild:source:provider
          Value: github
```

```
        - Key: codebuild:source:type
          Value: personal_access_token
  CodeBuildSecretsManagerAccountCredential:
    Type: AWS::CodeBuild::SourceCredential
    Properties:
      ServerType: GITHUB
      AuthType: SECRETS_MANAGER
      Token: !Ref CodeBuildAuthTokenSecret
```

> **Note**
>
> Jika Anda juga membuat proyek di tumpukan yang sama, gunakan CloudFormation atribut DependsOnuntuk memastikan `AccountCredential` dibuat sebelum proyek.

Anda juga dapat mengonfigurasi Secrets Manager beberapa kredenal tingkat sumber dengan menggunakan CloudFormation templat. Anda dapat menggunakan CloudFormation template berikut untuk menggunakan beberapa token untuk menarik beberapa sumber:

```
Parameters:
  GitHubTokenOne:
    Type: String
    NoEcho: true
    Default: placeholder
  GitHubTokenTwo:
    Type: String
    NoEcho: true
    Default: placeholder

Resources:
  CodeBuildSecretsManagerProject:
    Type: AWS::CodeBuild::Project
    Properties:
      Name: codebuild-multitoken-example
      ServiceRole: <service-role>
      Environment:
        Type: LINUX_CONTAINER
        ComputeType: BUILD_GENERAL1_SMALL
        Image: aws/codebuild/amazonlinux-x86_64-standard:5.0
      Source:
        Type: GITHUB
        Location: <github-repository-one>
```

```
        Auth:
          Type: SECRETS_MANAGER
          Resource: !Ref CodeBuildAuthTokenSecretOne
      SecondarySources:
        - Type: GITHUB
          Location: <github-repository-two>
          Auth:
            Type: SECRETS_MANAGER
            Resource: !Ref CodeBuildAuthTokenSecretTwo
          SourceIdentifier: secondary
      Artifacts:
        Type: NO_ARTIFACTS
      LogsConfig:
        CloudWatchLogs:
          Status: ENABLED
  CodeBuildProjectIAMRoleSecretAccess:
    Type: AWS::IAM::RolePolicy
    Properties:
      RoleName: <role-name>
      PolicyName: CodeBuildProjectIAMRoleSecretAccessPolicy
      PolicyDocument:
        Version: '2012-10-17       '
        Statement:
          - Effect: Allow
            Action:
              - secretsmanager:GetSecretValue
            Resource:
              - !Ref CodeBuildAuthTokenSecretOne
              - !Ref CodeBuildAuthTokenSecretTwo
  CodeBuildAuthTokenSecretOne:
    Type: AWS::SecretsManager::Secret
    Properties:
      Description: CodeBuild auth token one
      Name: codebuild-auth-token-one
      SecretString:
        !Join
          - ''
          - - '{"ServerType":"GITHUB","AuthType":"PERSONAL_ACCESS_TOKEN","Token":"'
            - !Ref GitHubTokenOne
            - '"}'
      Tags:
        - Key: codebuild:source:provider
          Value: github
        - Key: codebuild:source:type
```

```
          Value: personal_access_token
  CodeBuildAuthTokenSecretTwo:
    Type: AWS::SecretsManager::Secret
    Properties:
      Description: CodeBuild auth token two
      Name: codebuild-auth-token-two
      SecretString:
        !Join
          - ''
          - - '{"ServerType":"GITHUB","AuthType":"PERSONAL_ACCESS_TOKEN","Token":"'
            - !Ref GitHubTokenTwo
            - '"}'
      Tags:
        - Key: codebuild:source:provider
          Value: github
        - Key: codebuild:source:type
          Value: personal_access_token
```

Hapus proyek build di AWS CodeBuild

Anda dapat menggunakan CodeBuild konsol AWS CLI, atau AWS SDK untuk menghapus proyek build. CodeBuild Jika Anda menghapus proyek, build-nya tidak dihapus.

> ⚠ Warning
>
> Anda tidak dapat menghapus proyek yang memiliki build dan kebijakan sumber daya. Untuk menghapus proyek dengan kebijakan sumber daya dan build, Anda harus terlebih dahulu menghapus kebijakan sumber daya dan menghapus buildnya.

Topik

- Hapus proyek build (konsol)
- Menghapus proyek build (AWS CLI)
- Menghapus proyek build (AWS SDK)

Hapus proyek build (konsol)

1. Buka AWS CodeBuild konsol di https://console.aws.amazon.com/codesuite/codebuild/home.

2. Di panel navigasi, pilih Membangun proyek.
3. Lakukan salah satu tindakan berikut:
   - Pilih tombol radio di sebelah proyek build yang ingin Anda hapus, lalu pilih Hapus.
   - Pilih tautan untuk proyek build yang ingin Anda hapus, lalu pilih Hapus.

> **Note**
>
> Secara default, hanya 10 proyek build terbaru yang ditampilkan. Untuk melihat proyek build lainnya, pilih nilai berbeda untuk Proyek per halaman atau gunakan panah mundur dan maju untuk melihat proyek.

Menghapus proyek build (AWS CLI)

1. Jalankan perintah `delete-project`:

```
aws codebuild delete-project --name name
```

   Ganti placeholder berikut:

   - *name*: String yang dibutuhkan. Nama proyek build yang akan dihapus. Untuk mendapatkan daftar proyek build yang tersedia, jalankan `list-projects` perintah. Untuk informasi selengkapnya, lihat Lihat daftar nama proyek build (AWS CLI).
2. Jika berhasil, tidak ada data dan tidak ada kesalahan yang muncul di output.

Untuk informasi lebih lanjut tentang menggunakan AWS CLI with AWS CodeBuild, lihatReferensi baris perintah.

Menghapus proyek build (AWS SDK)

Untuk informasi selengkapnya tentang penggunaan AWS CodeBuild dengan AWS SDK, lihat. AWS Referensi SDK dan alat

Dapatkan URL proyek build publik

AWS CodeBuild memungkinkan Anda membuat hasil build, log, dan artefak untuk proyek build Anda tersedia untuk masyarakat umum. Ini memungkinkan kontributor ke repositori sumber Anda untuk melihat hasil dan mengunduh artefak build, tanpa mengharuskan mereka memiliki akses ke akun. AWS

Ketika Anda membuat build proyek Anda tersedia untuk umum, semua hasil build proyek, log, dan artefak, termasuk build yang dijalankan saat proyek bersifat pribadi, tersedia untuk umum. Demikian juga, ketika Anda membuat proyek pembangunan publik pribadi, hasil build untuk proyek itu tidak lagi tersedia untuk umum.

Untuk informasi tentang cara mengubah visibilitas publik dari hasil build proyek Anda, lihatAktifkan akses build publik.

CodeBuild menyediakan URL untuk build publik untuk proyek Anda yang unik untuk proyek Anda.

Untuk mendapatkan URL publik untuk proyek build Anda, gunakan prosedur berikut.

Untuk mendapatkan URL proyek pembangunan publik

1. Buka AWS CodeBuild konsol di https://console.aws.amazon.com/codesuite/codebuild/home.
2. Di panel navigasi, pilih Membangun proyek.
3. Pilih tautan untuk proyek build yang ingin Anda dapatkan URL publiknya.
4. URL publik ditampilkan di bidang URL proyek Publik di bagian Konfigurasi. Anda dapat memilih tautan untuk membuka URL, atau menyalin URL dengan tombol salin.

⚠ Warning

Hal-hal berikut harus diingat saat membuat hasil build proyek Anda menjadi publik:

- Semua hasil build proyek, log, dan artefak, termasuk build yang dijalankan saat proyek bersifat pribadi, tersedia untuk umum.
- Semua log build dan artefak tersedia untuk umum. Variabel lingkungan, kode sumber, dan informasi sensitif lainnya mungkin telah dihasilkan ke log dan artefak build. Anda harus berhati-hati tentang informasi apa yang dihasilkan ke log build. Beberapa praktik terbaik adalah:

- Jangan menyimpan nilai sensitif, terutama ID kunci AWS akses dan kunci akses rahasia, dalam variabel lingkungan. Sebaiknya gunakan Parameter Store Amazon EC2 Systems Manager AWS Secrets Manager atau untuk menyimpan nilai sensitif.
- Ikuti Praktik terbaik untuk menggunakan webhook untuk membatasi entitas mana yang dapat memicu build, dan jangan menyimpan buildspec dalam proyek itu sendiri, untuk memastikan bahwa webhook Anda seaman mungkin.

- Pengguna jahat dapat menggunakan build publik untuk mendistribusikan artefak berbahaya. Kami menyarankan agar administrator proyek meninjau semua permintaan tarik untuk memverifikasi bahwa permintaan tarik adalah perubahan yang sah. Kami juga menyarankan Anda memvalidasi artefak apa pun dengan checksum mereka untuk memastikan bahwa artefak yang benar sedang diunduh.

Bagikan proyek pembangunan

Berbagi proyek memungkinkan pemilik proyek untuk berbagi AWS CodeBuild proyek mereka dengan AWS akun atau pengguna lain. Dalam model ini, akun yang memiliki proyek (pemilik) berbagi proyek dengan akun lain (konsumen). Konsumen tidak dapat mengedit atau menjalankan proyek.

Topik

- Bagikan proyek
- Layanan terkait
- Akses CodeBuild proyek yang dibagikan dengan Anda
- Batalkan berbagi proyek bersama
- Mengidentifikasi proyek bersama
- Izin proyek bersama

Bagikan proyek

Konsumen dapat menggunakan AWS CLI dan AWS CodeBuild konsol untuk melihat proyek dan build yang telah Anda bagikan. Konsumen tidak dapat mengedit atau menjalankan proyek.

Anda dapat menambahkan proyek ke berbagi sumber daya yang ada atau Anda dapat membuatnya di AWS RAM konsol.

> **Note**
>
> Anda tidak dapat menghapus proyek dengan build yang telah ditambahkan ke berbagi sumber daya.

Untuk berbagi proyek dengan unit organisasi atau seluruh organisasi, Anda harus mengaktifkan berbagi dengan AWS Organizations. Untuk informasi lebih lanjut, lihat Aktifkan pembagian dengan AWS Organizations dalam Panduan Pengguna AWS RAM .

Anda dapat menggunakan AWS CodeBuild konsol, AWS RAM konsol, atau AWS CLI untuk berbagi proyek yang Anda miliki.

Prasyarat untuk berbagi proyek

Sebelum Anda mulai berbagi proyek, pastikan AWS akun Anda memilikinya. Anda tidak dapat membagikan proyek yang telah dibagikan dengan Anda.

Untuk berbagi proyek yang Anda miliki (CodeBuild konsol)

1. Buka AWS CodeBuild konsol di https://console.aws.amazon.com/codesuite/codebuild/home.
2. Di panel navigasi, pilih Bangun proyek.

   > **Note**
   >
   > Secara default, hanya 10 proyek build terbaru yang ditampilkan. Untuk melihat lebih banyak proyek build, pilih ikon roda gigi, lalu pilih nilai yang berbeda untuk Proyek per halaman atau gunakan panah mundur dan maju.

3. Pilih proyek yang ingin Anda bagikan, lalu pilih Bagikan. Untuk informasi selengkapnya, lihat Membuat berbagi sumber daya di Panduan AWS RAM Pengguna.

Untuk berbagi proyek yang Anda miliki (AWS RAM konsol)

Lihat Membuat berbagi sumber daya di Panduan AWS RAM Pengguna.

Untuk berbagi proyek yang Anda miliki (AWS RAM perintah)

Gunakan perintah create-resource-share.

Untuk berbagi proyek yang Anda miliki (CodeBuildperintah)

Gunakan perintah put-resource-policy:

1. Buat file bernama `policy.json` dan salin yang berikut ke dalamnya.

   JSON

```
{
  "Version":"2012-10-17",
  "Statement":[{
    "Effect":"Allow",
    "Action":[
      "codebuild:BatchGetProjects",
      "codebuild:BatchGetBuilds",
      "codebuild:ListBuildsForProject"],
    "Resource":"arn:aws:iam::*:role/Service*"
  }]
}
```

2. Perbarui `policy.json` dengan proyek ARN dan pengidentifikasi untuk dibagikan. Contoh berikut memberikan akses read-only ke pengguna root untuk AWS akun yang diidentifikasi oleh 123456789012.

   JSON

```
{
  "Version":"2012-10-17",
  "Statement":[{
    "Effect":"Allow",
    "Principal":{
      "AWS": [
        "123456789012"
      ]
    },
    "Action":[
      "codebuild:BatchGetProjects",
      "codebuild:BatchGetBuilds",
      "codebuild:ListBuildsForProject"],
    "Resource":"arn:aws:codebuild:us-west-2:123456789012:project/my-project"
  }]
```

```
}
```

3. Jalankan perintah put-resource-policy.

```
aws codebuild put-resource-policy --resource-arn <project-arn> --policy file://
policy.json
```

4. Dapatkan pembagian AWS RAM sumber daya ARN.

```
aws ram list-resources --resource-owner SELF --resource-arns <project-arn>
```

Ini akan mengembalikan respons yang mirip dengan ini:

```
{
  "resources": [
    {
      "arn": "<project-arn>",
      "type": "<type>",
      "resourceShareArn": "<resource-share-arn>",
      "creationTime": "<creation-time>",
      "lastUpdatedTime": "<last-update-time>"
    }
  ]
}
```

Dari respons, salin *<resource-share-arn>* nilai untuk digunakan pada langkah berikutnya.

5. Jalankan perintah AWS RAM promote-resour ce-share-created-from-policy.

```
aws ram promote-resource-share-created-from-policy --resource-share-arn <resource-
share-arn>
```

Layanan terkait

Berbagi proyek terintegrasi dengan AWS Resource Access Manager (AWS RAM), layanan yang memungkinkan Anda berbagi AWS sumber daya dengan AWS akun apa pun atau melalui AWS Organizations. Dengan AWS RAM, Anda berbagi sumber daya dengan membuat pembagian sumber daya yang menentukan sumber daya dan konsumen untuk dibagikan. Konsumen dapat berupa AWS akun individu, unit organisasi di AWS Organizations, atau seluruh organisasi di dalamnya AWS Organizations.

Untuk informasi selengkapnya, silakan lihat Panduan Pengguna AWS RAM.

Akses CodeBuild proyek yang dibagikan dengan Anda

Untuk mengakses proyek bersama, peran IAM konsumen memerlukan `BatchGetProjects` izin. Anda dapat melampirkan kebijakan berikut ke peran IAM mereka:

```
{
    "Effect": "Allow",
    "Resource": [
        "*"
    ],
    "Action": [
        "codebuild:BatchGetProjects"
    ]
}
```

Untuk informasi selengkapnya, lihat Menggunakan kebijakan berbasis identitas untuk AWS CodeBuild.

Batalkan berbagi proyek bersama

Proyek yang tidak dibagikan, termasuk build-nya, hanya dapat diakses oleh pemiliknya. Jika Anda membatalkan berbagi proyek, AWS akun atau pengguna yang sebelumnya Anda bagikan tidak dapat mengakses proyek atau build-nya.

Untuk membatalkan berbagi proyek bersama yang Anda miliki, Anda harus menghapusnya dari berbagi sumber daya. Anda dapat menggunakan AWS CodeBuild konsol, AWS RAM konsol, atau AWS CLI untuk melakukan ini.

Untuk membatalkan berbagi proyek bersama yang Anda miliki (AWS RAM konsol)

Lihat Memperbarui Pembagian Sumber Daya di Panduan Pengguna.AWS RAM

Untuk membatalkan berbagi proyek bersama yang Anda miliki (AWS CLI)

Gunakan perintah disassociate-resource-share.

Untuk membatalkan berbagi proyek yang Anda miliki (CodeBuild perintah)

Jalankan perintah delete-resource-policy dan tentukan ARN proyek yang ingin Anda batalkan berbagi:

```
aws codebuild delete-resource-policy --resource-arn project-arn
```

Mengidentifikasi proyek bersama

Pemilik dan konsumen dapat menggunakan AWS CLI untuk mengidentifikasi proyek bersama.

Untuk mengidentifikasi proyek yang dibagikan dengan Anda AWS akun atau pengguna (AWS CLI)

Gunakan perintah list-shared-projects untuk mengembalikan proyek yang dibagikan dengan Anda.

Izin proyek bersama

Izin untuk pemilik

Pemilik proyek dapat mengedit proyek dan menggunakannya untuk menjalankan build.

Izin untuk konsumen

Konsumen proyek dapat melihat proyek dan build-nya, tetapi tidak dapat mengedit proyek atau menggunakannya untuk menjalankan build.

Tag membangun proyek

Tag adalah label atribut kustom yang Anda atau AWS tetapkan ke AWS sumber daya. Setiap AWS tag memiliki dua bagian:

- Kunci tag (misalnya, `CostCenter`, `Environment`, `Project`, atau `Secret`). Kunci tanda peka terhadap huruf besar dan kecil.
- Bidang opsional yang dikenal sebagai nilai tag (misalnya, `111122223333`, `Production`, atau nama tim). Mengabaikan nilai tag sama dengan menggunakan rangkaian kosong. Seperti kunci tanda, nilai tanda peka huruf besar dan kecil.

Bersama-sama ini dikenal sebagai pasangan nilai-kunci. Untuk informasi tentang jumlah tag yang dapat Anda miliki pada proyek dan pembatasan pada kunci dan nilai tag, lihatTanda.

Tag membantu Anda mengidentifikasi dan mengatur AWS sumber daya Anda. Banyak AWS layanan mendukung penandaan, sehingga Anda dapat menetapkan tag yang sama ke sumber daya dari layanan yang berbeda untuk menunjukkan bahwa sumber daya terkait. Misalnya, Anda dapat

menetapkan tag yang sama ke CodeBuild project yang Anda tetapkan ke bucket S3. Untuk informasi selengkapnya tentang penggunaan tag, lihat Menandai Praktik Terbaik.

Di CodeBuild, sumber daya utama adalah proyek dan kelompok laporan. Anda dapat menggunakan CodeBuild konsol, CodeBuild API AWS CLI, atau AWS SDK untuk menambahkan, mengelola, dan menghapus tag untuk proyek. Selain mengidentifikasi, mengatur, dan melacak proyek Anda dengan tag, Anda dapat menggunakan tag dalam kebijakan IAM untuk membantu mengontrol siapa yang dapat melihat dan berinteraksi dengan proyek Anda. Untuk contoh kebijakan akses berbasis tag, lihat Menggunakan tag untuk mengontrol akses AWS CodeBuild sumber daya.

> **Important**
>
> Saat menggunakan fitur kapasitas cadangan, data yang di-cache pada instance armada, termasuk file sumber, lapisan Docker, dan direktori cache yang ditentukan dalam buildspec, dapat diakses oleh proyek lain dalam akun yang sama. Ini dirancang dan memungkinkan proyek dalam akun yang sama untuk berbagi instance armada.

Topik

- Menambahkan tag ke proyek
- Lihat tag untuk proyek
- Mengedit tag untuk proyek
- Menghapus tag dari proyek

Menambahkan tag ke proyek

Menambahkan tag ke proyek dapat membantu Anda mengidentifikasi dan mengatur AWS sumber daya Anda dan mengelola akses ke sana. Pertama, Anda menambahkan satu atau lebih tag (pasangan nilai kunci) ke proyek. Perlu diingat bahwa ada batasan jumlah tag yang dapat Anda miliki pada suatu proyek. Ada pembatasan pada karakter yang dapat Anda gunakan di bidang kunci dan nilai. Untuk informasi selengkapnya, lihat Tanda. Setelah Anda memiliki tag, Anda dapat membuat kebijakan IAM untuk mengelola akses ke proyek berdasarkan tag ini. Anda dapat menggunakan CodeBuild konsol atau AWS CLI untuk menambahkan tag ke proyek.

> **Important**
>
> Saat menggunakan fitur kapasitas cadangan, data yang di-cache pada instance armada, termasuk file sumber, lapisan Docker, dan direktori cache yang ditentukan dalam buildspec, dapat diakses oleh proyek lain dalam akun yang sama. Ini dirancang dan memungkinkan proyek dalam akun yang sama untuk berbagi instance armada.

Untuk informasi selengkapnya tentang menambahkan tag ke proyek saat Anda membuatnya, lihatTambahkan tag ke proyek (konsol).

> **Important**
>
> Sebelum menambahkan tag ke proyek, pastikan untuk meninjau kebijakan IAM apa pun yang mungkin menggunakan tag untuk mengontrol akses ke sumber daya seperti proyek build. Untuk contoh kebijakan akses berbasis tag, lihat Menggunakan tag untuk mengontrol akses AWS CodeBuild sumber daya.

Topik

- Tambahkan tag ke proyek (konsol)
- Tambahkan tag ke project (AWS CLI)

Tambahkan tag ke proyek (konsol)

Anda dapat menggunakan CodeBuild konsol untuk menambahkan satu atau beberapa tag ke CodeBuild proyek.

1. Buka CodeBuild konsol di https://console.aws.amazon.com/codebuild/.
2. Di proyek Build, pilih nama proyek tempat Anda ingin menambahkan tag.
3. Pada panel navigasi, silakan pilih Pengaturan. Pilih Buat tag proyek.
4. Jika tidak ada tag yang ditambahkan ke proyek, pilih Tambah tag. Jika tidak, pilih Edit, lalu pilih Tambahkan tanda.
5. Di Kunci, masukkan sebuah nama untuk tag tersebut. Anda dapat menambahkan nilai opsional untuk tag di Nilai.
6. (Opsional) Untuk menambahkan tag lain, pilih Tambahkan tag lagi.

7. Setelah Anda selesai menambahkan tag, pilih Kirim.

Tambahkan tag ke project (AWS CLI)

Untuk menambahkan tag ke proyek saat Anda membuatnya, lihatBuat proyek build (AWS CLI). Di`create-project.json`, tambahkan tag Anda.

Dalam langkah-langkah ini, kami menganggap bahwa Anda telah menginstal versi terbaru dari AWS CLI atau diperbarui ke versi terkini. Untuk informasi lebih lanjut, lihat Menginstal AWS Command Line Interface.

Jika berhasil, perintah ini tidak mengembalikan apa pun.

Lihat tag untuk proyek

Tag dapat membantu Anda mengidentifikasi dan mengatur AWS sumber daya Anda dan mengelola akses ke sana. Untuk informasi lebih lanjut tentang penggunaan tanda, lihat laporan resmi Praktik terbaik. Untuk contoh kebijakan akses berbasis tag, lihat Menggunakan tag untuk mengontrol akses AWS CodeBuild sumber daya.

Lihat tag untuk proyek (konsol)

Anda dapat menggunakan CodeBuild konsol untuk melihat tag yang terkait dengan CodeBuild proyek.

1. Buka CodeBuild konsol di https://console.aws.amazon.com/codebuild/.
2. Di proyek Build, pilih nama proyek tempat Anda ingin melihat tag.
3. Pada panel navigasi, silakan pilih Pengaturan. Pilih Buat tag proyek.

Lihat tag untuk proyek (AWS CLI)

Untuk melihat tag untuk proyek build, jalankan perintah berikut. Gunakan nama proyek Anda untuk `--names` parameter.

```
aws codebuild batch-get-projects --names your-project-name
```

Jika berhasil, perintah ini menampilkan JSON-formatted informasi tentang proyek build Anda yang menyertakan sesuatu seperti berikut:

```
{
    "tags": {
        "Status": "Secret",
        "Team": "JanesProject"
    }
}
```

Jika proyek tidak memiliki tag, `tags` bagian kosong:

```
"tags": []
```

Mengedit tag untuk proyek

Anda dapat mengubah nilai untuk tag yang terkait dengan proyek. Anda juga dapat mengubah nama kunci, yang setara dengan menghapus tag saat ini dan menambahkan tag yang berbeda dengan nama baru dan nilai yang sama dengan kunci lainnya. Perlu diingat bahwa ada batasan pada karakter yang dapat Anda gunakan di bidang kunci dan nilai. Untuk informasi selengkapnya, lihat Tanda.

> **Important**
>
> Mengedit tag untuk proyek dapat memengaruhi akses ke proyek tersebut. Sebelum Anda mengedit nama (kunci) atau nilai tag untuk proyek, pastikan untuk meninjau kebijakan IAM apa pun yang mungkin menggunakan kunci atau nilai tag untuk mengontrol akses ke sumber daya seperti proyek build. Untuk contoh kebijakan akses berbasis tag, lihat Menggunakan tag untuk mengontrol akses AWS CodeBuild sumber daya.

Mengedit tag untuk proyek (konsol)

Anda dapat menggunakan CodeBuild konsol untuk mengedit tag yang terkait dengan CodeBuild proyek.

1. Buka CodeBuild konsol di https://console.aws.amazon.com/codebuild/.
2. Di proyek Build, pilih nama proyek tempat Anda ingin mengedit tag.
3. Pada panel navigasi, silakan pilih Pengaturan. Pilih Buat tag proyek.
4. Pilih Edit.
5. Lakukan salah satu tindakan berikut:

- Untuk mengubah tag, masukkan nama baru di Kunci. Mengubah nama tag adalah setara dengan menghapus tag dan menambahkan tag baru dengan nama kunci baru.
- Untuk mengubah nilai sebuah tag, masukkan nilai baru. Jika Anda ingin mengubah nilai menjadi nol, hapus nilai saat ini dan biarkan bidang kosong.

6. Setelah Anda selesai mengedit tag, pilih Kirim.

Mengedit tag untuk proyek (AWS CLI)

Untuk menambahkan, mengubah, atau menghapus tag dari proyek build, lihatMengubah setelan proyek build (AWS CLI). Perbarui `tags` bagian dalam JSON-formatted data yang Anda gunakan untuk memperbarui proyek.

Menghapus tag dari proyek

Anda dapat menghapus satu atau beberapa tag yang terkait dengan proyek. Menghapus tag tidak menghapus tag dari AWS sumber daya lain yang terkait dengan tag tersebut.

> **Important**
>
> Menghapus tag untuk proyek dapat memengaruhi akses ke proyek tersebut. Sebelum Anda menghapus tag dari proyek, pastikan untuk meninjau kebijakan IAM apa pun yang mungkin menggunakan kunci atau nilai tag untuk mengontrol akses ke sumber daya seperti proyek build. Untuk contoh kebijakan akses berbasis tag, lihat Menggunakan tag untuk mengontrol akses AWS CodeBuild sumber daya.

Menghapus tag dari proyek (konsol)

Anda dapat menggunakan CodeBuild konsol untuk menghapus asosiasi antara tag dan CodeBuild proyek.

1. Buka CodeBuild konsol di https://console.aws.amazon.com/codebuild/.
2. Di proyek Build, pilih nama proyek tempat Anda ingin menghapus tag.
3. Pada panel navigasi, silakan pilih Pengaturan. Pilih Buat tag proyek.
4. Pilih Edit.
5. Temukan tag yang ingin Anda hapus, lalu pilih Hapus tag.

6. Setelah Anda selesai menghapus tag, pilih Kirim.

Menghapus tag dari project (AWS CLI)

Untuk menghapus satu atau beberapa tag dari proyek build, lihatMengubah setelan proyek build (AWS CLI). Perbarui `tags` bagian dalam JSON-formatted data dengan daftar tag yang diperbarui yang tidak berisi tag yang ingin Anda hapus. Jika Anda ingin menghapus semua tag, perbarui `tags` bagian ke:

```
"tags: []"
```

> **Note**
>
> Jika Anda menghapus project CodeBuild build, semua asosiasi tag akan dihapus dari project build yang dihapus. Anda tidak perlu menghapus tag sebelum menghapus proyek build.

Gunakan pelari dengan AWS CodeBuild

AWS CodeBuild mendukung integrasi dengan pelari GitHub Actions, pelari yang dikelola sendiri, dan GitLab pelari Buildkite.

Topik

- Self-hosted GitHub Aksi pelari di AWS CodeBuild
- Self-managed GitLab pelari di AWS CodeBuild
- Self-managed Pelari buildkite di AWS CodeBuild

Self-hosted GitHub Aksi pelari di AWS CodeBuild

Anda dapat mengonfigurasi project untuk menyiapkan runner GitHub Actions yang dihosting sendiri dalam CodeBuild container untuk memproses pekerjaan alur kerja GitHub Actions. Hal ini dapat dilakukan dengan menyiapkan webhook menggunakan CodeBuild project Anda, dan memperbarui alur kerja GitHub Actions YALL untuk menggunakan runner yang di-host sendiri yang dihosting di mesin. CodeBuild

Langkah-langkah tingkat tinggi untuk mengonfigurasi CodeBuild proyek untuk menjalankan pekerjaan GitHub Actions adalah sebagai berikut:

1. Jika Anda belum melakukannya, buat token akses pribadi atau sambungkan dengan aplikasi OAuth untuk menghubungkan proyek Anda. GitHub
2. Arahkan ke CodeBuild konsol dan buat CodeBuild proyek dengan webhook dan atur filter webhook Anda.
3. Perbarui alur kerja GitHub Actions YAMM GitHub untuk mengonfigurasi lingkungan build Anda.

Untuk prosedur yang lebih rinci, lihatTutorial: Konfigurasikan pelari CodeBuild-hosted GitHub Tindakan.

Fitur ini memungkinkan pekerjaan alur kerja GitHub Actions Anda untuk mendapatkan integrasi asli AWS, yang memberikan keamanan dan kenyamanan melalui fitur seperti IAM, AWS Secrets Manager integrasi AWS CloudTrail, dan Amazon VPC. Anda dapat mengakses jenis instans terbaru, termasuk ARM-based instance.

Topik

- Tentang CodeBuild-hosted GitHub Actions runner
- Tutorial: Konfigurasikan pelari CodeBuild-hosted GitHub Tindakan
- Memecahkan masalah webhook
- Penggantian label didukung dengan pelari Actions CodeBuild-hosted GitHub
- Hitung gambar yang didukung dengan runner CodeBuild-hosted GitHub Actions

Tentang CodeBuild-hosted GitHub Actions runner

Berikut ini adalah beberapa pertanyaan umum tentang CodeBuild-hosted GitHub Actions runner.

Kapan saya harus menyertakan penggantian gambar dan instance di label?

Anda dapat menyertakan penggantian gambar dan instance dalam label untuk menentukan lingkungan build yang berbeda untuk setiap pekerjaan alur kerja GitHub Actions Anda. Ini dapat dilakukan tanpa perlu membuat beberapa CodeBuild proyek atau webhook. Misalnya, ini berguna ketika Anda perlu menggunakan matriks untuk pekerjaan alur kerja Anda.

```
name: Hello World
on: [push]
jobs:
  Hello-World-Job:
    runs-on:
```

```
      - codebuild-myProject-${{ github.run_id }}-${{ github.run_attempt }}
        image:${{ matrix.os }}
        instance-size:${{ matrix.size }}
    strategy:
      matrix:
        include:
          - os: arm-3.0
            size: small
          - os: linux-5.0
            size: large
    steps:
      - run: echo "Hello World!"
```

> **Note**
>
> Tanda kutip mungkin diperlukan jika `runs-on` memiliki beberapa label yang berisi konteks GitHub Tindakan.

Dapatkah saya menggunakan CloudFormation untuk fitur ini?

Ya, Anda dapat menyertakan grup filter dalam CloudFormation template Anda yang menentukan filter peristiwa pekerjaan alur kerja GitHub Actions di webhook proyek Anda.

```
Triggers:
  Webhook: true
  FilterGroups:
    - - Type: EVENT
        Pattern: WORKFLOW_JOB_QUEUED
```

Untuk informasi selengkapnya, lihat Filter acara GitHub webhook (CloudFormation).

Jika Anda memerlukan bantuan untuk menyiapkan kredensil proyek di CloudFormation templat Anda, lihat AWS::CodeBuild:: SourceCredential di Panduan AWS CloudFormation Pengguna untuk informasi selengkapnya.

Bagaimana saya bisa menutupi rahasia saat menggunakan fitur ini?

Secara default, rahasia yang dicetak di log tidak ditutupi. Jika Anda ingin menutupi rahasia Anda, Anda dapat menggunakan sintaks berikut: `::add-mask::`*`value`*. Berikut ini adalah contoh bagaimana Anda dapat menggunakan sintaks ini di YAMAL Anda:

```
name: Secret Job
on: [push]
jobs:
  Secret-Job:
    runs-on: codebuild-myProject-${{ github.run_id }}-${{ github.run_attempt }}
    env:
      SECRET_NAME: "secret-name"
    steps:
      - run: echo "::add-mask::$SECRET_NAME"
```

Untuk informasi selengkapnya, lihat Menyembunyikan nilai dalam log on GitHub.

Dapatkah saya menerima peristiwa webhook GitHub Actions dari beberapa repositori dalam satu proyek?

CodeBuild mendukung organisasi dan webhook tingkat global, yang menerima acara dari organisasi atau perusahaan tertentu. Untuk informasi selengkapnya, lihat GitHub webhook global dan organisasi.

Wilayah mana yang mendukung penggunaan runner CodeBuild-hosted GitHub Actions?

CodeBuild-hosted GitHub Tindakan pelari didukung di semua CodeBuild wilayah. Untuk informasi selengkapnya tentang Wilayah AWS tempat CodeBuild tersedia, lihat AWS Layanan menurut Wilayah.

Platform mana yang mendukung penggunaan pelari CodeBuild-hosted GitHub Actions?

CodeBuild-hosted GitHub Pelari tindakan didukung di Amazon EC2 AWS Lambdadan komputasi. Anda dapat menggunakan platform berikut: Amazon Linux 2, Amazon Linux 2023, Ubuntu, dan Windows Server Core 2019. Untuk informasi selengkapnya, lihat Gambar komputasi EC2 dan Gambar komputasi Lambda.

Tutorial: Konfigurasikan pelari CodeBuild-hosted GitHub Tindakan

Tutorial ini menunjukkan cara mengkonfigurasi CodeBuild proyek Anda untuk menjalankan pekerjaan GitHub Actions. Untuk informasi selengkapnya tentang menggunakan GitHub Tindakan dengan CodeBuild lihatTutorial: Konfigurasikan pelari CodeBuild-hosted GitHub Tindakan.

Untuk menyelesaikan tutorial ini, Anda harus terlebih dahulu:

- Terhubung dengan token akses pribadi, rahasia Secrets Manager, aplikasi OAuth, atau GitHub Aplikasi. Jika Anda ingin terhubung dengan aplikasi OAuth, Anda harus menggunakan CodeBuild

konsol untuk melakukannya. Jika Anda ingin membuat token akses pribadi, Anda dapat menggunakan CodeBuild konsol atau menggunakan ImportSourceCredentials API. Untuk instruksi lebih lanjut, lihatGitHub dan akses Server GitHub Perusahaan di CodeBuild.

- Connect CodeBuild ke GitHub akun Anda. Untuk melakukannya, Anda dapat melakukan salah satu dari yang berikut:
  - Anda dapat menambahkan GitHub sebagai penyedia sumber di konsol. Anda dapat terhubung dengan token akses pribadi, rahasia Secrets Manager, aplikasi OAuth, atau GitHub App. Untuk petunjuk, lihat GitHub dan akses Server GitHub Perusahaan di CodeBuild.
  - Anda dapat mengimpor GitHub kredensil Anda melalui API. ImportSourceCredentials Ini hanya dapat dilakukan dengan token akses pribadi. Jika Anda terhubung menggunakan aplikasi OAuth, Anda harus terhubung menggunakan konsol sebagai gantinya. Untuk petunjuk, lihat Connect GitHub dengan token akses (CLI).

> **Note**
>
> Ini hanya perlu dilakukan jika Anda belum terhubung ke GitHub akun Anda.

Langkah 1: Buat CodeBuild proyek dengan webhook

Pada langkah ini, Anda akan membuat CodeBuild proyek dengan webhook dan meninjaunya di GitHub konsol. Anda juga dapat memilih GitHub Enterprise sebagai penyedia sumber Anda. Untuk mempelajari selengkapnya tentang membuat webhook dalam GitHub Enterprise, lihatGitHub webhook manual.

Untuk membuat CodeBuild proyek dengan webhook

1. Buka AWS CodeBuild konsol di https://console.aws.amazon.com/codesuite/codebuild/home.
2. Buat proyek build. Untuk informasi selengkapnya, lihat Buat proyek build (konsol) dan Jalankan build (konsol).
3. Pada tipe Project, pilih Runner project.

   Di Runner:

   a. Untuk penyedia Runner, pilih GitHub.

   b. Untuk lokasi Runner, pilih Repository.

c. Untuk URL Repositori di bawah Repositori, pilih. https://github.com/user-name/repository-name

> **Note**
>
> Secara default, proyek Anda hanya akan menerima `WORKFLOW_JOB_QUEUED` peristiwa untuk satu repositori. Jika Anda ingin menerima acara untuk semua repositori dalam organisasi atau perusahaan, lihat. GitHub webhook global dan organisasi

4. • Di Lingkungan:
   - Pilih gambar Lingkungan yang didukung dan Compute. Perhatikan bahwa Anda memiliki opsi untuk mengganti setelan gambar dan instans dengan menggunakan label di alur kerja GitHub Actions YAMP. Untuk informasi selengkapnya, lihat Langkah 2: Perbarui alur kerja GitHub Tindakan Anda YAMB
   - Di Buildspec:
     - Perhatikan bahwa buildspec Anda akan diabaikan kecuali `buildspec-override:true` ditambahkan sebagai label. Sebagai gantinya, CodeBuild akan menimpanya untuk menggunakan perintah yang akan mengatur pelari yang dihosting sendiri.
5. Lanjutkan dengan nilai default dan kemudian pilih Create build project.
6. Buka GitHub konsol di `https://github.com/`*`user-name`*`/`*`repository-name`*`/settings/hooks` untuk memverifikasi bahwa webhook telah dibuat dan diaktifkan untuk mengirimkan peristiwa pekerjaan Alur Kerja.

Langkah 2: Perbarui alur kerja GitHub Tindakan Anda YAMB

Pada langkah ini, Anda akan memperbarui file YAMAL alur kerja GitHub Actions `GitHub`untuk mengonfigurasi lingkungan build dan menggunakan runner yang dihosting sendiri GitHub Actions di. CodeBuild Untuk informasi selengkapnya, lihat Menggunakan label dengan pelari yang dihosting sendiri dan. Penggantian label didukung dengan pelari Actions CodeBuild-hosted GitHub

Memperbarui alur kerja GitHub Actions YAMM

Arahkan ke `GitHub`dan perbarui `runs-on`setelan di alur kerja GitHub Tindakan YAMM untuk mengonfigurasi lingkungan build Anda. Untuk melakukannya, Anda dapat melakukan salah satu dari yang berikut:

- Anda dapat menentukan nama proyek dan menjalankan ID, dalam hal ini build akan menggunakan konfigurasi proyek yang ada untuk komputasi, gambar, versi gambar, dan ukuran instance. Nama proyek diperlukan untuk menautkan pengaturan AWS terkait pekerjaan GitHub Tindakan Anda ke CodeBuild proyek tertentu. Dengan memasukkan nama proyek di YAMAL, CodeBuild diizinkan untuk memanggil pekerjaan dengan pengaturan proyek yang benar. Dengan memberikan ID run, CodeBuild memetakan build Anda ke alur kerja tertentu yang berjalan dan menghentikan build saat alur kerja dijalankan dibatalkan. Untuk informasi lebih lanjut, lihat `github`konteks.

```
runs-on: codebuild-<project-name>-${{ github.run_id }}-${{ github.run_attempt }}
```

> **Note**
>
> Pastikan bahwa Anda *<project-name>* cocok dengan nama proyek yang Anda buat pada langkah sebelumnya. Jika tidak cocok, tidak CodeBuild akan memproses webhook dan alur kerja GitHub Actions mungkin hang.

Berikut ini adalah contoh alur kerja GitHub Actions YAMAL:

```
name: Hello World
on: [push]
jobs:
  Hello-World-Job:
    runs-on:
      - codebuild-myProject-${{ github.run_id }}-${{ github.run_attempt }}
    steps:
      - run: echo "Hello World!"
```

- Anda juga dapat mengganti gambar dan jenis komputasi di label. Lihat Hitung gambar yang didukung dengan runner CodeBuild-hosted GitHub Actions untuk daftar gambar yang dikuratori. Untuk menggunakan gambar kustom, lihatPenggantian label didukung dengan pelari Actions CodeBuild-hosted GitHub. Jenis komputasi dan gambar dalam label akan mengganti pengaturan lingkungan pada proyek Anda. Untuk mengganti setelan lingkungan untuk build komputasi CodeBuild EC2 atau Lambda, gunakan sintaks berikut:

```
runs-on:
  - codebuild-<project-name>-${{ github.run_id }}-${{ github.run_attempt }}
    image:<environment-type>-<image-identifier>
```

```
    instance-size:<instance-size>
```

Berikut ini adalah contoh alur kerja GitHub Actions YAMAL:

```
name: Hello World
on: [push]
jobs:
  Hello-World-Job:
    runs-on:
      - codebuild-myProject-${{ github.run_id }}-${{ github.run_attempt }}
        image:arm-3.0
        instance-size:small
    steps:
      - run: echo "Hello World!"
```

- Anda dapat mengganti armada yang digunakan untuk build Anda di label. Ini akan menggantikan pengaturan armada yang dikonfigurasi pada proyek Anda untuk menggunakan armada yang ditentukan. Untuk informasi selengkapnya, lihat Jalankan build pada armada kapasitas yang dicadangkan. Untuk mengganti setelan armada untuk build komputasi Amazon EC2, gunakan sintaks berikut:

```
runs-on:
  - codebuild-<project-name>-${{ github.run_id }}-${{ github.run_attempt }}
    fleet:<fleet-name>
```

Untuk mengganti armada dan gambar yang digunakan untuk build, gunakan sintaks berikut:

```
runs-on:
  - codebuild-<project-name>-${{ github.run_id }}-${{ github.run_attempt }}
    fleet:<fleet-name>
    image:<environment-type>-<image-identifier>
```

Berikut ini adalah contoh alur kerja GitHub Actions YAMAL:

```
name: Hello World
on: [push]
jobs:
  Hello-World-Job:
    runs-on:
      - codebuild-myProject-${{ github.run_id }}-${{ github.run_attempt }}
        fleet:myFleet
```

```
        image:arm-3.0
    steps:
      - run: echo "Hello World!"
```

- Untuk menjalankan pekerjaan GitHub Actions pada gambar kustom, Anda dapat mengonfigurasi gambar kustom dalam CodeBuild proyek Anda dan menghindari memberikan label penggantian gambar. CodeBuild akan menggunakan gambar yang dikonfigurasi dalam proyek jika tidak ada label penggantian gambar yang disediakan.
- Secara opsional, Anda dapat memberikan label di luar label yang CodeBuild mendukung. Label ini akan diabaikan untuk tujuan mengesampingkan atribut build, tetapi tidak akan gagal dalam permintaan webhook. Misalnya, menambahkan `testLabel` sebagai label tidak akan mencegah build berjalan.

> **Note**
>
> Jika dependensi yang disediakan oleh GitHub-hosted pelari tidak tersedia di CodeBuild lingkungan, Anda dapat menginstal dependensi menggunakan GitHub Tindakan dalam proses alur kerja Anda. Misalnya, Anda dapat menggunakan `setup-python`tindakan untuk menginstal Python untuk lingkungan build Anda.

Jalankan perintah buildspec di fase INSTALL, PRE_BUILD, dan POST_BUILD

Secara default, CodeBuild abaikan perintah buildspec apa pun saat menjalankan build Actions yang dihosting sendiri. GitHub Untuk menjalankan perintah buildspec selama build, `buildspec-override:true` dapat ditambahkan sebagai akhiran label:

```
runs-on:
  - codebuild-<project-name>-${{ github.run_id }}-${{ github.run_attempt }}
    buildspec-override:true
```

Dengan menggunakan perintah ini, CodeBuild akan membuat folder yang disebut `actions-runner` dalam folder sumber utama wadah. Ketika GitHub Actions runner dimulai selama `BUILD` fase, runner akan berjalan di direktori. `actions-runner`

Ada beberapa batasan saat menggunakan penggantian buildspec dalam build Tindakan yang dihosting sendiri: GitHub

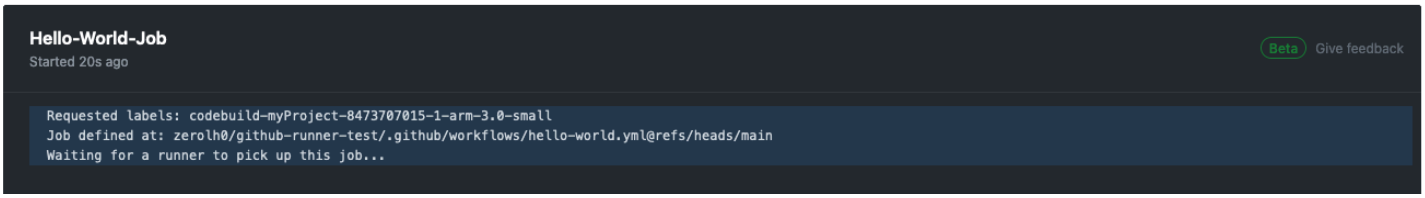
- CodeBuild tidak akan menjalankan perintah buildspec selama `BUILD` fase, karena pelari yang di-host sendiri berjalan dalam fase tersebut. `BUILD`
- CodeBuild tidak akan mengunduh sumber primer atau sekunder selama `DOWNLOAD_SOURCE` fase. Jika Anda memiliki file buildspec yang dikonfigurasi, hanya file itu yang akan diunduh dari sumber utama proyek.
- Jika perintah build gagal di `INSTALL` fase `PRE_BUILD` atau, tidak CodeBuild akan memulai runner yang dihosting sendiri dan pekerjaan alur kerja GitHub Actions harus dibatalkan secara manual.
- CodeBuild mengambil token runner selama `DOWNLOAD_SOURCE` fase, yang memiliki waktu kedaluwarsa satu jam. Jika `INSTALL` fase `PRE_BUILD` atau fase Anda melebihi satu jam, token runner mungkin kedaluwarsa sebelum runner yang GitHub di-host sendiri dimulai.

Langkah 3: Tinjau hasil Anda

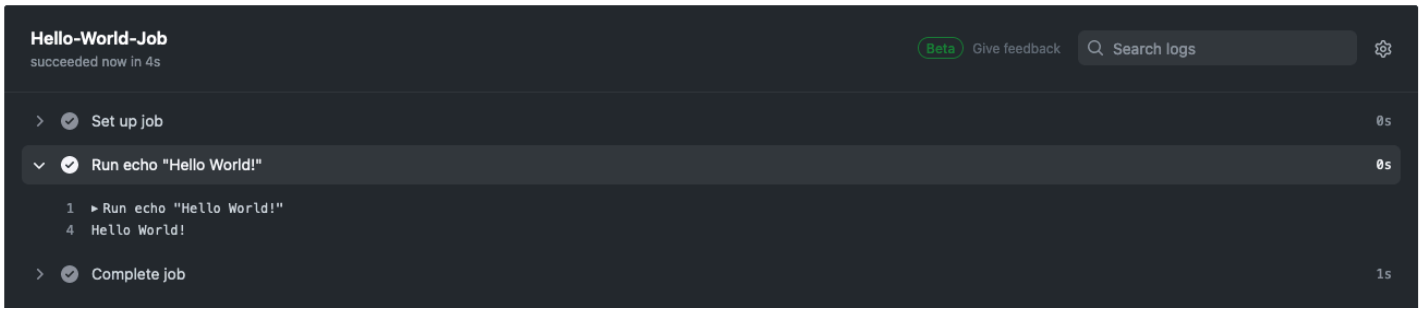
Setiap kali alur kerja GitHub Tindakan dijalankan, CodeBuild akan menerima peristiwa pekerjaan alur kerja melalui webhook. Untuk setiap pekerjaan dalam alur kerja, CodeBuild mulai build untuk menjalankan runner Actions sesaat. GitHub Pelari bertanggung jawab untuk mengeksekusi satu pekerjaan alur kerja. Setelah pekerjaan selesai, runner dan proses build terkait akan segera dihentikan.

Untuk melihat log pekerjaan alur kerja, navigasikan ke repositori GitHub, pilih Tindakan, pilih alur kerja yang diinginkan, lalu pilih Job spesifik yang ingin Anda tinjau log.

Anda dapat meninjau label yang diminta di log saat pekerjaan menunggu untuk diambil oleh pelari yang di-host sendiri. CodeBuild



Setelah pekerjaan selesai, Anda akan dapat melihat log pekerjaan.



GitHub Opsi konfigurasi pelari tindakan

Anda dapat menentukan variabel lingkungan berikut dalam konfigurasi proyek Anda untuk mengubah konfigurasi penyiapan pelari yang dihosting sendiri.

`CODEBUILD_CONFIG_GITHUB_ACTIONS_ORG_REGISTRATION_NAME`

CodeBuild akan mendaftarkan pelari yang dihosting sendiri ke nama organisasi yang ditentukan sebagai nilai variabel lingkungan ini. Untuk informasi selengkapnya tentang mendaftarkan pelari di tingkat organisasi dan izin yang diperlukan, lihat Membuat konfigurasi untuk runner tepat waktu untuk organisasi.

`CODEBUILD_CONFIG_GITHUB_ACTIONS_ENTERPRISE_REGISTRATION_NAME`

CodeBuild akan mendaftarkan pelari yang dihosting sendiri ke nama perusahaan yang ditentukan sebagai nilai variabel lingkungan ini. Untuk informasi selengkapnya tentang mendaftarkan pelari di tingkat perusahaan dan izin yang diperlukan, lihat Membuat konfigurasi untuk runner just-in-time untuk Perusahaan.

> **Note**
>
> Pelari perusahaan tidak tersedia untuk repositori organisasi secara default. Agar pelari yang dihosting sendiri dapat mengambil pekerjaan alur kerja, Anda mungkin perlu mengonfigurasi pengaturan akses grup pelari Anda. Untuk informasi selengkapnya, lihat Membuat pelari perusahaan tersedia untuk repositori.

`CODEBUILD_CONFIG_GITHUB_ACTIONS_RUNNER_GROUP_ID`

CodeBuild akan mendaftarkan pelari yang dihosting sendiri ke ID grup pelari integer yang disimpan sebagai nilai variabel lingkungan ini. Secara default, nilai ini adalah 1. Untuk informasi selengkapnya tentang grup pelari yang dihosting sendiri, lihat Mengelola akses ke pelari yang dihosting sendiri menggunakan grup.

`CODEBUILD_CONFIG_GITHUB_ACTIONS_ORG_REGISTRATION_NAME`

Untuk mengonfigurasi pendaftaran runner tingkat organisasi menggunakan file GitHub YAMM alur kerja Actions, Anda dapat menggunakan sintaks berikut:

```
name: Hello World
```

```
on: [push]
jobs:
  Hello-World-Job:
    runs-on:
      - codebuild-myProject-${{ github.run_id }}-${{ github.run_attempt }}
        organization-registration-name:myOrganization
    steps:
      - run: echo "Hello World!"
```

`CODEBUILD_CONFIG_GITHUB_ACTIONS_ENTERPRISE_REGISTRATION_NAME`

Untuk mengonfigurasi pendaftaran runner tingkat perusahaan menggunakan file GitHub YAMM alur kerja Actions, Anda dapat menggunakan sintaks berikut:

```
name: Hello World
on: [push]
jobs:
  Hello-World-Job:
    runs-on:
      - codebuild-myProject-${{ github.run_id }}-${{ github.run_attempt }}
        enterprise-registration-name:myEnterprise
    steps:
      - run: echo "Hello World!"
```

`CODEBUILD_CONFIG_GITHUB_ACTIONS_RUNNER_GROUP_ID`

Untuk mengonfigurasi registrasi runner ke ID grup runner tertentu menggunakan file YAMM alur kerja GitHub Actions, Anda dapat menggunakan sintaks berikut:

```
name: Hello World
on: [push]
jobs:
  Hello-World-Job:
    runs-on:
      - codebuild-myProject-${{ github.run_id }}-${{ github.run_attempt }}
        registration-group-id:3
    steps:
      - run: echo "Hello World!"
```

Filter GitHub Tindakan acara webhook (CloudFormation)

YAML-formatted Bagian berikut dari CloudFormation template membuat grup filter yang memicu build saat dievaluasi ke true. Grup filter berikut menentukan permintaan pekerjaan alur kerja GitHub Tindakan dengan nama alur kerja yang cocok dengan ekspresi reguler. `\[CI-CodeBuild\]`

```
CodeBuildProject:
  Type: AWS::CodeBuild::Project
  Properties:
    Name: MyProject
    ServiceRole: service-role
    Artifacts:
      Type: NO_ARTIFACTS
    Environment:
      Type: LINUX_CONTAINER
      ComputeType: BUILD_GENERAL1_SMALL
      Image: aws/codebuild/standard:5.0
    Source:
      Type: GITHUB
      Location: CODEBUILD_DEFAULT_WEBHOOK_SOURCE_LOCATION
    Triggers:
      Webhook: true
      ScopeConfiguration:
        Name: organization-name
        Scope: GITHUB_ORGANIZATION
      FilterGroups:
        - - Type: EVENT
            Pattern: WORKFLOW_JOB_QUEUED
          - Type: WORKFLOW_NAME
            Pattern: \[CI-CodeBuild\]
```

Filter GitHub Tindakan acara webhook (AWS CDK)

AWS CDK Template berikut membuat grup filter yang memicu build saat dievaluasi ke true. Grup filter berikut menentukan permintaan pekerjaan alur kerja GitHub Tindakan.

```
import { aws_codebuild as codebuild } from 'aws-cdk-lib';
import {EventAction, FilterGroup} from "aws-cdk-lib/aws-codebuild";

const source = codebuild.Source.gitHub({
      owner: 'owner',
      repo: 'repo',
      webhook: true,
```

```
      webhookFilters: [FilterGroup.inEventOf(EventAction.WORKFLOW_JOB_QUEUED)],
    })
```

Filter GitHub Tindakan acara webhook (Terraform)

Template Terraform berikut membuat grup filter yang memicu build saat dievaluasi ke true. Grup filter berikut menentukan permintaan pekerjaan alur kerja GitHub Tindakan.

```
resource "aws_codebuild_webhook" "example" {
  project_name = aws_codebuild_project.example.name
  build_type   = "BUILD"
  filter_group {
    filter {
      type    = "EVENT"
      pattern = "WORKFLOW_JOB_QUEUED"
    }
  }
}
```

Filter GitHub Tindakan acara webhook (AWS CLI)

AWS CLI Perintah berikut membuat project runner GitHub Actions yang dihosting sendiri dengan grup filter permintaan pekerjaan alur kerja GitHub Actions yang memicu build saat dievaluasi ke true.

```
aws codebuild create-project \
--name <project name> \
--source "{\"type\":\"GITHUB\",\"location\":\"<repository location>\",\"buildspec\":
\"\"}" \
--artifacts {"\"type\":\"NO_ARTIFACTS\""} \
--environment "{\"type\": \"LINUX_CONTAINER\",\"image\": \"aws/codebuild/amazonlinux-
x86_64-standard:5.0\",\"computeType\": \"BUILD_GENERAL1_MEDIUM\"}" \
--service-role "<service role ARN>"
```

```
aws codebuild create-webhook \
--project-name <project name> \
--filter-groups "[[{\"type\":\"EVENT\",\"pattern\":\"WORKFLOW_JOB_QUEUED\"}]]"
```

Memecahkan masalah webhook

Masalah: Webhook yang Anda atur Tutorial: Konfigurasikan pelari CodeBuild-hosted GitHub Tindakan tidak berfungsi atau pekerjaan alur kerja Anda terus berlanjut. GitHub

Kemungkinan penyebabnya:

- Acara lowongan kerja alur kerja webhook Anda mungkin gagal memicu build. Tinjau log Respons untuk melihat respons atau pesan kesalahan.
- Pekerjaan Anda ditugaskan ke agen pelari yang salah karena konfigurasi labelnya. Masalah ini dapat terjadi ketika salah satu pekerjaan Anda dalam satu alur kerja memiliki label yang lebih sedikit daripada pekerjaan lain. Misalnya, jika Anda memiliki dua pekerjaan dengan label berikut dalam alur kerja yang sama, jalankan:
  - Job 1: `codebuild-myProject-${{ github.run_id }}-${{ github.run_attempt }}`
  - Job 2:`codebuild-myProject-${{ github.run_id }}-${{ github.run_attempt }}`, `instance-size:medium`

  Saat merutekan pekerjaan GitHub Tindakan yang dihosting sendiri, GitHub akan merutekan pekerjaan ke pelari mana pun dengan semua label yang ditentukan pekerjaan. Perilaku ini berarti bahwa Job 1 dapat diambil oleh pelari yang dibuat untuk Job 1 atau Job 2, tetapi Job 2 hanya dapat diambil oleh pelari yang dibuat untuk Job 2 karena memiliki label tambahan. Jika Job 1 diambil oleh pelari yang dibuat untuk Job 2, maka Job 2 akan macet karena pelari Job 1 tidak memiliki label. `instance-size:medium`

Solusi yang direkomendasikan:

Saat membuat beberapa pekerjaan dalam alur kerja yang sama, gunakan jumlah penggantian label yang sama untuk setiap pekerjaan atau tetapkan setiap lowongan label khusus, seperti atau. `job1` `job2`

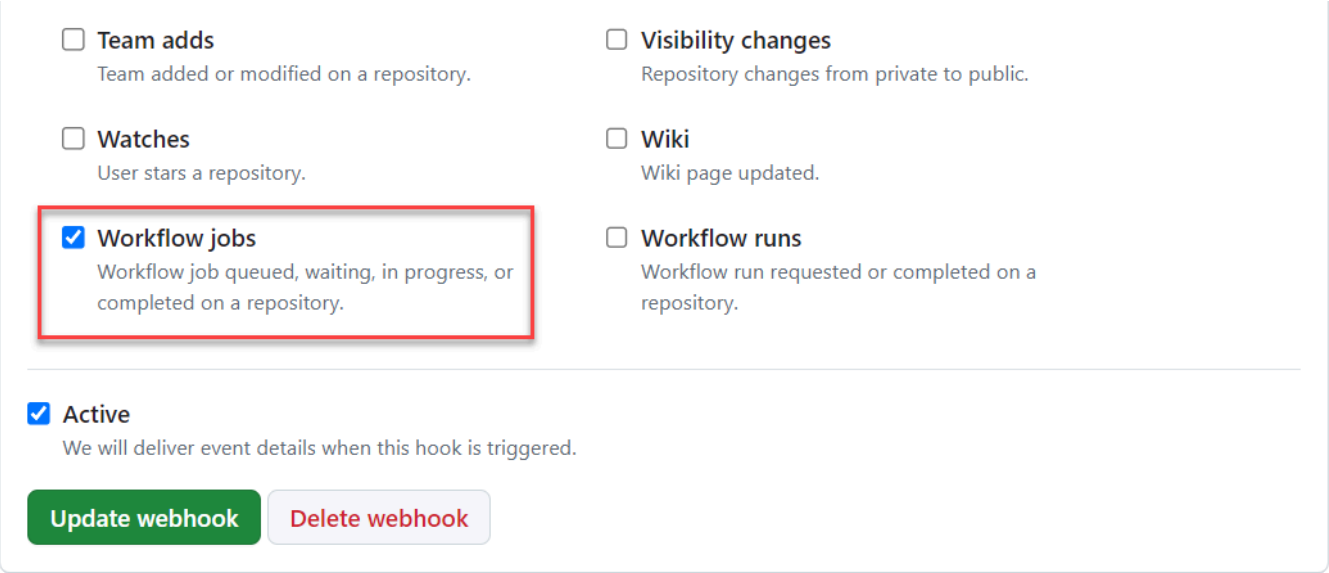
Menetapkan setiap pekerjaan label kustom uniknya sendiri adalah pilihan yang paling dapat diandalkan. Ini memastikan bahwa tidak ada set label pekerjaan yang merupakan bagian dari pekerjaan lain. Ketika setiap pekerjaan membawa label yang tidak dimiliki pekerjaan lain, GitHub mencocokkan setiap pekerjaan hanya dengan pelari yang dibuat untuknya, terlepas dari berapa banyak penggantian lain yang digunakan setiap pekerjaan. Dalam alur kerja berikut, setiap pekerjaan memiliki label unik (`job1`dan`job2`). Meskipun Job 2 memiliki `instance-size:medium` penggantian tambahan, Job 1 tidak dapat lagi dicocokkan dengan pelari yang dibuat untuk Job 2, jadi tidak ada pekerjaan yang macet:

```
name: Hello World
on: [push]
jobs:
  job1:
```

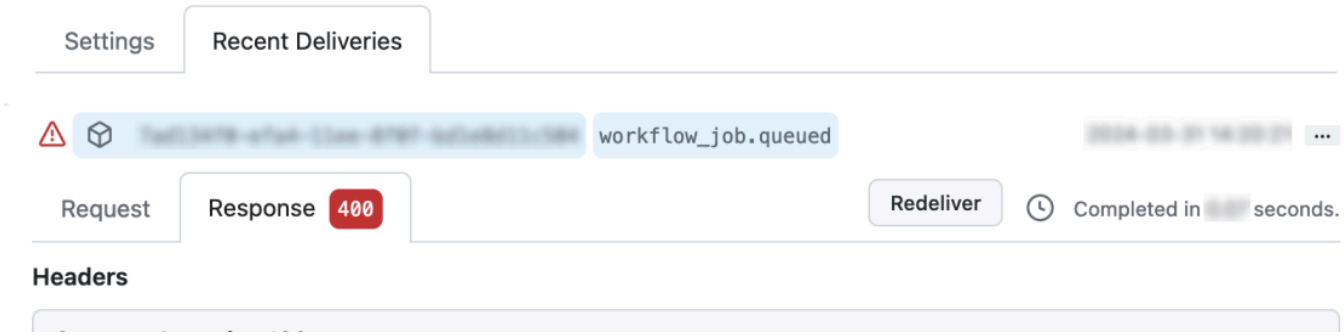
```
    runs-on:
      - codebuild-myProject-${{ github.run_id }}-${{ github.run_attempt }}
      - job1
    steps:
      - run: echo "Hello from Job 1!"
  job2:
    runs-on:
      - codebuild-myProject-${{ github.run_id }}-${{ github.run_attempt }}
      - instance-size:medium
      - job2
    steps:
      - run: echo "Hello from Job 2!"
```

Jika kesalahan berlanjut, gunakan instruksi berikut untuk men-debug masalah.

1. Buka GitHub konsol di `https://github.com/`*`user-name`*`/`*`repository-name`*`/settings/hooks` untuk melihat pengaturan webhook repositori Anda. Di halaman ini, Anda akan melihat webhook yang dibuat untuk repositori Anda.
2. Pilih Edit dan konfirmasikan bahwa webhook diaktifkan untuk mengirimkan peristiwa pekerjaan Alur Kerja.



3. Arahkan ke tab Pengiriman Terbaru, temukan `workflow_job.queued` acara yang sesuai, dan perluas acara.
4. Tinjau bidang label di Payload dan pastikan itu seperti yang diharapkan.
5. Terakhir, tinjau tab Respons, karena ini berisi respons atau pesan kesalahan yang dikembalikan CodeBuild.



6. Atau, Anda dapat men-debug kegagalan webhook menggunakan API GitHub. Anda dapat melihat pengiriman terbaru untuk webhook menggunakan pengiriman Daftar untuk API webhook repositori:

```
gh api \
  -H "Accept: application/vnd.github+json" \
  -H "X-GitHub-Api-Version: 2022-11-28" \
  /repos/owner/repo/hooks/hook-id/deliveries
```

Setelah menemukan pengiriman webhook yang ingin Anda debug dan mencatat ID pengiriman, Anda dapat menggunakan Get a delivery for a repository webhook API. CodeBuildTanggapan terhadap muatan pengiriman webhook dapat ditemukan di bagian`response`:

```
gh api \
  -H "Accept: application/vnd.github+json" \
  -H "X-GitHub-Api-Version: 2022-11-28" \
  /repos/owner/repo/hooks/hook-id/deliveries/delivery-id
```

Masalah: GitHub Tindakan Anda dengan aturan perlindungan penerapan mengaktifkan pemicu build di dalamnya CodeBuild sebelum penerapan disetujui.

Kemungkinan penyebabnya: CodeBuild mengambil penerapan dan lingkungan yang terkait dengan pekerjaan GitHub Tindakan jika ada untuk memverifikasi apakah penerapan disetujui. Jika CodeBuild gagal mengambil penerapan atau lingkungan, CodeBuild build dapat dipicu sebelum waktunya.

Solusi yang disarankan: Verifikasi bahwa kredensil yang terkait dengan CodeBuild proyek Anda telah membaca izin untuk penerapan dan tindakan di dalamnya. GitHub

Penggantian label didukung dengan pelari Actions CodeBuild-hosted GitHub

Dalam alur kerja GitHub Actions YAMAL, Anda dapat memberikan berbagai penggantian label yang mengubah build runner yang dihosting sendiri. Setiap build yang tidak dikenali oleh CodeBuild akan diabaikan tetapi tidak akan gagal permintaan webhook Anda. Alur kerja berikut YAMAL mendefinisikan dua pekerjaan: `job1` menggunakan penggantian untuk image, ukuran instance, armada, dan buildspec, dan hanya menggunakan penggantian gambar. `job2` Berikan setiap pekerjaan label unik (`job1`dan`job2`) sehingga GitHub merutekan setiap pekerjaan ke pelari yang dibuat untuknya:

```
name: Hello World
on: [push]
jobs:
  job1:
    runs-on:
      - codebuild-myProject-${{ github.run_id }}-${{ github.run_attempt }}
      - image:arm-3.0
      - instance-size:small
      - fleet:myFleet
      - buildspec-override:true
      - job1
    steps:
      - run: echo "Hello from Job 1!"
  job2:
    runs-on:
      - codebuild-myProject-${{ github.run_id }}-${{ github.run_attempt }}
      - image:linux-5.0
      - job2
    steps:
      - run: echo "Hello from Job 2!"
```

Anda juga dapat menggunakan strategi matriks untuk menjalankan satu pekerjaan di beberapa konfigurasi. Perluasan pekerjaan matriks tunggal selalu memiliki jumlah label yang sama, sehingga masalah pencocokan runner tidak mempengaruhinya. Namun, jika alur kerja berisi lebih dari satu tugas matriks, berikan setiap pekerjaan matriks label kustom uniknya sendiri (misalnya, `matrix-job-1` dan`matrix-job-2`) sehingga GitHub tidak cocok dengan pekerjaan dengan pelari yang dibuat untuk pekerjaan matriks yang berbeda:

```
name: Hello World
on: [push]
jobs:
```

```
  Hello-World-Job:
    runs-on:
      - codebuild-myProject-${{ github.run_id }}-${{ github.run_attempt }}
      - image:${{ matrix.os }}
      - instance-size:${{ matrix.size }}
      - fleet:myFleet
      - buildspec-override:true
      - matrix-job-1
    strategy:
      matrix:
        include:
          - os: arm-3.0
            size: small
          - os: linux-5.0
            size: large
    steps:
      - run: echo "Hello World!"
```

> **Note**
>
> Jika pekerjaan alur kerja Anda tergantung GitHub, lihat Memecahkan masalah webhook dan Menggunakan label khusus untuk merutekan pekerjaan.

`codebuild-<project-name>-${{github.run_id}}-${{github.run_attempt}}` (wajib)

- Contoh: `codebuild-fake-project-${{ github.run_id }}-${{ github.run_attempt }}`
- Diperlukan untuk semua alur kerja GitHub Tindakan YAML. *<project name>*harus sama dengan nama proyek tempat webhook pelari yang dihosting sendiri dikonfigurasi.

`image:<environment-type>-<image-identifier>`

- Contoh: `image:arm-3.0`
- Mengganti jenis gambar dan lingkungan yang digunakan saat memulai build runner yang dihosting sendiri dengan gambar yang dikurasi. Untuk mempelajari nilai yang didukung, lihatHitung gambar yang didukung dengan runner CodeBuild-hosted GitHub Actions.
  - Untuk mengganti jenis gambar dan lingkungan yang digunakan dengan gambar khusus, gunakan `image:custom-<environment-type>-<custom-image-identifier>`

- Contoh: `image:custom-arm-public.ecr.aws/codebuild/amazonlinux-aarch64-standard:3.0`

> **Note**
>
> Jika gambar kustom berada di registri pribadi, lihatKonfigurasikan kredenal registri pribadi untuk pelari yang dihosting sendiri.

`instance-size:<instance-size>`

- Contoh: `instance-size:medium`
- Mengganti jenis instance yang digunakan saat memulai build runner yang dihosting sendiri. Untuk mempelajari nilai yang didukung, lihatHitung gambar yang didukung dengan runner CodeBuild-hosted GitHub Actions.

`fleet:<fleet-name>`

- Contoh: `fleet:myFleet`
- Mengganti pengaturan armada yang dikonfigurasi pada proyek Anda untuk menggunakan armada yang ditentukan. Untuk informasi selengkapnya, lihat Jalankan build pada armada kapasitas yang dicadangkan.

`buildspec-override:<boolean>`

- Contoh: `buildspec-override:true`
- Mengizinkan build menjalankan perintah buildspec di`INSTALL`,`PRE_BUILD`, dan `POST_BUILD` fase jika disetel ke. `true`

`<unique-label>`(direkomendasikan ketika alur kerja menjalankan memiliki banyak pekerjaan)

- Contoh: `job1`
- Label kustom yang Anda tentukan dan tidak ada pekerjaan lain dalam alur kerja yang sama yang digunakan. Menetapkan setiap pekerjaan label unik memastikan bahwa GitHub merutekan setiap pekerjaan hanya ke pelari yang dibuat untuknya. Tanpa itu, ketika pekerjaan dalam alur kerja yang sama memiliki jumlah penggantian label yang berbeda, pekerjaan dengan label lebih

sedikit dapat diambil oleh pelari yang dibuat untuk pekerjaan dengan lebih banyak label, yang membuat pekerjaan kedua tanpa pelari. Untuk informasi selengkapnya, lihat Memecahkan masalah webhook.

Pengesampingan label tunggal

CodeBuild memungkinkan Anda memberikan beberapa penggantian dalam satu label menggunakan yang berikut:

> **Note**
>
> Karena semua penggantian digabungkan menjadi satu label, setiap pelari pekerjaan mendaftar dengan tepat satu label. Ini berarti masalah pencocokan multiple-job runner yang dijelaskan dalam Memecahkan masalah webhook tidak dapat terjadi dengan format ini, bahkan tanpa label unik pada setiap pekerjaan. Gunakan format label tunggal jika Anda lebih suka perilaku ini dan pekerjaan Anda hanya perlu penggantian yang didukungnya. Jangan menggunakannya saat Anda membutuhkan penggantian yang tidak didukungnya. Misalnya, gambar khusus (termasuk gambar dalam registri pribadi) hanya dapat ditentukan dengan menggunakan format `image:custom-`*`<environment-type>`*`-`*`<custom-image-identifier>` label. Untuk penggantian tersebut, gunakan format label terpisah yang dijelaskan sebelumnya dalam topik ini.

- Untuk mengganti setelan lingkungan Anda untuk build EC2/Lambda komputasi Amazon, gunakan sintaks berikut:

```
runs-on: codebuild-<project-name>-${{ github.run_id }}-
${{ github.run_attempt }}-<environment-type>-<image-identifier>-<instance-size>
```

- Untuk mengganti setelan armada untuk build komputasi Amazon EC2, gunakan sintaks berikut:

```
runs-on: codebuild-<project-name>-${{ github.run_id }}-${{ github.run_attempt }}-
fleet-<fleet-name>
```

- Untuk mengganti armada dan gambar yang digunakan untuk build, gunakan sintaks berikut:

```
runs-on: codebuild-<project-name>-${{ github.run_id }}-
${{ github.run_attempt }}-image-<image-version>-fleet-<fleet-name>
```

- Untuk menjalankan perintah buildspec selama build, `-with-buildspec` dapat ditambahkan sebagai akhiran label:

```
runs-on: codebuild-<project-name>-${{ github.run_id }}-
${{ github.run_attempt }}-<image>-<image-version>-<instance-size>-with-buildspec
```

- Secara opsional, Anda dapat memberikan penggantian ukuran instance tanpa mengganti gambar. Untuk build Amazon EC2, Anda dapat mengecualikan jenis lingkungan dan pengenal gambar. Untuk build Lambda, Anda dapat mengecualikan pengenal gambar.

Hitung gambar yang didukung dengan runner CodeBuild-hosted GitHub Actions

Pada label yang Anda konfigurasikanTutorial: Konfigurasikan pelari CodeBuild-hosted GitHub Tindakan, Anda dapat mengganti pengaturan lingkungan Amazon EC2 dengan menggunakan nilai di tiga kolom pertama. CodeBuild menyediakan gambar komputasi Amazon EC2 berikut. Untuk informasi lebih lanjut tentang

Tipe lingkungan	Pengidentifikasi gambar	Ukuran instans	Platform	Gambar terselesaikan	Definisi

`linux`	`4.0`	`small`
		`medium`
		`large`
		`xlarge`
			Amazon Linux 2	`aws/codebuild/amazonlinux-x86_64-standard:4.0`	al/standard/4.0

`linux`	`5.0`	`2xlarge`
		`gpu_small`
		`gpu_large`
			Amazon Linux 2023	`aws/codebuild/amazonlinux-x86_64-standard:5.0`	al/standard/5.0

`linux-ec2`	`latest`	`small` `medium`	Amazon Linux 2023	`aws/codebuild/ami/amazonlinux-x86_64`	Tidak ada

Tipe lingkungan	Pengidentifikasi gambar	Ukuran instans	Platform	Gambar terselesaikan	Definisi
		large		-base:latest	
arm	2.0	small medium large xlarge	Amazon Linux 2	aws/codebuild/amazonlinux-aarch64-standard:2.0	al/aarch64/standard/2.0
arm	3.0	2xlarge	Amazon Linux 2023	aws/codebuild/amazonlinux-aarch64-standard:3.0	al/aarch64/standard/3.0
arm-ec2	latest	small medium large	Amazon Linux 2023	aws/codebuild/ami/amazonlinux-arm-base:latest	Tidak ada

Tipe lingkungan	Pengidentifikasi gambar	Ukuran instans	Platform	Gambar terselesaikan	Definisi
ubuntu	5.0	small medium large xlarge 2xlarge gpu_small gpu_large	Ubuntu 20.04	aws/codebuild/standard:5.0	ubuntu/standard/5.0
ubuntu	6.0		Ubuntu 22.04	aws/codebuild/standard:6.0	ubuntu/standard/6.0
ubuntu	7.0		Ubuntu 22.04	aws/codebuild/standard:7.0	ubuntu/standard/7.0
windows	1.0	medium large	Inti Server Windows 2019	aws/codebuild/windows-base:2019-1.0	N/A
			Inti Server Windows 2022	aws/codebuild/windows-base:2022-1.0	N/A
windows	2.0		Inti Server Windows 2019	aws/codebuild/windows-base:2019-2.0	N/A
windows	3.0		Inti Server Windows 2019	aws/codebuild/windows-base:2019-3.0	N/A

Tipe lingkungan	Pengidentifikasi gambar	Ukuran instans	Platform	Gambar terselesaikan	Definisi
`windows-ec2`	`2022`	`medium` `large`	Inti Server Windows 2022	`aws/codebuild/ami/windows-base:2022`	Tidak ada

Selain itu, Anda dapat mengganti pengaturan lingkungan Lambda Anda dengan menggunakan nilai berikut. Untuk informasi selengkapnya tentang komputasi CodeBuild Lambda, lihat. Jalankan dibangun di AWS Lambda komputasi CodeBuild mendukung gambar komputasi Lambda berikut:

Tipe lingkungan	Pengidentifikasi gambar	Ukuran instans			

`linux-lambda`
`arm-lambda`
	`dotnet6`
	`go1.21`
	`corretto11`
	`corretto17`
	`corretto21`
	`nodejs18`
	`nodejs20`
	`python3.11`
	`python3.12`
		`1GB`
		`2GB`
		`4GB`
		`8GB`
		`10GB`			

Tipe lingkungan	Pengidentifikasi gambar	Ukuran instans			
	`ruby3.2`				

Untuk informasi selengkapnya, lihat Membangun mode dan tipe komputasi lingkungan dan Gambar Docker disediakan oleh CodeBuild.

Self-managed GitLab pelari di AWS CodeBuild

GitLab menyediakan dua mode eksekusi untuk menjalankan GitLab pekerjaan di CI/CD pipeline Anda. Salah satu mode adalah GitLab-hosted pelari, yang dikelola oleh GitLab dan terintegrasi penuh dengan GitLab. Mode lainnya adalah pelari yang dikelola sendiri, yang memungkinkan Anda membawa lingkungan khusus Anda sendiri untuk menjalankan pekerjaan dalam pipeline. GitLab CI/CD

Langkah-langkah tingkat tinggi untuk mengonfigurasi CodeBuild proyek untuk menjalankan pekerjaan GitLab CI/CD pipeline adalah sebagai berikut:

1. Jika Anda belum melakukannya, sambungkan dengan aplikasi OAuth untuk menghubungkan proyek Anda. GitLab
2. Arahkan ke CodeBuild konsol dan buat CodeBuild proyek dengan webhook dan atur filter webhook Anda.
3. Perbarui GitLab CI/CD pipeline YAMM GitLab untuk mengonfigurasi lingkungan build Anda.

Untuk prosedur yang lebih rinci, lihatTutorial: Konfigurasikan CodeBuild-hosted GitLab pelari.

Fitur ini memungkinkan pekerjaan GitLab CI/CD pipeline Anda untuk mendapatkan integrasi asli AWS, yang memberikan keamanan dan kenyamanan melalui fitur seperti IAM, AWS CloudTrail, dan Amazon VPC. Anda dapat mengakses jenis instans terbaru, termasuk ARM-based instance.

Topik

- Tentang CodeBuild-hosted GitLab pelari
- Tutorial: Konfigurasikan CodeBuild-hosted GitLab pelari
- Penggantian label didukung dengan pelari CodeBuild-hosted GitLab
- Hitung gambar yang didukung dengan pelari CodeBuild-hosted GitLab

Tentang CodeBuild-hosted GitLab pelari

Berikut ini adalah beberapa pertanyaan umum tentang CodeBuild-hosted GitLab pelari.

Jenis sumber apa yang didukung untuk CodeBuild-hosted GitLab pelari?

CodeBuild-hosted GitLab pelari didukung untuk tipe `GITLAB` dan `GITLAB_SELF_MANAGED` sumber.

Kapan saya harus menyertakan penggantian gambar dan instance di label?

Anda dapat menyertakan penggantian gambar dan instance dalam label untuk menentukan lingkungan build yang berbeda untuk setiap pekerjaan GitLab CI/CD pipeline Anda. Ini dapat dilakukan tanpa perlu membuat beberapa CodeBuild proyek atau webhook.

Dapatkah saya menggunakan CloudFormation untuk fitur ini?

Ya, Anda dapat menyertakan grup filter dalam CloudFormation template Anda yang menentukan filter peristiwa pekerjaan GitLab alur kerja di webhook proyek Anda.

```
Triggers:
  Webhook: true
  FilterGroups:
    - - Type: EVENT
        Pattern: WORKFLOW_JOB_QUEUED
```

Untuk informasi selengkapnya, lihat Filter acara GitLab webhook (CloudFormation).

Jika Anda memerlukan bantuan untuk menyiapkan kredensil proyek di CloudFormation templat Anda, lihat AWS::CodeBuild:: SourceCredential di Panduan AWS CloudFormation Pengguna untuk informasi selengkapnya.

Bagaimana saya bisa menutupi rahasia saat menggunakan fitur ini?

Secara default, rahasia yang dicetak di log tidak ditutupi. Jika Anda ingin menutupi rahasia Anda, Anda dapat melakukannya dengan memperbarui pengaturan variabel CI/CD lingkungan Anda:

Untuk menutupi rahasia di GitLab

1. Di GitLab Pengaturan Anda, pilih CI/CD.
2. Di Variabel, pilih Edit untuk rahasia yang ingin Anda tutupi.

3. Di Visibilitas, pilih Variabel topeng, lalu pilih Perbarui variabel untuk menyimpan perubahan Anda.

Dapatkah saya menerima acara GitLab webhook dari beberapa proyek dalam satu grup?

CodeBuild mendukung webhook grup, yang menerima acara dari grup tertentu GitLab . Untuk informasi selengkapnya, lihat GitLab kelompok webhooks.

Bisakah saya menjalankan pekerjaan di docker executor untuk pelari yang dikelola sendiri? Misalnya, saya ingin menjalankan pekerjaan pipeline pada gambar tertentu untuk mempertahankan lingkungan build yang sama dalam wadah terpisah dan terisolasi.

Anda dapat menjalankan runner yang GitLab dikelola sendiri CodeBuild dengan gambar tertentu dengan membuat proyek dengan gambar khusus atau mengganti gambar di file Anda. `.gitlab-ci.yml`

Eksekutor apa yang dijalankan oleh pelari yang dikelola sendiri? CodeBuild

Runner yang dikelola sendiri CodeBuild berjalan dengan eksekutor shell, di mana build berjalan secara lokal bersama dengan GitLab runner yang berjalan di dalam wadah docker.

Bisakah saya memberikan perintah buildspec bersama dengan pelari yang dikelola sendiri?

Ya, dimungkinkan untuk menambahkan perintah buildspec bersama dengan pelari yang dikelola sendiri. Anda dapat menyediakan file buildspec.yl di GitLab repositori Anda dan menggunakan **`buildspec-override:true`** tag di bagian Tag pekerjaan. Untuk informasi selengkapnya, lihat Nama file buildspec dan lokasi penyimpanan.

Wilayah mana yang mendukung penggunaan CodeBuild-hosted GitLab pelari?

CodeBuild-hosted GitLab pelari didukung di semua CodeBuild wilayah. Untuk informasi selengkapnya tentang Wilayah AWS tempat CodeBuild tersedia, lihat AWS Layanan menurut Wilayah.

Platform mana yang mendukung penggunaan CodeBuild-hosted GitLab pelari?

CodeBuild-hosted GitLab pelari didukung di Amazon EC2 AWS Lambdadan komputasi. Anda dapat menggunakan platform berikut: Amazon Linux 2, Amazon Linux 2023, Ubuntu, dan Windows Server Core 2019. Untuk informasi selengkapnya, lihat Gambar komputasi EC2 dan Gambar komputasi Lambda.

Tutorial: Konfigurasikan CodeBuild-hosted GitLab pelari

Tutorial ini menunjukkan cara mengkonfigurasi CodeBuild proyek Anda untuk menjalankan pekerjaan GitLab CI/CD pipeline. Untuk informasi selengkapnya tentang menggunakan GitLab atau Dikelola GitLab Sendiri dengan CodeBuild, lihatSelf-managed GitLab pelari di AWS CodeBuild.

Untuk menyelesaikan tutorial ini, Anda harus terlebih dahulu:

- Connect dengan aplikasi OAuth dengan menggunakan. AWS CodeConnections Perhatikan bahwa saat menghubungkan dengan aplikasi OAuth, Anda harus menggunakan CodeBuild konsol untuk melakukannya. Untuk instruksi lebih lanjut, lihatGitLab akses di CodeBuild.
- Connect CodeBuild ke GitLab akun Anda. Untuk melakukannya, Anda dapat menambahkan GitLab sebagai penyedia sumber di konsol. Untuk petunjuk, lihat GitLab akses di CodeBuild.

> **Note**
>
> Ini hanya perlu dilakukan jika Anda belum terhubung ke GitLab akun Anda. Dengan fitur ini, CodeBuild perlu izin tambahan, seperti `create_runner` dan `manage_runner` dari aplikasi GitLab OAuth. Jika ada AWS CodeConnections untuk GitLab akun tertentu, maka akun tersebut tidak secara otomatis meminta pembaruan izin. Untuk melakukannya, Anda dapat pergi ke AWS CodeConnections konsol dan membuat koneksi dummy ke GitLab akun yang sama untuk memicu otorisasi ulang untuk mendapatkan izin tambahan. Setelah menyelesaikan langkah ini, semua koneksi yang ada dapat menggunakan fitur runner. Setelah selesai, Anda dapat menghapus koneksi dummy.

Langkah 1: Buat CodeBuild proyek dengan webhook

Pada langkah ini, Anda akan membuat CodeBuild proyek dengan webhook dan meninjaunya di GitLab konsol.

Untuk membuat CodeBuild proyek dengan webhook

1. Buka AWS CodeBuild konsol di https://console.aws.amazon.com/codesuite/codebuild/home.
2. Buat proyek build. Untuk informasi selengkapnya, lihat Buat proyek build (konsol) dan Jalankan build (konsol).

   Pada tipe Project, pilih Runner project.

- Di Runner:
  - Untuk penyedia Runner, pilih GitLab.
  - Untuk Credential, pilih salah satu dari berikut ini:
    - Pilih Default source credential. Koneksi default menerapkan GitLab koneksi default di semua proyek.
    - Pilih Kredensi sumber kustom. Koneksi kustom menerapkan GitLab koneksi khusus yang mengesampingkan pengaturan default akun Anda.

    > **Note**
    >
    > Jika Anda belum membuat koneksi ke penyedia Anda, Anda harus membuat GitLab koneksi baru. Untuk petunjuk, lihat Connect CodeBuild ke GitLab.

  - Untuk lokasi Runner, pilih Repository.
  - Untuk Repositori, pilih nama proyek Anda GitLab dengan menentukan jalur proyek dengan namespace.
- Di Lingkungan:
  - Pilih gambar Lingkungan yang didukung dan Compute. Perhatikan bahwa Anda memiliki opsi untuk mengganti pengaturan gambar dan instans dengan menggunakan label di YAMAL GitLab CI/CD pipeline Anda. Untuk informasi selengkapnya, lihat Langkah 2: Buat file.gitlab-ci.yl di repositori Anda.
- Di Buildspec:
  - Perhatikan bahwa buildspec Anda akan diabaikan kecuali `buildspec-override:true` ditambahkan sebagai label. Sebagai gantinya, CodeBuild akan menimpanya untuk menggunakan perintah yang akan mengatur pelari yang dikelola sendiri.
-

3. Lanjutkan dengan nilai default dan kemudian pilih Create build project.
4. Buka GitLab konsol di `https://gitlab.com/`*`user-name`*`/`*`repository-name`*`/-/hooks` untuk memverifikasi bahwa webhook telah dibuat dan diaktifkan untuk mengirimkan peristiwa pekerjaan Alur Kerja.

Langkah 2: Buat file.gitlab-ci.yl di repositori Anda

Pada langkah ini, Anda akan membuat `.gitlab-ci.yml` file GitLabuntuk mengonfigurasi lingkungan build Anda dan menggunakan pelari yang GitLab dikelola sendiri. CodeBuild Untuk informasi selengkapnya, lihat Menggunakan pelari yang dikelola sendiri.

Perbarui GitLab CI/CD saluran pipa YAMM

Arahkan ke `https://gitlab.com/`*`user-name`*`/`*`project-name`*`/-/tree/`*`branch-name` dan buat `.gitlab-ci.yml` file di repositori Anda. Anda dapat mengonfigurasi lingkungan build dengan melakukan salah satu hal berikut:

- Anda dapat menentukan nama CodeBuild proyek, dalam hal ini build akan menggunakan konfigurasi proyek yang ada untuk komputasi, gambar, versi gambar, dan ukuran instance. Nama proyek diperlukan untuk menautkan pengaturan AWS terkait GitLab pekerjaan Anda ke CodeBuild proyek tertentu. Dengan memasukkan nama proyek di YAMAL, CodeBuild diizinkan untuk memanggil pekerjaan dengan pengaturan proyek yang benar.

```
tags:
    - codebuild-<codebuild-project-name>-$CI_PROJECT_ID-$CI_PIPELINE_IID-$CI_JOB_NAME
```

`$CI_PROJECT_ID-$CI_PIPELINE_IID-$CI_JOB_NAME`diperlukan untuk memetakan build ke pekerjaan pipeline tertentu yang berjalan dan menghentikan pembangunan saat proses pipeline dibatalkan.

> **Note**
>
> Pastikan bahwa Anda *`<project-name>`* cocok dengan nama proyek yang Anda buat CodeBuild. Jika tidak cocok, tidak CodeBuild akan memproses webhook dan GitLab CI/CD pipeline mungkin hang.

Berikut ini adalah contoh dari GitLab CI/CD pipa YAMAL:

```
workflow:
  name: HelloWorld
stages:          # List of stages for jobs, and their order of execution
  - build

build-job:       # This job runs in the build stage, which runs first.
```

```
  stage: build
  script:
    - echo "Hello World!"
  tags:
    - codebuild-myProject-$CI_PROJECT_ID-$CI_PIPELINE_IID-$CI_JOB_NAME
```

- Anda juga dapat mengganti gambar dan jenis komputasi di tag. Lihat Hitung gambar yang didukung dengan pelari CodeBuild-hosted GitLab untuk daftar gambar yang dikuratori. Untuk menggunakan gambar kustom, lihatPenggantian label didukung dengan pelari CodeBuild-hosted GitLab. Jenis komputasi dan gambar dalam tag akan mengganti pengaturan lingkungan pada proyek Anda. Untuk mengganti setelan lingkungan Anda untuk build komputasi Amazon EC2, gunakan sintaks berikut:

  ```
  tags:
      - codebuild-<codebuild-project-name>-$CI_PROJECT_ID-$CI_PIPELINE_IID-$CI_JOB_NAME
      - image:<environment-type>-<image-identifier>
      - instance-size:<instance-size>
  ```

  Berikut ini adalah contoh dari GitLab CI/CD pipa YAMAL:

  ```
  stages:
    - build

  build-job:
    stage: build
    script:
      - echo "Hello World!"
    tags:
      - codebuild-myProject-$CI_PROJECT_ID-$CI_PIPELINE_IID-$CI_JOB_NAME
      - image:arm-3.0
      - instance-size:small
  ```

- Anda dapat mengganti armada yang digunakan untuk build Anda di tag. Ini akan menggantikan pengaturan armada yang dikonfigurasi pada proyek Anda untuk menggunakan armada yang ditentukan. Untuk informasi selengkapnya, lihat Jalankan build pada armada kapasitas yang dicadangkan. Untuk mengganti setelan armada untuk build komputasi Amazon EC2, gunakan sintaks berikut:

  ```
  tags:
      - codebuild-<codebuild-project-name>-$CI_PROJECT_ID-$CI_PIPELINE_IID-$CI_JOB_NAME
  ```

```
    - fleet:<fleet-name>
```

Untuk mengganti armada dan gambar yang digunakan untuk build, gunakan sintaks berikut:

```
tags:
    - codebuild-<codebuild-project-name>-$CI_PROJECT_ID-$CI_PIPELINE_IID-$CI_JOB_NAME
    - fleet:<fleet-name>
    - image:<environment-type>-<image-identifier>
```

Berikut ini adalah contoh dari GitLab CI/CD pipa YAMAL:

```
stages:
  - build

build-job:
  stage: build
  script:
    - echo "Hello World!"
  tags:
    - codebuild-myProject-$CI_PROJECT_ID-$CI_PIPELINE_IID-$CI_JOB_NAME
    - fleet:myFleet
    - image:arm-3.0
```

- Untuk menjalankan tugas GitLab CI/CD pipeline Anda pada gambar kustom, Anda dapat mengonfigurasi gambar kustom dalam CodeBuild proyek Anda dan menghindari memberikan label penggantian gambar. CodeBuild akan menggunakan gambar yang dikonfigurasi dalam proyek jika tidak ada label penggantian gambar yang disediakan.

Setelah Anda melakukan perubahan `.gitlab-ci.yml`, GitLab pipeline akan dipicu dan `build-job` akan mengirim notifikasi webhook yang akan memulai build in CodeBuild Anda.

Jalankan perintah buildspec di fase INSTALL, PRE_BUILD, dan POST_BUILD

Secara default, CodeBuild abaikan perintah buildspec apa pun saat menjalankan build yang dikelola sendiri. GitLab Untuk menjalankan perintah buildspec selama build, `buildspec-override:true` dapat ditambahkan sebagai akhiran untuk: `tags`

```
tags:
    - codebuild-<codebuild-project-name>-$CI_PROJECT_ID-$CI_PIPELINE_IID-$CI_JOB_NAME
    - buildspec-override:true
```

Dengan menggunakan perintah ini, CodeBuild akan membuat folder yang disebut `gitlab-runner` dalam folder sumber utama wadah. Ketika GitLab pelari dimulai selama `BUILD` fase, pelari akan berjalan di direktori. `gitlab-runner`

Ada beberapa batasan saat menggunakan penggantian buildspec dalam build yang dikelola sendiri: GitLab

- CodeBuild tidak akan menjalankan perintah buildspec selama `BUILD` fase, karena pelari yang dikelola sendiri berjalan dalam fase tersebut. `BUILD`
- CodeBuild tidak akan mengunduh sumber primer atau sekunder selama `DOWNLOAD_SOURCE` fase. Jika Anda memiliki file buildspec yang dikonfigurasi, hanya file itu yang akan diunduh dari sumber utama proyek.
- Jika perintah build gagal di `INSTALL` fase `PRE_BUILD` atau, tidak CodeBuild akan memulai runner yang dikelola sendiri dan tugas GitLab CI/CD pipeline perlu dibatalkan secara manual.
- CodeBuild mengambil token runner selama `DOWNLOAD_SOURCE` fase, yang memiliki waktu kedaluwarsa satu jam. Jika `INSTALL` fase `PRE_BUILD` atau fase Anda melebihi satu jam, token runner mungkin kedaluwarsa sebelum pelari yang GitLab dikelola sendiri dimulai.

Langkah 3: Tinjau hasil Anda

Setiap kali GitLab CI/CD proses pipeline terjadi, CodeBuild akan menerima peristiwa pekerjaan CI/CD pipeline melalui webhook. Untuk setiap pekerjaan dalam CI/CD pipeline, CodeBuild mulailah membangun untuk menjalankan pelari fana GitLab . Pelari bertanggung jawab untuk menjalankan pekerjaan CI/CD pipa tunggal. Setelah pekerjaan selesai, runner dan proses build terkait akan segera dihentikan.

Untuk melihat log pekerjaan CI/CD pipeline, navigasikan ke repositori GitLab, pilih Build, Jobs, lalu pilih Job spesifik yang ingin Anda tinjau log.

Anda dapat meninjau label yang diminta di log saat pekerjaan menunggu untuk diambil oleh pelari yang dikelola sendiri. CodeBuild

Filter acara GitLab webhook (CloudFormation)

YAML-formatted Bagian berikut dari CloudFormation template membuat grup filter yang memicu build saat dievaluasi ke true. Grup filter berikut menentukan permintaan pekerjaan GitLab CI/CD pipeline dengan nama CI/CD pipeline yang cocok dengan ekspresi `\[CI-CodeBuild\]` reguler.

```
CodeBuildProject:
```

```
  Type: AWS::CodeBuild::Project
  Properties:
    Name: MyProject
    ServiceRole: service-role
    Artifacts:
      Type: NO_ARTIFACTS
    Environment:
      Type: LINUX_CONTAINER
      ComputeType: BUILD_GENERAL1_SMALL
      Image: aws/codebuild/standard:5.0
    Source:
      Type: GITLAB
      Location: CODEBUILD_DEFAULT_WEBHOOK_SOURCE_LOCATION
    Triggers:
      Webhook: true
      ScopeConfiguration:
        Name: group-name
        Scope: GITLAB_GROUP
      FilterGroups:
        - - Type: EVENT
            Pattern: WORKFLOW_JOB_QUEUED
          - Type: WORKFLOW_NAME
            Pattern: \[CI-CodeBuild\]
```

Penggantian label didukung dengan pelari CodeBuild-hosted GitLab

Di GitLab CI/CD YAMAL pipeline, Anda dapat memberikan berbagai penggantian label yang memodifikasi build runner yang dikelola sendiri. Setiap build yang tidak dikenali oleh CodeBuild akan diabaikan tetapi tidak akan gagal permintaan webhook Anda. Misalnya, YAMAL berikut menyertakan penggantian untuk image, ukuran instans, armada, dan spesifikasi build:

```
workflow:
  name: HelloWorld
stages:
  - build

build-job:
  stage: build
  script:
    - echo "Hello World!"
  tags:
    - codebuild-myProject-$CI_PROJECT_ID-$CI_PIPELINE_IID-$CI_JOB_NAME
    - image:arm-3.0
```

```
    - instance-size:small
    - fleet:myFleet
    - buildspec-override:true
```

`codebuild-<project-name>-$CI_PROJECT_ID-$CI_PIPELINE_IID-$CI_JOB_NAME` (wajib)

- Contoh: `codebuild-myProject-$CI_PROJECT_ID-$CI_PIPELINE_IID-$CI_JOB_NAME`
- Diperlukan untuk semua YAML GitLab CI/CD pipeline. *<project name>*harus sama dengan nama proyek tempat webhook pelari yang dikelola sendiri dikonfigurasi.

`image:<environment-type>-<image-identifier>`

- Contoh: `image:arm-3.0`
- Mengganti jenis gambar dan lingkungan yang digunakan saat memulai build runner yang dikelola sendiri. Untuk mempelajari nilai yang didukung, lihatHitung gambar yang didukung dengan pelari CodeBuild-hosted GitLab.
  - Untuk mengganti jenis gambar dan lingkungan yang digunakan dengan gambar khusus, gunakan `image:custom-<environment-type>-<custom-image-identifier>`
  - Contoh: `image:custom-arm-public.ecr.aws/codebuild/amazonlinux-aarch64-standard:3.0`

> **Note**
>
> Jika gambar kustom berada di registri pribadi, lihatKonfigurasikan kredenal registri pribadi untuk pelari yang dihosting sendiri.

`instance-size:<instance-size>`

- Contoh: `instance-size:small`
- Mengganti tipe instance yang digunakan saat memulai build runner yang dikelola sendiri. Untuk mempelajari nilai yang didukung, lihatHitung gambar yang didukung dengan pelari CodeBuild-hosted GitLab.

`fleet:<fleet-name>`

- Contoh: `fleet:myFleet`

- Mengganti pengaturan armada yang dikonfigurasi pada proyek Anda untuk menggunakan armada yang ditentukan. Untuk informasi selengkapnya, lihat Jalankan build pada armada kapasitas yang dicadangkan.

`buildspec-override:<boolean>`

- Contoh: `buildspec-override:true`
- Mengizinkan build menjalankan perintah buildspec di`INSTALL`,`PRE_BUILD`, dan `POST_BUILD` fase jika disetel ke. `true`

Hitung gambar yang didukung dengan pelari CodeBuild-hosted GitLab

Pada label yang Anda konfigurasikanTutorial: Konfigurasikan CodeBuild-hosted GitLab pelari, Anda dapat mengganti pengaturan lingkungan Amazon EC2 dengan menggunakan nilai di tiga kolom pertama. CodeBuild menyediakan gambar komputasi Amazon EC2 berikut. Untuk informasi lebih lanjut tentang

Tipe lingkungan	Pengidentifikasi gambar	Ukuran instans	Platform	Citra	Definisi

`linux`	`4.0`	`small`
		`medium`
		`large`
		`xlarge`
			Amazon Linux 2	`aws/codebuild/amazonlinux-x86_64-standard:4.0`	al/standard/4.0

`linux`	`5.0`	`2xlarge`
		`gpu_small`
		`gpu_large`
			Amazon Linux 2023	`aws/codebuild/amazonlinux-x86_64-standard:5.0`	al/standard/5.0

`linux-ec2`	`latest`	`small` `medium`	Amazon Linux 2023	`aws/codebuild/ami/amazonlin`	Tidak ada

Tipe lingkungan	Pengidentifikasi gambar	Ukuran instans	Platform	Citra	Definisi
		large		ux-x86_64-base:latest	
arm	2.0	small medium large xlarge	Amazon Linux 2	aws/codebuild/amazonlinux-aarch64-standard:2.0	al/aarch64/standard/2.0
arm	3.0	2xlarge	Amazon Linux 2023	aws/codebuild/amazonlinux-aarch64-standard:3.0	al/aarch64/standard/3.0
arm-ec2	latest	small medium large	Amazon Linux 2023	aws/codebuild/ami/amazonlinux-arm-base:latest	Tidak ada

Tipe lingkungan	Pengidentifikasi gambar	Ukuran instans	Platform	Citra	Definisi
ubuntu	5.0	small medium large xlarge 2xlarge gpu_small gpu_large	Ubuntu 20.04	aws/codebuild/standard:5.0	ubuntu/standard/5.0
ubuntu	6.0		Ubuntu 22.04	aws/codebuild/standard:6.0	ubuntu/standard/6.0
ubuntu	7.0		Ubuntu 22.04	aws/codebuild/standard:7.0	ubuntu/standard/7.0
windows	1.0	medium large	Inti Server Windows 2019	aws/codebuild/windows-base:2019-1.0	N/A
			Inti Server Windows 2022	aws/codebuild/windows-base:2022-1.0	N/A
windows	2.0		Inti Server Windows 2019	aws/codebuild/windows-base:2019-2.0	N/A
windows	3.0		Inti Server Windows 2019	aws/codebuild/windows-base:2019-3.0	N/A

Tipe lingkungan	Pengidentifikasi gambar	Ukuran instans	Platform	Citra	Definisi
`windows-ec2`	`2022`	`medium` `large`	Inti Server Windows 2022	`aws/codebuild/ami/windows-base:2022`	Tidak ada

Selain itu, Anda dapat mengganti pengaturan lingkungan Lambda Anda dengan menggunakan nilai berikut. Untuk informasi selengkapnya tentang komputasi CodeBuild Lambda, lihat. Jalankan dibangun di AWS Lambda komputasi CodeBuild mendukung gambar komputasi Lambda berikut:

Tipe lingkungan	Versi waktu aktif	Ukuran instans			

`linux-lambda`
`arm-lambda`
	`dotnet6`
	`go1.21`
	`corretto11`
	`corretto17`
	`corretto21`
	`nodejs18`
	`nodejs20`
	`python3.11`
	`python3.12`
		`1GB`
		`2GB`
		`4GB`
		`8GB`
		`10GB`			

Tipe lingkungan	Versi waktu aktif	Ukuran instans			
	`ruby3.2`				

Untuk informasi selengkapnya, lihat Membangun mode dan tipe komputasi lingkungan dan Gambar Docker disediakan oleh CodeBuild.

Self-managed Pelari buildkite di AWS CodeBuild

Anda dapat mengonfigurasi proyek Anda untuk menyiapkan runner Buildkite yang dihosting sendiri dalam CodeBuild kontainer untuk memproses pekerjaan Buildkite Anda. Ini dapat dilakukan dengan menyiapkan webhook menggunakan CodeBuild proyek Anda, dan memperbarui langkah-langkah YAMAL pipeline Buildkite Anda untuk menggunakan runner yang dihosting sendiri yang dihosting di mesin. CodeBuild

Langkah-langkah tingkat tinggi untuk mengonfigurasi CodeBuild proyek untuk menjalankan pekerjaan Buildkite adalah sebagai berikut:

- Arahkan ke CodeBuild konsol dan buat CodeBuild proyek dengan konfigurasi tipe runner project runner Buildkite
- Tambahkan `job.scheduled` webhook ke organisasi Buildkite Anda.
- Perbarui langkah YAMM pipeline Buildkite Anda di Buildkite untuk mengonfigurasi lingkungan build Anda.

Untuk prosedur yang lebih rinci, lihatTutorial: Konfigurasikan CodeBuild-hosted pelari Buildkite. Fitur ini memungkinkan pekerjaan Buildkite Anda untuk mendapatkan integrasi asli AWS, yang memberikan keamanan dan kenyamanan melalui fitur seperti IAM,, AWS Secrets Manager, dan AWS CloudTrail Amazon VPC. Anda dapat mengakses jenis instans terbaru, termasuk ARM-based instance.

Tentang pelari CodeBuild-hosted Buildkite

Berikut ini adalah beberapa pertanyaan umum tentang CodeBuild-hosted pelari Buildkite.

Kapan saya harus menyertakan penggantian gambar dan instance di label?

Anda dapat menyertakan penggantian gambar dan instance dalam label untuk menentukan lingkungan build yang berbeda untuk setiap pekerjaan Buildkite Anda. Ini dapat dilakukan tanpa perlu membuat beberapa CodeBuild proyek atau webhook. Misalnya, ini berguna ketika Anda perlu menggunakan matriks untuk pekerjaan Buildkite.

```
agents:
  queue: "myQueue"
steps:
  - command: "echo \"Hello World\""
    agents:
      project: "codebuild-myProject"
      image: "{{matrix.os}}"
      instance-size: "{{matrix.size}}"
    matrix:
      setup:
        os:
          - "arm-3.0"
          - "al2-5.0"
        size:
          - "small"
          - "large"
```

Bisakah CodeBuild membuat webhook dalam Buildkite secara otomatis?

Saat ini, Buildkite mengharuskan semua webhook dibuat secara manual menggunakan konsol mereka. Anda dapat mengikuti tutorial di Tutorial: Konfigurasikan CodeBuild-hosted pelari Buildkite untuk membuat webhook Buildkite secara manual di konsol Buildkite.

Dapatkah saya menggunakan CloudFormation untuk membuat webhook Buildkite?

CloudFormation saat ini tidak didukung untuk webhook runner Buildkite, karena Buildkite memerlukan webhook untuk dibuat secara manual menggunakan konsolnya.

Wilayah mana yang mendukung penggunaan CodeBuild-hosted pelari Buildkite?

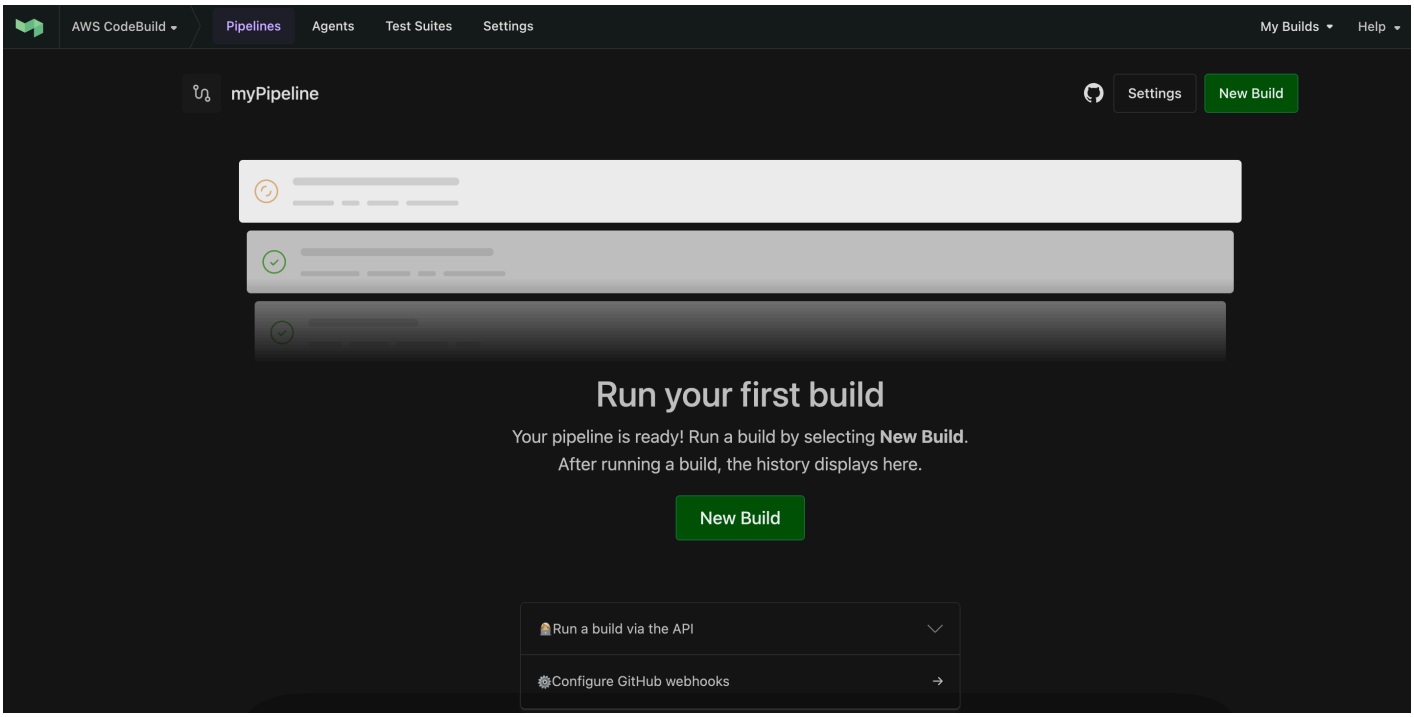
CodeBuild-hosted Pelari buildkite didukung di semua wilayah. CodeBuild Untuk informasi selengkapnya tentang AWS Wilayah yang CodeBuild tersedia, lihat AWS Layanan menurut Wilayah.

Tutorial: Konfigurasikan CodeBuild-hosted pelari Buildkite

Tutorial ini menunjukkan cara mengkonfigurasi CodeBuild proyek Anda untuk menjalankan pekerjaan Buildkite. Untuk informasi selengkapnya tentang menggunakan Buildkite dengan lihat. CodeBuild Self-managed Pelari buildkite di AWS CodeBuild

Untuk menyelesaikan tutorial ini, Anda harus terlebih dahulu:

- Memiliki akses ke organisasi Buildkite. Untuk informasi selengkapnya tentang menyiapkan akun dan organisasi Buildkite, Anda dapat mengikuti Tutorial Memulai ini.
- Buat pipeline, cluster, dan antrean Buildkite yang dikonfigurasi untuk menggunakan runner yang dihosting sendiri. Untuk informasi lebih lanjut tentang pengaturan sumber daya ini, Anda dapat mereferensikan Buildkite Pipeline Setup Tutorial.



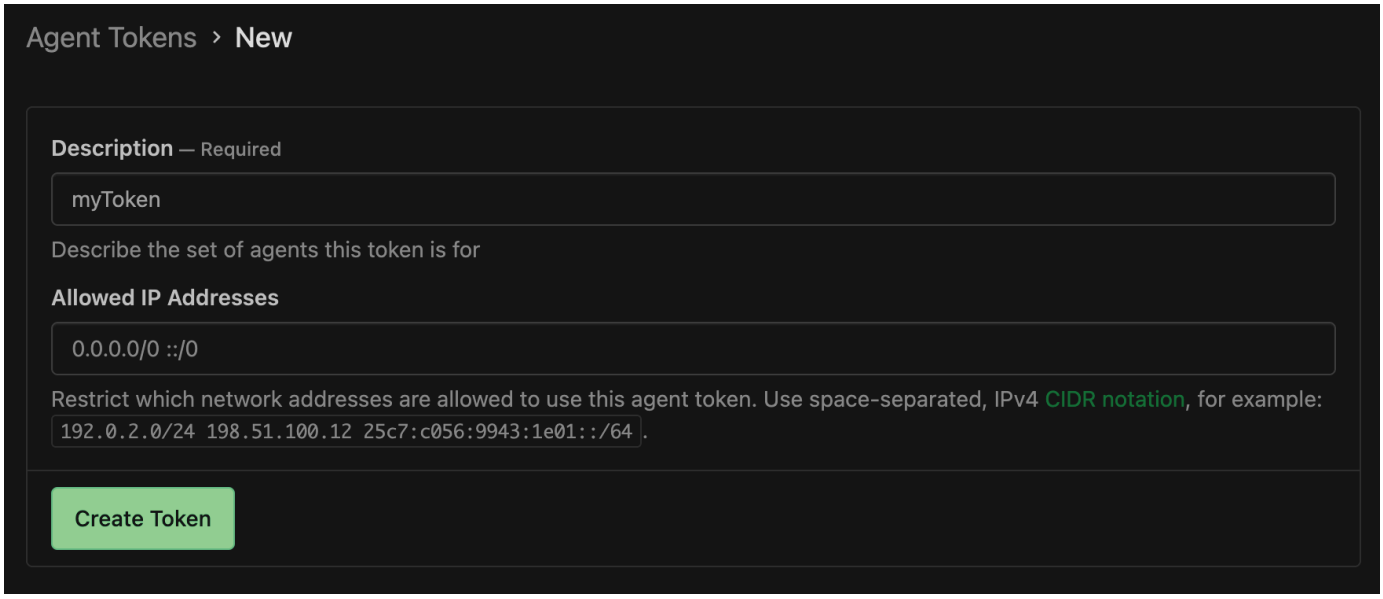
Langkah 1: Buat token agen Buildkite

Pada langkah ini, Anda akan menghasilkan token agen dalam Buildkite yang akan digunakan untuk mengautentikasi pelari yang dihosting sendiri. CodeBuild Untuk informasi selengkapnya tentang sumber daya ini, lihat Token Agen Buildkite.

Untuk menghasilkan token agen Buildkite

1. Di klaster Buildkite Anda, pilih Token Agen, lalu pilih Token Baru.

2. Tambahkan deskripsi ke token dan klik Buat Token.
3. Simpan nilai token agen, karena akan digunakan nanti selama penyiapan CodeBuild proyek.



Langkah 2: Buat CodeBuild proyek dengan webhook

Untuk membuat CodeBuild proyek dengan webhook

1. Buka AWS CodeBuild konsol di https://console.aws.amazon.com/codesuite/codebuild/home.
2. Buat proyek build yang dihosting sendiri. Untuk informasi selengkapnya, lihat Buat proyek build (konsol) dan Jalankan build (konsol).
   - Dalam konfigurasi Project, pilih Runner project. Di Runner:
     - Untuk penyedia Runner, pilih Buildkite.
     - Untuk token agen Buildkite, pilih Buat token agen baru dengan menggunakan halaman buat rahasia. Anda akan diminta untuk membuat rahasia baru dengan nilai rahasia yang sama AWS Secrets Manager dengan token agen Buildkite yang Anda buat di atas.
     - (Opsional) Jika Anda ingin menggunakan kredensil CodeBuild terkelola untuk pekerjaan Anda, pilih penyedia repositori sumber pekerjaan Anda di bawah opsi kredensi sumber Buildkite dan verifikasi bahwa kredensil dikonfigurasi untuk akun Anda. Selain itu, verifikasi bahwa pipeline Buildkite Anda menggunakan Checkout menggunakan HTTPS.

> **Note**
>
> Buildkite memerlukan kredensi sumber dalam lingkungan build untuk menarik sumber untuk pekerjaan Anda. Lihat Mengautentikasi Buildkite ke Repositori Pribadi opsi kredensi sumber yang tersedia.

- (Opsional) Di Lingkungan:
  - Pilih gambar Lingkungan yang didukung dan Compute.

    Perhatikan bahwa Anda memiliki opsi untuk mengganti setelan gambar dan instans dengan menggunakan label dalam langkah YAMAL Buildkite Anda. Untuk informasi selengkapnya, lihat Langkah 4: Perbarui langkah-langkah pipeline Buildkite.
- (Opsional) Di Buildspec:
  - Buildspec Anda akan diabaikan secara default kecuali `buildspec-override: "true"` ditambahkan sebagai label. Sebagai gantinya, CodeBuild akan menimpanya untuk menggunakan perintah yang akan mengatur pelari yang dihosting sendiri.

    > **Note**
    >
    > CodeBuild tidak mendukung file buildspec untuk build runner yang dihosting sendiri Buildkite. Untuk spesifikasi build sebaris, Anda harus mengaktifkan git-credential-helper di buildspec Anda jika Anda telah mengonfigurasi kredenal sumber terkelola CodeBuild

3. Lanjutkan dengan nilai default dan kemudian pilih Create build project.
4. Simpan URL Payload dan nilai Rahasia dari popup Create Webhook. Ikuti petunjuk di popup untuk membuat webhook organisasi Buildkite baru atau lanjutkan ke bagian berikutnya.

Langkah 3: Buat CodeBuild webhook dalam Buildkite

Pada langkah ini, Anda akan menggunakan URL Payload dan nilai Rahasia dari CodeBuild webhook untuk membuat webhook baru dalam Buildkite. Webhook ini akan digunakan untuk memicu build di dalam CodeBuild saat pekerjaan Buildkite yang valid dimulai.

Untuk membuat webhook baru di Buildkite

1. Buka halaman Pengaturan organisasi Buildkite Anda.

2. Di bawah Integrasi, pilih Layanan Pemberitahuan.
3. Pilih Tambahkan di sebelah kotak Webhook. Di halaman Add Webhook Notification, gunakan konfigurasi berikut:

   a. Di bawah URL Webhook, tambahkan nilai URL Payload yang disimpan.

   b. Di bawah Token, verifikasi bahwa Kirim token seperti X-Buildkite-Token yang dipilih. Tambahkan nilai Rahasia webhook Anda ke bidang Token.

   c. Di bawah, verifikasi itu Kirim token seperti X-Buildkite-Token yang dipilih. Tambahkan nilai Rahasia webhook Anda ke bidang Token.

   d. Di bawah Acara, pilih acara `job.scheduled` webhook.

   e. (Opsional) Di bawah Pipelines, Anda dapat memilih untuk hanya memicu build untuk pipeline tertentu.
4. Pilih Tambahkan Pemberitahuan Webhook.

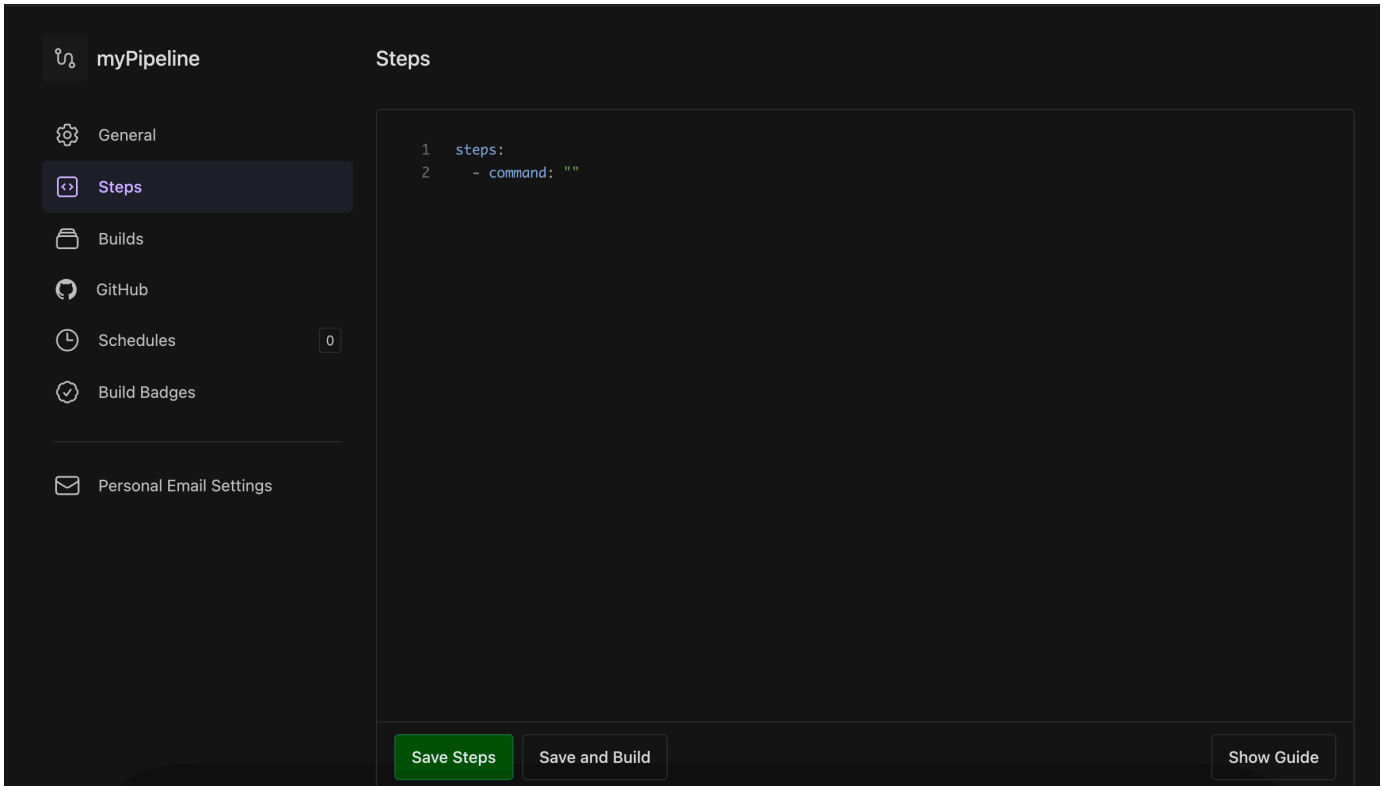
Langkah 4: Perbarui langkah-langkah pipeline Buildkite

Pada langkah ini, perbarui langkah-langkah pipeline Buildkite Anda untuk menambahkan label yang diperlukan dan penggantian opsional. Untuk daftar lengkap penggantian label yang didukung, lihat. Penggantian label didukung dengan runner CodeBuild-hosted Buildkite

Perbarui langkah pipeline Anda

1. Buka halaman langkah pipeline Buildkite dengan memilih pipeline Buildkite, pilih Pengaturan, lalu pilih Langkah.

   Jika Anda belum melakukannya, pilih Mengonversi ke langkah YAMAL.



2. Minimal, Anda harus menentukan tag agen Buildkite yang merujuk nama pipeline Anda. CodeBuild Nama proyek diperlukan untuk menautkan pengaturan AWS terkait pekerjaan Buildkite Anda ke proyek tertentu. CodeBuild Dengan memasukkan nama proyek di YAMAL, CodeBuild diizinkan untuk memanggil pekerjaan dengan pengaturan proyek yang benar.

```
agents:
  project: "codebuild-<project name>"
```

Berikut ini adalah contoh langkah pipeline Buildkite hanya dengan tag label proyek:

```
agents:
  project: "codebuild-myProject"
steps:
  - command: "echo \"Hello World\""
```

Anda juga dapat mengganti gambar dan jenis komputasi di label. Lihat Hitung gambar yang didukung dengan runner CodeBuild-hosted Buildkite daftar gambar yang tersedia. Jenis komputasi dan gambar dalam label akan mengganti pengaturan lingkungan pada proyek Anda. Untuk mengganti setelan lingkungan untuk build komputasi CodeBuild EC2 atau Lambda, gunakan sintaks berikut:

```
agents:
  project: "codebuild-<project name>"
  image: "<environment-type>-<image-identifier>"
  instance-size: "<instance-size>"
```

Berikut ini adalah contoh langkah pipeline Buildkite dengan penggantian ukuran gambar dan instance:

```
agents:
  project: "codebuild-myProject"
  image: "arm-3.0"
  instance-size: "small"
steps:
  - command: "echo \"Hello World\""
```

Anda dapat mengganti armada yang digunakan untuk build Anda di label. Ini akan menggantikan pengaturan armada yang dikonfigurasi pada proyek Anda untuk menggunakan armada yang ditentukan. Untuk informasi selengkapnya, lihat Menjalankan build di atas armada kapasitas cadangan.

Untuk mengganti setelan armada untuk build komputasi Amazon EC2, gunakan sintaks berikut:

```
agents:
  project: "codebuild-<project name>"
  fleet: "<fleet-name>"
```

Untuk mengganti armada dan gambar yang digunakan untuk build, gunakan sintaks berikut:

```
agents:
  project: "codebuild-<project name>"
  fleet: "<fleet-name>"
  image: "<environment-type>-<image-identifier>"
```

Berikut ini adalah contoh langkah pipeline Buildkite dengan penggantian armada dan gambar:

```
agents:
  project: "codebuild-myProject"
  fleet: "myFleet"
  image: "arm-3.0"
```

```
steps:
  - command: "echo \"Hello World\""
```

3. Anda dapat memilih untuk menjalankan perintah buildspec sebaris selama build runner Buildkite yang dihosting sendiri (lihat untuk detail selengkapnya). Jalankan perintah buildspec untuk fase INSTALL, PRE_BUILD, dan POST_BUILD Untuk menentukan bahwa CodeBuild build harus menjalankan perintah buildspec selama build runner yang di-host sendiri Buildkite, gunakan sintaks berikut:

   ```
   agents:
     project: "codebuild-<project name>"
     buildspec-override: "true"
   ```

   Berikut ini adalah contoh pipeline Buildkite dengan penggantian buildspec:

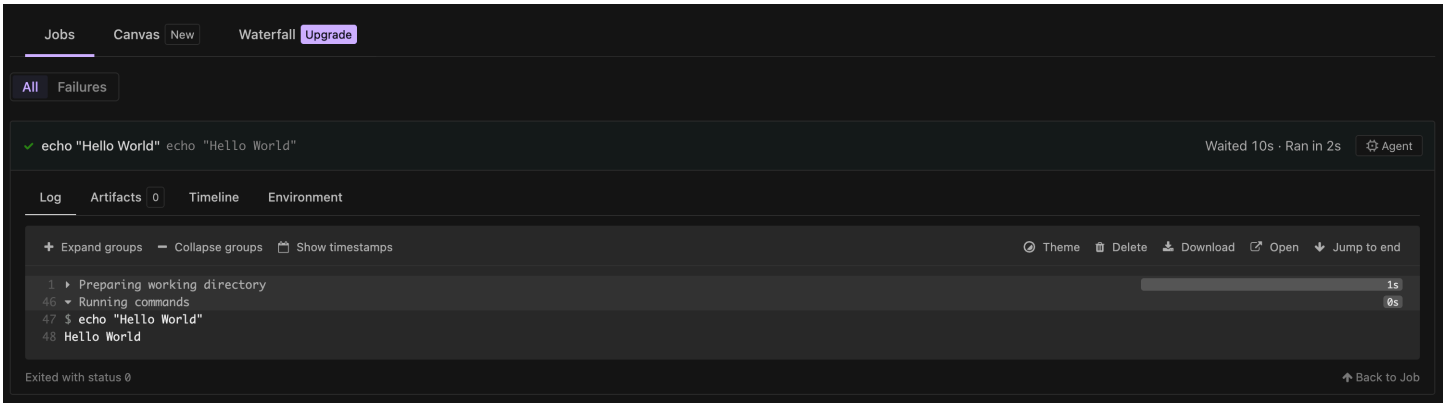
   ```
   agents:
     project: "codebuild-myProject"
     buildspec-override: "true"
   steps:
     - command: "echo \"Hello World\""
   ```

4. Secara opsional, Anda dapat memberikan label di luar label yang CodeBuild mendukung. Label ini akan diabaikan untuk tujuan mengesampingkan atribut build, tetapi tidak akan gagal dalam permintaan webhook. Misalnya, menambahkan `myLabel: "testLabel"` sebagai label tidak akan mencegah build berjalan.

Langkah 5: Tinjau hasil Anda

Setiap kali pekerjaan Buildkite dimulai di pipeline Anda, CodeBuild akan menerima acara webhook melalui `job.scheduled` webhook Buildkite. Untuk setiap pekerjaan di build Buildkite Anda, CodeBuild akan memulai build untuk menjalankan runner Buildkite sementara. Pelari bertanggung jawab untuk menjalankan satu pekerjaan Buildkite. Setelah pekerjaan selesai, runner dan proses build terkait akan segera dihentikan.

Untuk melihat log pekerjaan alur kerja, navigasikan ke pipeline Buildkite dan pilih build terbaru (Anda dapat memicu build baru dengan memilih New Build). Setelah CodeBuild build terkait untuk setiap pekerjaan Anda dimulai dan mengambil pekerjaan, Anda akan melihat log untuk pekerjaan tersebut di dalam konsol Buildkite



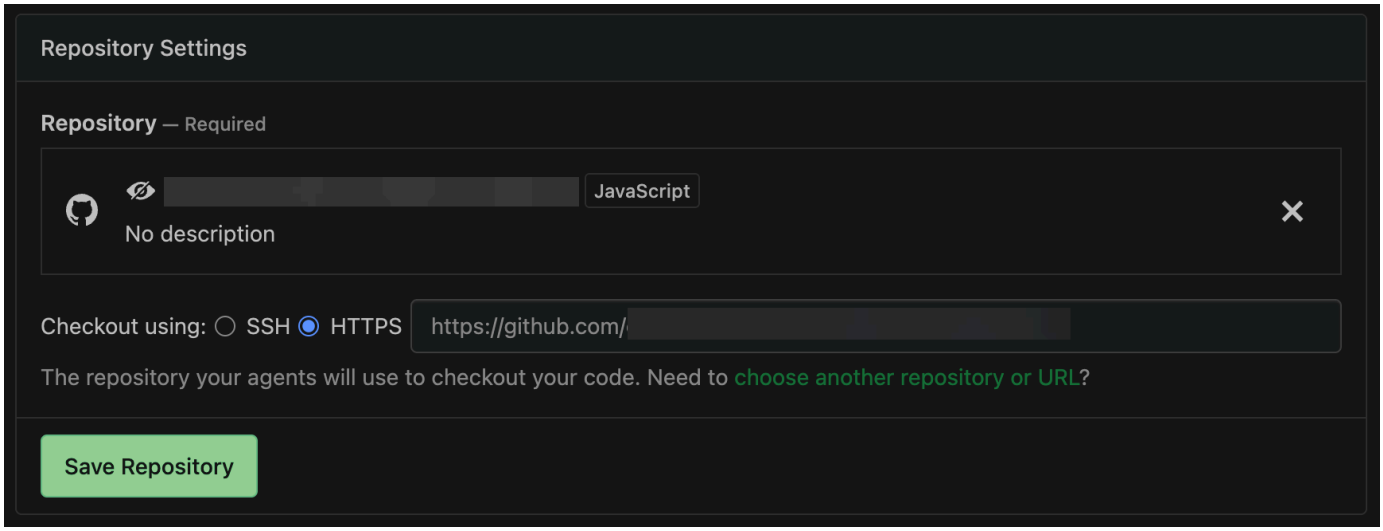
Mengautentikasi Buildkite ke Repositori Pribadi

Jika Anda memiliki repositori pribadi yang dikonfigurasi dalam pipeline Buildkite Anda, Buildkite memerlukan izin tambahan dalam lingkungan build untuk menarik repositori, karena Buildkite tidak menjual kredensil ke pelari yang dihosting sendiri untuk menarik dari repositori pribadi. Untuk mengautentikasi agen runner yang dihosting sendiri Buildkite ke repositori sumber pribadi eksternal Anda, Anda dapat menggunakan salah satu opsi berikut.

Untuk mengautentikasi dengan CodeBuild

CodeBuild menawarkan penanganan kredensil terkelola untuk tipe sumber yang didukung. Untuk menggunakan kredensi CodeBuild sumber untuk menarik repositori sumber pekerjaan Anda, Anda dapat menggunakan langkah-langkah berikut:

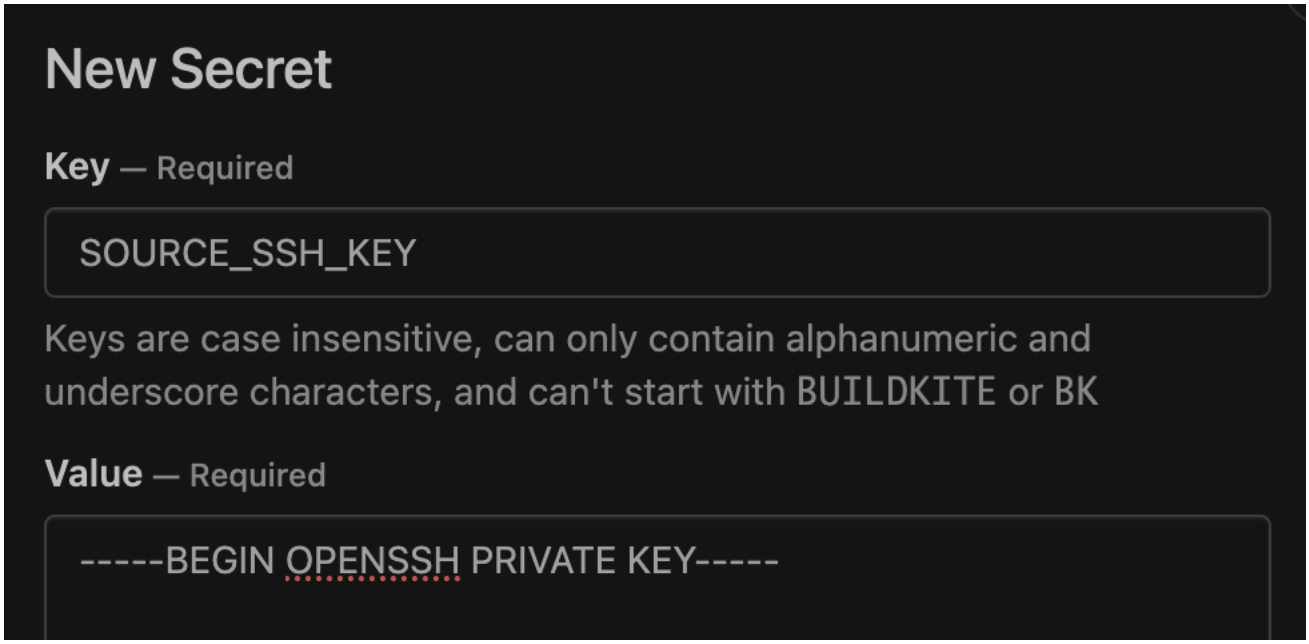
1. Di CodeBuild konsol, navigasikan ke Edit proyek atau buat CodeBuild proyek baru menggunakan langkah-langkah diLangkah 2: Buat CodeBuild proyek dengan webhook.
2. Di bawah opsi kredensi sumber Buildkite, pilih penyedia repositori sumber pekerjaan Anda.
    1. Jika Anda ingin menggunakan kredensil tingkat akun, verifikasi bahwa CodeBuild kredensialnya dikonfigurasi dengan benar. Selain itu, jika project Anda memiliki buildspec inline yang dikonfigurasi, verifikasi bahwa git-credential-helper diaktifkan.
    2. Jika Anda ingin menggunakan kredensial tingkat proyek, pilih Gunakan CodeBuild kredenal ganti untuk proyek ini saja dan siapkan kredensil untuk proyek Anda.
3. Di pengaturan pipeline Buildkite Anda, navigasikan ke Pengaturan Repositori. Atur pengaturan checkout repositori sumber Anda ke Checkout menggunakan HTTPS



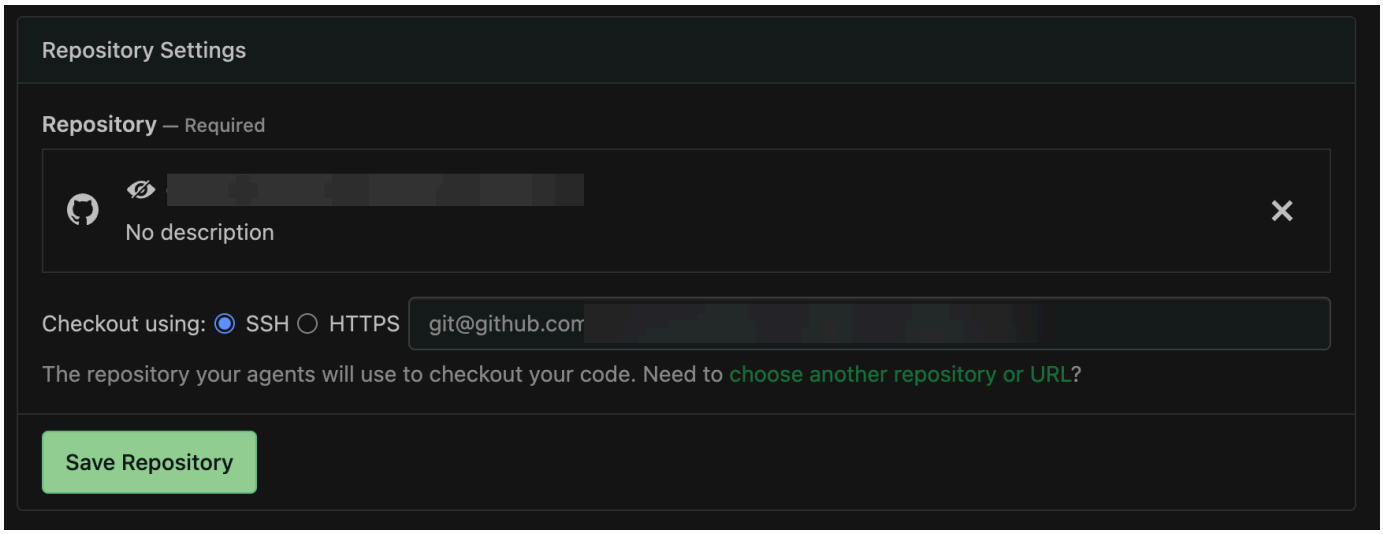
Untuk mengautentikasi dengan rahasia Buildkite

Buildkite memelihara plugin ssh-checkout yang dapat digunakan untuk mengautentikasi pelari yang dihosting sendiri ke repositori sumber eksternal menggunakan kunci ssh. Nilai kunci disimpan sebagai rahasia Buildkite dan diambil secara otomatis oleh agen runner yang di-host sendiri Buildkite saat mencoba menarik repositori pribadi. Untuk mengonfigurasi plugin ssh-checkout untuk pipeline Buildkite Anda, Anda dapat menggunakan langkah-langkah berikut:

1. Hasilkan kunci ssh pribadi dan publik menggunakan alamat email Anda, mis. `ssh-keygen -t rsa -b 4096 -C "myEmail@address.com"`
2. Tambahkan kunci publik ke repositori sumber pribadi Anda. Misalnya, Anda dapat mengikuti panduan ini untuk menambahkan kunci ke GitHub akun.
3. Tambahkan rahasia kunci SSH baru ke cluster Buildkite Anda. Dalam cluster Buildkite Anda, pilih Rahasia → Rahasia Baru. Tambahkan nama untuk rahasia Anda di bidang Kunci dan tambahkan kunci SSH pribadi Anda ke bidang Nilai:



4. Dalam pipeline Buildkite Anda, navigasikan ke pengaturan repositori Anda dan atur checkout untuk menggunakan SSH.



5. Perbarui langkah-langkah YAMAL pipeline Anda untuk menggunakan `git-ssh-checkout` plugin. Misalnya, file YAMAL pipeline berikut menggunakan tindakan checkout dengan kunci rahasia Buildkite di atas:

```
agents:
  project: "codebuild-myProject"
steps:
  - command: "npm run build"
    plugins:
      - git-ssh-checkout#v0.4.1:
```

```
            ssh-secret-key-name: 'SOURCE_SSH_KEY'
```

6. Saat menjalankan pekerjaan runner yang di-host sendiri Buildkite di dalamnya CodeBuild, Buildkite sekarang akan secara otomatis menggunakan nilai rahasia yang dikonfigurasi saat menarik repositori pribadi Anda

Opsi konfigurasi pelari

Anda dapat menentukan variabel lingkungan berikut dalam konfigurasi proyek untuk mengubah konfigurasi penyiapan pelari yang dihosting sendiri:

- `CODEBUILD_CONFIG_BUILDKITE_AGENT_TOKEN`: CodeBuild akan mengambil nilai rahasia yang dikonfigurasi sebagai nilai variabel lingkungan ini untuk mendaftarkan agen AWS Secrets Manager runner yang dihosting sendiri Buildkite. Variabel lingkungan ini harus bertipe`SECRETS_MANAGER`, dan nilainya harus menjadi nama rahasia Anda di Secrets Manager. Variabel lingkungan token agen Buildkite diperlukan untuk semua proyek runner Buildkite.
- `CODEBUILD_CONFIG_BUILDKITE_CREDENTIAL_DISABLE`: Secara default, CodeBuild akan memuat kredenal sumber tingkat akun atau proyek ke dalam lingkungan build, karena kredensil ini digunakan oleh agen Buildkite untuk menarik repositori sumber pekerjaan. Untuk menonaktifkan perilaku ini, Anda dapat menambahkan variabel lingkungan ini ke proyek Anda dengan nilai yang disetel ke`true`, yang akan mencegah kredensi sumber dimuat ke lingkungan build.

Jalankan perintah buildspec untuk fase INSTALL, PRE_BUILD, dan POST_BUILD

Secara default, CodeBuild abaikan perintah buildspec apa pun saat menjalankan build runner Buildkite yang dihosting sendiri. Untuk menjalankan perintah buildspec selama build,

```
buildspec-override: "true"
```

dapat ditambahkan sebagai akhiran pada label:

```
agents:
  project: "codebuild-<project name>"
  buildspec-override: "true"
```

Dengan menggunakan perintah ini, CodeBuild akan membuat folder yang disebut `buildkite-runner` dalam folder sumber utama wadah. Ketika pelari Buildkite dimulai selama `BUILD` fase, pelari akan berjalan di direktori. `buildkite-runner`

Ada beberapa batasan saat menggunakan penggantian buildspec dalam build Buildkite yang dihosting sendiri:

- Agen Buildkite mengharuskan kredensi sumber ada di dalam lingkungan build untuk menarik repositori sumber pekerjaan. Jika Anda menggunakan kredensi CodeBuild sumber untuk autentikasi, Anda harus mengaktifkan `git-credential-helper` di buildspec Anda. Misalnya, Anda dapat menggunakan buildspec berikut `git-credential-helper` untuk mengaktifkan build Buildkite Anda:

  ```
  version: 0.2
  env:
    git-credential-helper: yes
  phases:
    pre_build:
      commands:
         - echo "Hello World"
  ```

- CodeBuild tidak akan menjalankan perintah buildspec selama `BUILD` fase, karena pelari yang di-host sendiri berjalan dalam fase tersebut. `BUILD`
- CodeBuild tidak mendukung file buildspec untuk build runner Buildkite. Hanya spesifikasi build inline yang didukung untuk runner yang di-host sendiri Buildkite
- Jika perintah build gagal di `INSTALL` fase `PRE_BUILD` atau, tidak CodeBuild akan memulai runner yang dihosting sendiri dan pekerjaan Buildkite harus dibatalkan secara manual.

Menyiapkan pelari Buildkite secara terprogram

Untuk mengonfigurasi proyek runner Buildkite secara terprogram, Anda perlu mengonfigurasi sumber daya berikut:

Untuk membuat runner Buildkite secara terprogram

1. Buat token agen Buildkite dan simpan token dalam teks biasa di dalamnya. AWS Secrets Manager
2. Siapkan CodeBuild proyek dengan konfigurasi pilihan Anda. Anda perlu mengkonfigurasi atribut tambahan berikut:

   1. Nilai lingkungan dengan nama`CODEBUILD_CONFIG_BUILDKITE_AGENT_TOKEN`, tipe`SECRETS_MANAGER`, dan nilai yang sama dengan token agen Buildkite yang terkait dengan cluster Buildkite Anda.

2. Jenis sumber sama dengan `NO_SOURCE`
3. Izin untuk mengakses rahasia yang dibuat pada langkah 1 dalam peran layanan proyek Anda

Misalnya, Anda dapat menggunakan perintah berikut untuk membuat proyek runner Buildkite yang valid melalui CLI:

```
aws codebuild create-project \
--name buildkite-runner-project \
--source "{\"type\": \"NO_SOURCE\",\"buildspec\":\"\"}" \
--environment "{\"image\":\"aws/codebuild/amazonlinux-x86_64-standard:5.0\",
\"type\":\"LINUX_CONTAINER\",\"computeType\":\"BUILD_GENERAL1_MEDIUM\",
\"environmentVariables\":[{\"name\":\"CODEBUILD_CONFIG_BUILDKITE_AGENT_TOKEN\",
\"type\":\"SECRETS_MANAGER\",\"value\":\"<buildkite-secret-name>\"}]}" \
--artifacts "{\"type\": \"NO_ARTIFACTS\"}" \
--service-role <service-role>
```

3. Buat webhook runner Buildkite pada proyek yang dibuat pada langkah 2. Anda harus menggunakan opsi konfigurasi berikut saat membuat webhook:

   1. build-type harus sama dengan `RUNNER_BUILDKITE_BUILD`
   2. Filter dengan tipe `EVENT` dan pola yang sama dengan `WORKFLOW_JOB_QUEUED`

   Misalnya, Anda dapat menggunakan perintah berikut untuk membuat webhook runner Buildkite yang valid melalui CLI:

   ```
   aws codebuild create-webhook \
   --project-name buildkite-runner-project \
   --filter-groups "[[{\"type\":\"EVENT\",\"pattern\":\"WORKFLOW_JOB_QUEUED\"}]]" \
   --build-type RUNNER_BUILDKITE_BUILD
   ```

4. Simpan URL Payload dan nilai Rahasia yang ditampilkan oleh `create-webhook` panggilan dan gunakan kredensialnya untuk membuat webhook di dalam konsol Buildkite. Anda dapat mereferensikan Langkah 3: Buat CodeBuild webhook dalam Buildkite Tutorial: Konfigurasikan CodeBuild-hosted pelari Buildkite untuk panduan tentang cara mengatur sumber daya ini.

Memecahkan masalah webhook untuk build yang gagal atau pekerjaan yang menggantung

Masalah:

Webhook yang Anda atur Tutorial: Konfigurasikan CodeBuild-hosted pelari Buildkite tidak berfungsi atau pekerjaan alur kerja Anda tergantung di Buildkite.

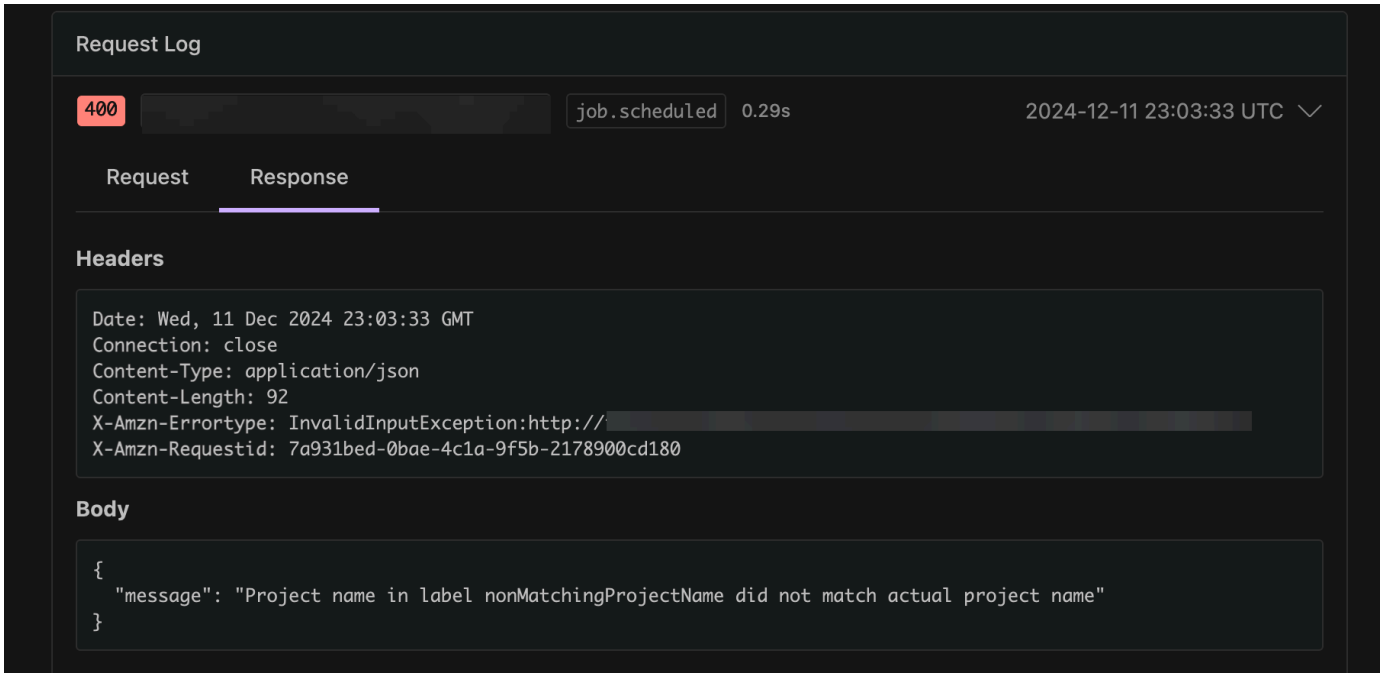
Kemungkinan penyebabnya:

- Acara job.scheduled webhook Anda mungkin gagal memicu build. Tinjau log Respons untuk melihat respons atau pesan kesalahan.
- CodeBuild Build Anda gagal sebelum memulai agen runner yang dihosting sendiri Buildkite untuk menangani pekerjaan Anda.

Solusi yang direkomendasikan:

Untuk men-debug peristiwa webhook Buildkite yang gagal:

1. Di pengaturan organisasi Buildkite Anda, navigasikan ke Layanan Pemberitahuan, pilih CodeBuild webhook Anda, lalu temukan Log Permintaan.
2. Temukan acara `job.scheduled` webhook yang terkait dengan pekerjaan Buildkite Anda yang macet. Anda dapat menggunakan bidang ID pekerjaan dalam payload webhook untuk menghubungkan peristiwa webhook dengan pekerjaan Buildkite Anda.
3. Pilih tab Respons dan periksa badan respons. Verifikasi bahwa kode status Response adalah `200` dan badan Response tidak berisi pesan yang tidak terduga.



Memecahkan masalah izin webhook

Masalah:

Pekerjaan Buildkite gagal memeriksa repositori sumber pekerjaan karena masalah izin.

Kemungkinan penyebabnya:

- CodeBuild tidak memiliki izin yang cukup untuk memeriksa repositori sumber pekerjaan.
- Pengaturan repositori pipeline diatur untuk memeriksa menggunakan SSH untuk CodeBuild kredensi terkelola.

Solusi yang direkomendasikan:

- Verifikasi bahwa CodeBuild memiliki izin yang cukup dikonfigurasi untuk memeriksa repositori sumber pekerjaan. Selain itu, verifikasi bahwa peran layanan CodeBuild proyek Anda memiliki izin yang cukup untuk mengakses opsi izin sumber yang dikonfigurasi.
- Verifikasi bahwa pipeline Buildkite Anda dikonfigurasi untuk menggunakan checkout menggunakan HTTPS jika Anda menggunakan kredenal repositori sumber CodeBuild terkelola.

Penggantian label didukung dengan runner CodeBuild-hosted Buildkite

Di label tag agen langkah pipeline Buildkite, Anda dapat memberikan berbagai penggantian label yang mengubah build runner yang dihosting sendiri. Setiap build yang tidak dikenali oleh CodeBuild akan diabaikan tetapi tidak akan gagal permintaan webhook Anda. Misalnya, alur kerja berikut YAMAL menyertakan penggantian untuk image, ukuran instance, armada, dan buildspec:

```
agents:
  queue: "myQueue"
steps:
  - command: "echo \"Hello World\""
    agents:
      project: "codebuild-myProject"
      image: "{{matrix.os}}"
      instance-size: "{{matrix.size}}"
      buildspec-override: "true"
    matrix:
      setup:
        os:
          - "arm-3.0"
          - "al2-5.0"
        size:
          - "small"
          - "large"
```

`project:codebuild-<project-name>` (wajib)

- Contoh: `project: "codebuild-myProject"`
- Diperlukan untuk semua konfigurasi langkah pipeline Buildkite. *<project name>*harus sama dengan nama proyek tempat webhook pelari yang dihosting sendiri dikonfigurasi.

`queue: "<queue-name>"`

- Contoh: `queue: "<queue-name>"`
- Digunakan untuk merutekan pekerjaan Buildkite ke antrian tertentu. Lihat Tag Antrian Agen Buildkite untuk informasi selengkapnya.

`image: "<environment-type>-<image-identifier>"`

- Contoh: `image: "arm-3.0"`

- Mengganti jenis gambar dan lingkungan yang digunakan saat memulai build runner yang dihosting sendiri dengan gambar yang dikurasi. Untuk mempelajari nilai yang didukung, lihatHitung gambar yang didukung dengan runner CodeBuild-hosted Buildkite.
  1. Untuk mengganti jenis gambar dan lingkungan yang digunakan dengan gambar khusus, gunakan `image: "custom-<environment-type>-<custom-image-identifier>"`
  2. Contoh:

```
image:
      "custom-arm-public.ecr.aws/codebuild/amazonlinux-aarch64-standard:3.0"
```

> **Note**
>
> Jika gambar kustom berada di registri pribadi, Anda harus mengonfigurasi kredensi registri yang sesuai dalam proyek Anda. CodeBuild

`instance-size: "<instance-size>"`

- Contoh: `instance-size: "medium"`
- Mengganti jenis instance yang digunakan saat memulai build runner yang dihosting sendiri. Untuk mempelajari nilai yang didukung, lihatHitung gambar yang didukung dengan runner CodeBuild-hosted Buildkite.

`fleet: "<fleet-name>"`

- Contoh: `fleet: "myFleet"`
- Mengganti pengaturan armada yang dikonfigurasi pada proyek Anda untuk menggunakan armada yang ditentukan. Untuk informasi selengkapnya, lihat Menjalankan build di atas armada kapasitas cadangan.

`buildspec-override: "<boolean>"`

- Contoh: `buildspec-override: "true"`
- Mengizinkan build menjalankan perintah buildspec di`INSTALL`,`PRE_BUILD`, dan `POST_BUILD` fase jika disetel ke. `true`

Hitung gambar yang didukung dengan runner CodeBuild-hosted Buildkite

Pada label yang Anda konfigurasikanSelf-managed Pelari buildkite di AWS CodeBuild, Anda dapat mengganti pengaturan lingkungan Amazon EC2 dengan menggunakan nilai di tiga kolom pertama. CodeBuild menyediakan gambar komputasi Amazon EC2 berikut. Untuk informasi lebih lanjut tentang

Tipe lingkungan	Pengidentifikasi gambar	Ukuran instans	Platform	Gambar terselesaikan	Definisi

`linux`	`4.0`	`small`
		`medium`
		`large`
		`xlarge`
			Amazon Linux 2	`aws/codebuild/amazonlinux-x86_64-standard:4.0`	al/standard/4.0

`linux`	`5.0`	`2xlarge`
		`gpu_small`
		`gpu_large`
			Amazon Linux 2023	`aws/codebuild/amazonlinux-x86_64-standard:5.0`	al/standard/5.0

`linux-ec2`	`latest`	`small`
		`medium`
		`large`
			Amazon Linux 2023	`aws/codebuild/ami/amazonlinux-x86_64-base:latest`	Tidak ada

`arm`	`2.0`	`small`
		`medium`
		`large`
		`xlarge`
		`2xlarge`
			Amazon Linux 2	`aws/codebuild/amazonlinux-aarch64-standard:2.0`	al/aarch64/standard/2.0

Tipe lingkungan	Pengidentifikasi gambar	Ukuran instans	Platform	Gambar terselesaikan	Definisi
`arm`	`3.0`		Amazon Linux 2023	`aws/codebuild/amazonlinux-aarch64-standard:3.0`	al/aarch64/standard/3.0

`arm-ec2`	`latest`	`small`
		`medium`
		`large`
			Amazon Linux 2023	`aws/codebuild/ami/amazonlinux-arm-base:latest`	Tidak ada

`ubuntu`	`5.0`	`small`
		`medium`
		`large`
		`xlarge`
		`2xlarge`
		`gpu_small`
		`gpu_large`
			Ubuntu 20.04	`aws/codebuild/standard:5.0`	ubuntu/standard/5.0

`ubuntu`	`6.0`		Ubuntu 22.04	`aws/codebuild/standard:6.0`	ubuntu/standard/6.0
`ubuntu`	`7.0`		Ubuntu 22.04	`aws/codebuild/standard:7.0`	ubuntu/standard/7.0
`windows`	`1.0`	`medium` `large`	Inti Server Windows 2019	`aws/codebuild/windows-base:2019-1.0`	N/A

Tipe lingkungan	Pengidentifikasi gambar	Ukuran instans	Platform	Gambar terselesaikan	Definisi
			Inti Server Windows 2022	`aws/codebuild/windows-base:2022-1.0`	N/A
`windows`	`2.0`		Inti Server Windows 2019	`aws/codebuild/windows-base:2019-2.0`	N/A
`windows`	`3.0`		Inti Server Windows 2019	`aws/codebuild/windows-base:2019-3.0`	N/A
`windows-ec2`	`2022`	`medium` `large`	Inti Server Windows 2022	`aws/codebuild/ami/windows-base:2022`	Tidak ada

Selain itu, Anda dapat mengganti pengaturan lingkungan Lambda Anda dengan menggunakan nilai berikut. Untuk informasi selengkapnya tentang komputasi CodeBuild Lambda, lihat. Jalankan dibangun di AWS Lambda komputasi CodeBuild mendukung gambar komputasi Lambda berikut:

Tipe lingkungan	Pengidentifikasi gambar	Ukuran instans

`linux-lambda`	`dotnet6`
	`go1.21`
	`corretto11`
		`1GB`
		`2GB`
		`4GB`
		`8GB`

`arm-lambda`		

Tipe lingkungan	Pengidentifikasi gambar	Ukuran instans			

	`corretto17`
	
	`corretto21`
	
	`nodejs18`
	
	`nodejs20`
	
	`python3.11`
	
	`python3.12`
	
	`ruby3.2`
		`10GB`			

Lihat informasi yang lebih lengkap di Membangun mode dan tipe komputasi lingkungan dan Gambar Docker disediakan oleh CodeBuild.

Gunakan webhook dengan AWS CodeBuild

AWS CodeBuild mendukung integrasi webhook dengan GitHub, Server GitHub Perusahaan, Managed GitLab Man GitLab diri, dan Bitbucket.

Topik

- Praktik terbaik untuk menggunakan webhook dengan AWS CodeBuild
- Acara webhook Bitbucket
- GitHub webhook global dan organisasi
- GitHub webhook manual
- GitHub acara webhook
- GitLab kelompok webhooks
- GitLab webhook manual

- GitLab acara webhook
- Buildkite webhook manual
- Persetujuan komentar permintaan tarik

Praktik terbaik untuk menggunakan webhook dengan AWS CodeBuild

Untuk proyek yang menggunakan repositori publik untuk mengatur webhook, kami merekomendasikan opsi berikut:

Menyiapkan `ACTOR_ACCOUNT_ID` filter

Tambahkan `ACTOR_ACCOUNT_ID` filter ke grup filter webhook proyek Anda untuk menentukan pengguna mana yang dapat memicu build. Setiap acara webhook yang dikirimkan CodeBuild dilengkapi dengan informasi pengirim yang menentukan pengenal aktor. CodeBuild akan memfilter webhook berdasarkan pola ekspresi reguler yang disediakan dalam filter. Anda dapat menentukan pengguna tertentu yang diizinkan untuk memicu build dengan filter ini. Untuk informasi selengkapnya, lihat GitHub acara webhook dan Acara webhook Bitbucket.

Menyiapkan `FILE_PATH` filter

Tambahkan `FILE_PATH` filter ke grup filter webhook proyek Anda untuk menyertakan atau mengecualikan file yang dapat memicu build saat diubah. Misalnya, Anda dapat menolak permintaan build untuk perubahan pada `buildspec.yml` file menggunakan pola ekspresi reguler seperti`^buildspec.yml$`, bersama dengan `excludeMatchedPattern` properti. Untuk informasi selengkapnya, lihat GitHub acara webhook dan Acara webhook Bitbucket.

Cakupan izin untuk peran IAM build Anda

Build yang dipicu oleh webhook menggunakan peran layanan IAM yang ditentukan dalam proyek. Sebaiknya atur izin dalam peran layanan ke kumpulan izin minimum yang diperlukan untuk menjalankan build. Misalnya, dalam skenario pengujian dan penerapan, buat satu proyek untuk pengujian dan proyek lain untuk penerapan. Proyek pengujian menerima build webhook dari repositori, tetapi tidak memberikan izin menulis ke sumber daya Anda. Proyek penerapan memberikan izin menulis ke sumber daya Anda, dan filter webhook dikonfigurasi untuk hanya mengizinkan pengguna tepercaya memicu build.

Gunakan spesifikasi buildspec yang disimpan sebaris atau Amazon S3

Jika Anda menentukan buildspec sebaris di dalam proyek itu sendiri, atau menyimpan file buildspec di bucket Amazon S3, file buildspec hanya dapat dilihat oleh pemilik proyek. Ini

mencegah permintaan tarik membuat perubahan kode pada file buildspec dan memicu build yang tidak diinginkan. Untuk informasi selengkapnya, lihat ProjectSource.buildspec dalam Referensi API CodeBuild.

Acara webhook Bitbucket

Anda dapat menggunakan grup filter webhook untuk menentukan peristiwa webhook Bitbucket mana yang memicu build. Misalnya, Anda dapat menentukan bahwa build hanya dipicu untuk perubahan pada cabang tertentu.

Anda dapat membuat satu atau beberapa grup filter webhook untuk menentukan peristiwa webhook mana yang memicu build. Build dipicu jika ada grup filter yang dievaluasi menjadi true, yang terjadi ketika semua filter dalam grup mengevaluasi ke true. Ketika Anda membuat grup filter, tentukan:

Sebuah acara

Untuk Bitbucket, Anda dapat memilih satu atau beberapa acara berikut:

- `PUSH`
- `PULL_REQUEST_CREATED`
- `PULL_REQUEST_UPDATED`
- `PULL_REQUEST_MERGED`
- `PULL_REQUEST_CLOSED`

Jenis acara webhook ada di header di `X-Event-Key` lapangan. Tabel berikut menunjukkan bagaimana nilai `X-Event-Key` header dipetakan ke jenis acara.

> **Note**
>
> Anda harus mengaktifkan `merged` acara dalam pengaturan webhook Bitbucket jika Anda membuat grup filter webhook yang menggunakan jenis peristiwa. `PULL_REQUEST_MERGED` Anda juga harus mengaktifkan `declined` acara dalam pengaturan webhook Bitbucket jika Anda membuat grup filter webhook yang menggunakan jenis peristiwa. `PULL_REQUEST_CLOSED`

`X-Event-Key` Nilai header	Tipe peristiwa
`repo:push`	`PUSH`
`pullrequest:created`	`PULL_REQUEST_CREATED`
`pullrequest:updated`	`PULL_REQUEST_UPDATED`
`pullrequest:fulfilled`	`PULL_REQUEST_MERGED`
`pullrequest:rejected`	`PULL_REQUEST_CLOSED`

KarenaPULL_REQUEST_MERGED, jika permintaan tarik digabungkan dengan strategi squash dan cabang permintaan tarik ditutup, komit permintaan tarik asli tidak ada lagi. Dalam hal ini, variabel CODEBUILD_WEBHOOK_MERGE_COMMIT lingkungan berisi pengidentifikasi komit gabungan yang dihancurkan.

Satu atau lebih filter opsional

Gunakan ekspresi reguler untuk menentukan filter. Agar suatu peristiwa memicu build, setiap filter dalam grup yang terkait dengannya harus dievaluasi menjadi true.

CodeBuild mengevaluasi pola filter menggunakan sintaks ekspresi RE2 reguler, yang tidak mendukung lookahead, lookbehind, backreference, atau grup atom. Untuk informasi selengkapnya, lihat halaman RE2 Sintaks di GitHub situs web.

`ACTOR_ACCOUNT_ID`(`ACTOR_ID`di konsol)

Peristiwa webhook memicu build ketika ID akun Bitbucket cocok dengan pola ekspresi reguler. Nilai ini muncul di `account_id` properti `actor` objek di payload filter webhook.

`HEAD_REF`

Peristiwa webhook memicu build ketika referensi kepala cocok dengan pola ekspresi reguler (misalnya, `refs/heads/branch-name` dan`refs/tags/tag-name`). `HEAD_REF`Filter mengevaluasi nama referensi Git untuk cabang atau tag. Nama cabang atau tag muncul di `name` bidang `new` objek di `push` objek payload webhook. Untuk peristiwa pull request, nama cabang muncul di `name` bidang di `branch` objek `source` objek di payload webhook.

`BASE_REF`

Peristiwa webhook memicu build ketika referensi dasar cocok dengan pola ekspresi reguler. Filter `BASE_REF` dapat digunakan untuk peristiwa permintaan pull saja (misalnya, `refs/heads/branch-name`). `BASE_REF`Filter mengevaluasi nama referensi Git untuk cabang. Nama cabang muncul di `name` bidang `branch` objek di `destination` objek di payload webhook.

`FILE_PATH`

Webhook memicu build ketika jalur file yang diubah cocok dengan pola ekspresi reguler.

> **Note**
>
> `FILE_PATH`Filter dievaluasi terhadap 3.000 file pertama yang diubah dalam permintaan push atau pull. Untuk memastikan filter diterapkan sebagaimana dimaksud, simpan jumlah file yang diubah dalam batas ini. Untuk peristiwa pull request, sebaiknya gunakan kebijakan build pull request untuk mengontrol permintaan tarik mana yang dapat memulai build.

`COMMIT_MESSAGE`

Webhook memicu build ketika pesan komit kepala cocok dengan pola ekspresi reguler.

`WORKFLOW_NAME`

Webhook memicu build ketika nama alur kerja cocok dengan pola ekspresi reguler.

> **Note**
>
> Anda dapat menemukan payload webhook di pengaturan webhook repositori Bitbucket Anda.

Topik

- Filter peristiwa webhook Bitbucket (konsol)
- Filter peristiwa webhook Bitbucket (SDK)
- Filter acara webhook Bitbucket (CloudFormation)

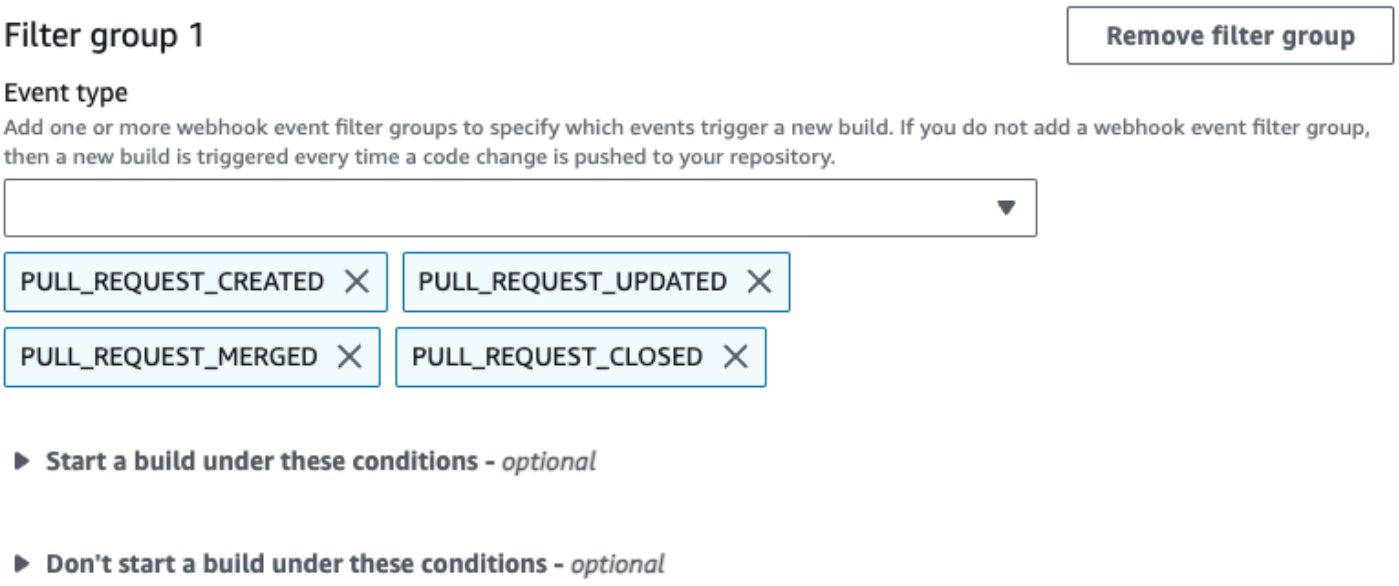
Filter peristiwa webhook Bitbucket (konsol)

Untuk menggunakan Konsol Manajemen AWS untuk memfilter peristiwa webhook:

1. Pilih Bangun kembali setiap kali perubahan kode didorong ke repositori ini saat Anda membuat proyek Anda.
2. Dari Jenis acara, pilih satu atau beberapa acara.
3. Untuk memfilter saat peristiwa memicu build, di bawah Mulai build dalam kondisi ini, tambahkan satu atau beberapa filter opsional.
4. Untuk memfilter saat peristiwa tidak dipicu, di bawah Jangan memulai build dalam kondisi ini, tambahkan satu atau beberapa filter opsional.
5. Pilih Tambahkan grup filter untuk menambahkan grup filter lain.

Untuk informasi selengkapnya, lihat Buat proyek build (konsol) dan WebhookFilter di Refer AWS CodeBuild ensi API.

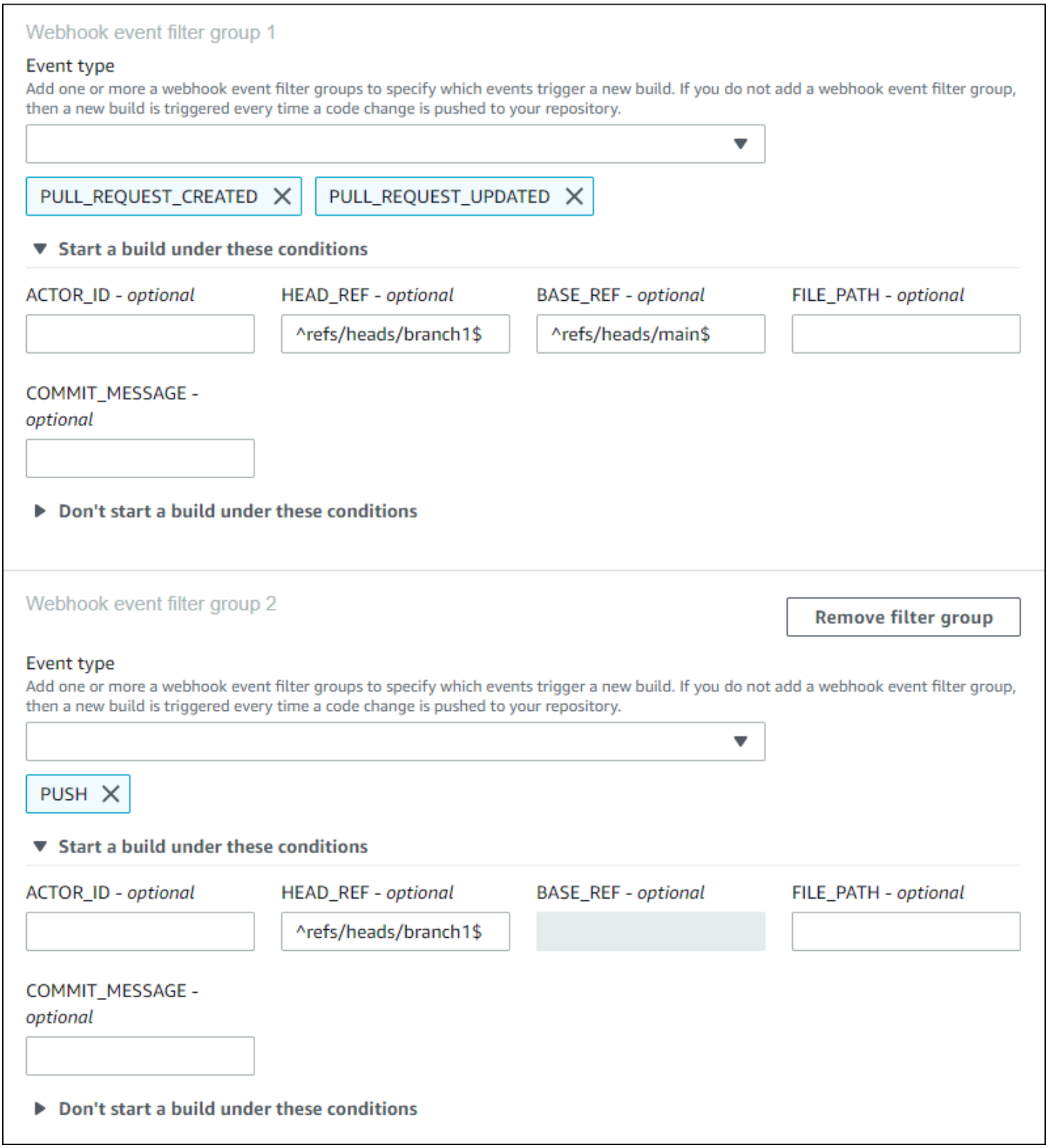
Dalam contoh ini, grup filter webhook memicu build untuk permintaan tarik saja:



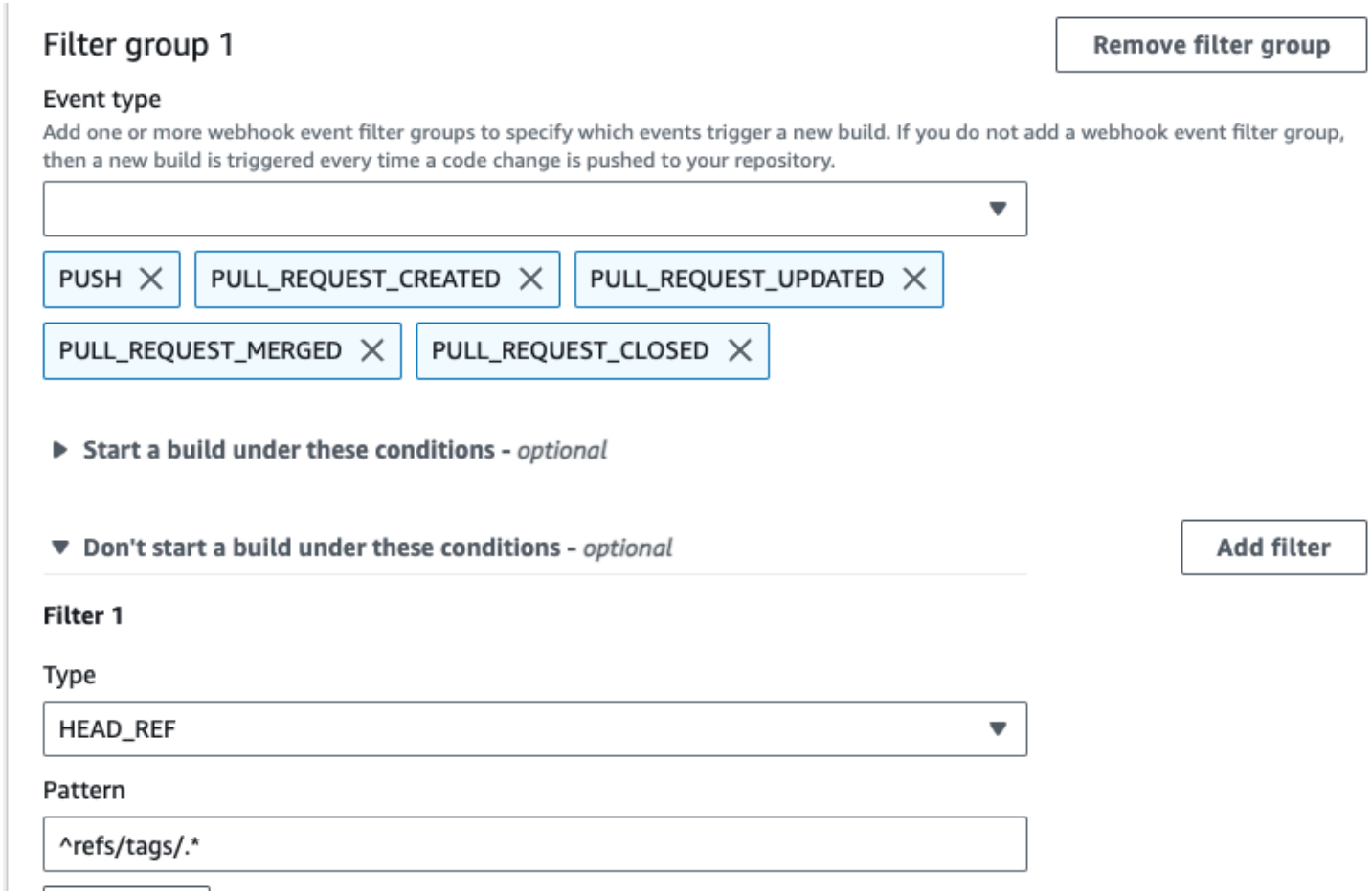
Menggunakan contoh dua grup filter, build dipicu ketika salah satu atau keduanya mengevaluasi menjadi true:

- Grup filter pertama menentukan permintaan tarik yang dibuat atau diperbarui pada cabang dengan nama referensi Git yang cocok dengan ekspresi reguler `^refs/heads/main$` dan referensi kepala yang cocok`^refs/heads/branch1!`.

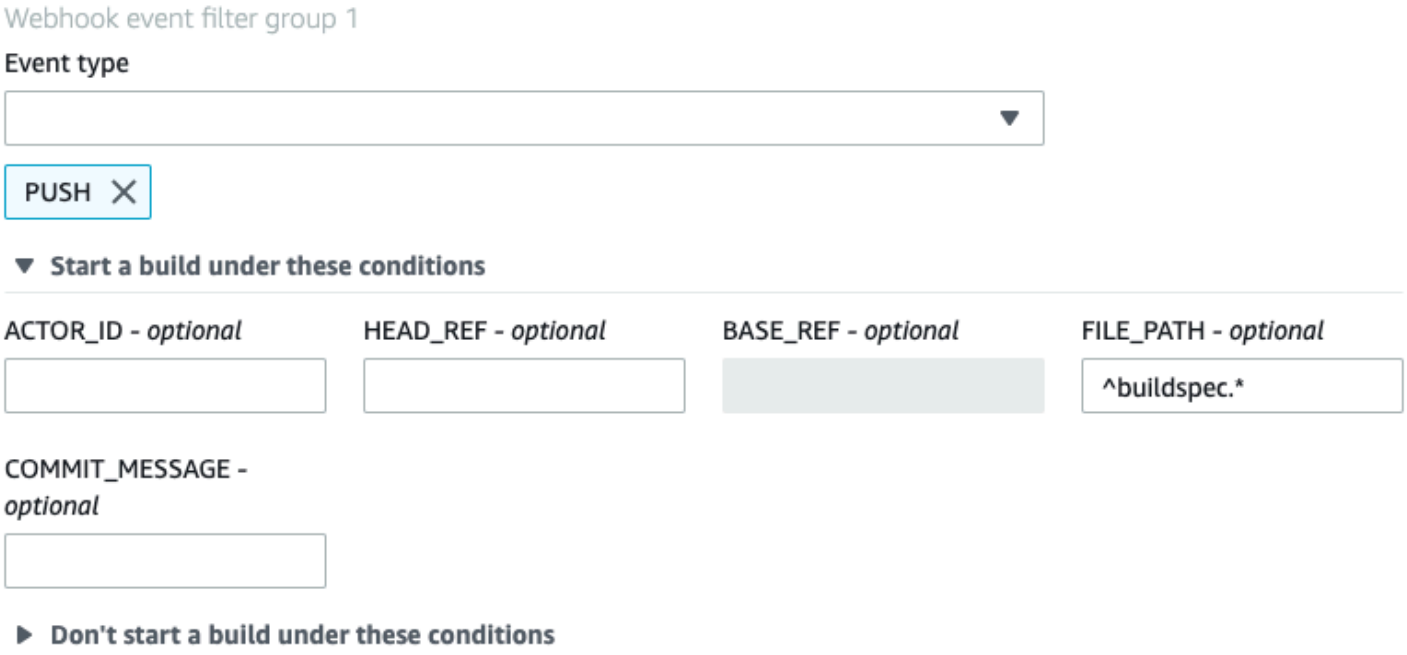
- Grup filter kedua menentukan permintaan push pada cabang dengan nama referensi Git yang cocok dengan ekspresi reguler`^refs/heads/branch1$`.



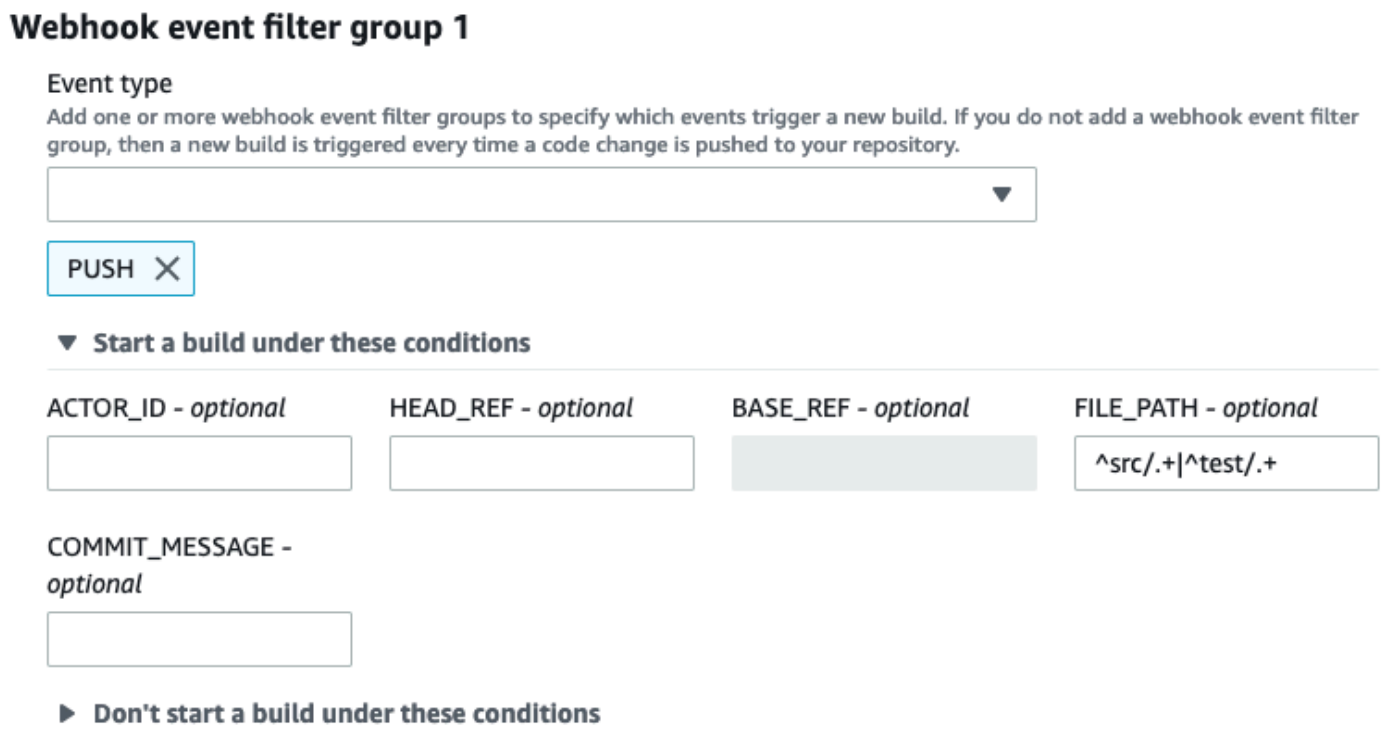
Dalam contoh ini, grup filter webhook memicu build untuk semua permintaan kecuali peristiwa tag.



Dalam contoh ini, grup filter webhook memicu build hanya ketika file dengan nama yang cocok dengan ekspresi reguler `^buildspec.*` berubah.

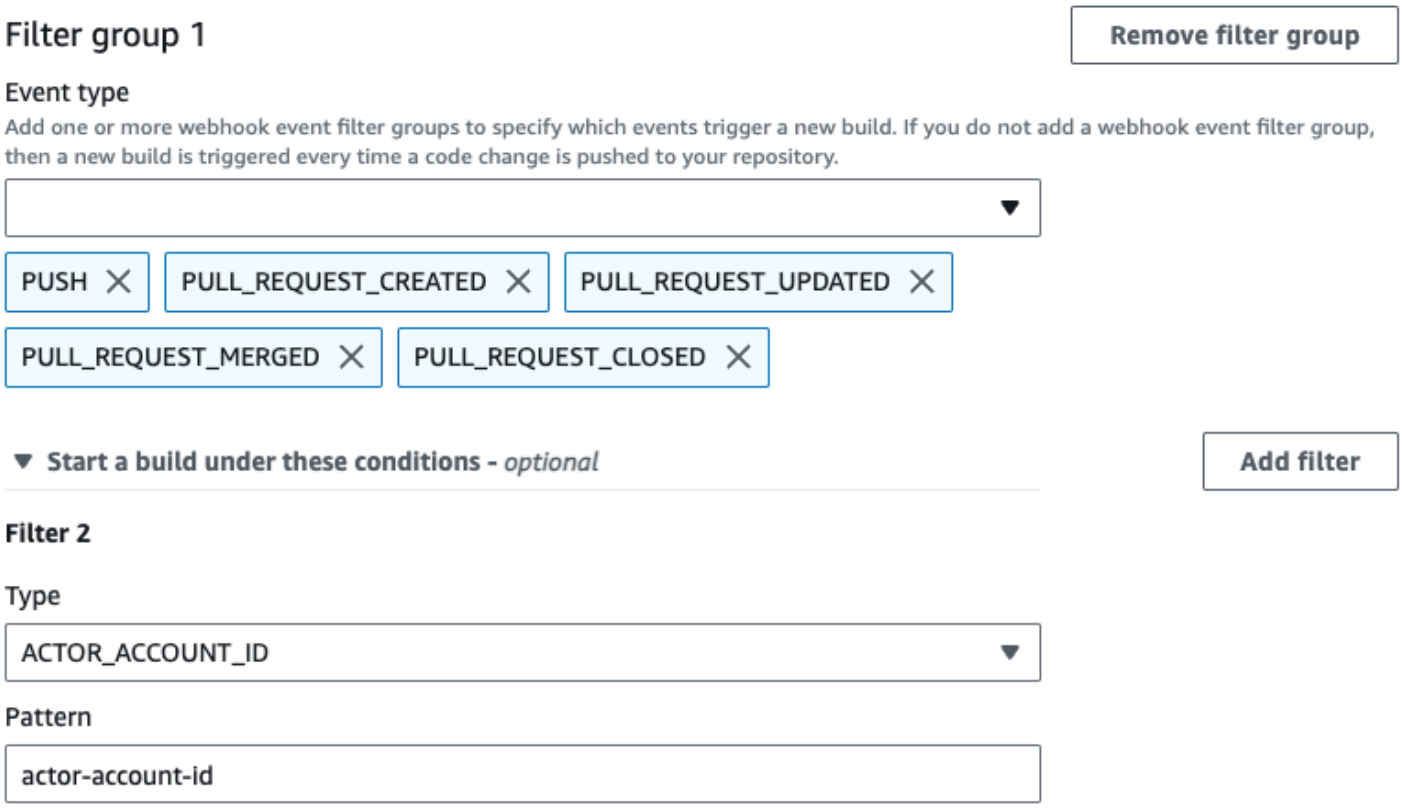


Dalam contoh ini, grup filter webhook memicu build hanya ketika file diubah di dalam `test` folder `src` atau.

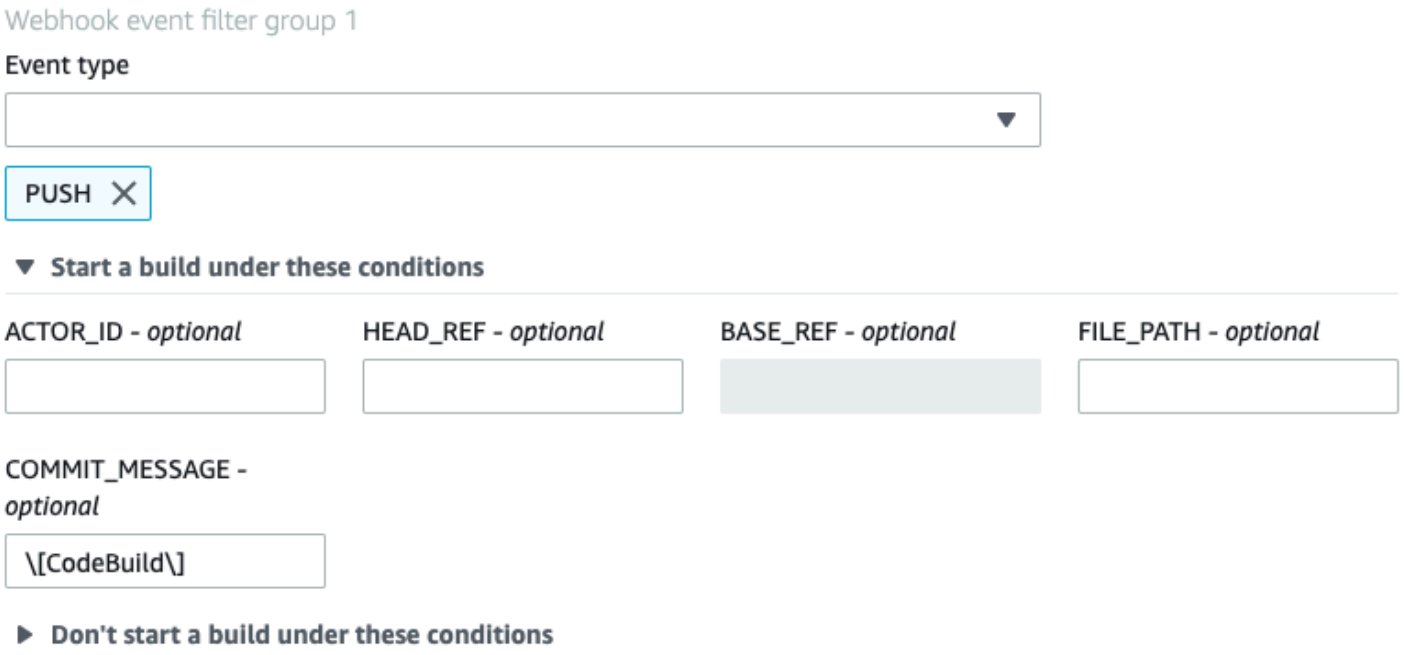


Dalam contoh ini, grup filter webhook memicu build hanya ketika perubahan dilakukan oleh pengguna Bitbucket yang tidak memiliki ID akun yang cocok dengan ekspresi reguler. `actor-account-id`

> **Note**
>
> Untuk informasi tentang cara menemukan ID akun Bitbucket Anda, lihat https://api.bitbucket.org/2.0/users/*user-name*, di *user-name* mana nama pengguna Bitbucket Anda.



Dalam contoh ini, grup filter webhook memicu build untuk acara push ketika pesan komit kepala cocok dengan ekspresi `\[CodeBuild\]` reguler.



Filter peristiwa webhook Bitbucket (SDK)

Untuk menggunakan AWS CodeBuild SDK untuk memfilter peristiwa webhook, gunakan `filterGroups` bidang dalam sintaks permintaan metode `UpdateWebhook` API `CreateWebhook` atau. Untuk informasi selengkapnya, lihat WebhookFilter di dalam Referensi API CodeBuild .

Untuk membuat filter webhook yang memicu build hanya untuk permintaan tarik, masukkan yang berikut ini ke dalam sintaks permintaan:

```
"filterGroups": [
  [
    {
      "type": "EVENT",
      "pattern": "PULL_REQUEST_CREATED, PULL_REQUEST_UPDATED, PULL_REQUEST_MERGED,
 PULL_REQUEST_CLOSED"
    }
  ]
]
```

Untuk membuat filter webhook yang memicu build untuk cabang tertentu saja, gunakan `pattern` parameter untuk menentukan ekspresi reguler untuk memfilter nama cabang. Menggunakan contoh dua grup filter, build dipicu ketika salah satu atau keduanya mengevaluasi menjadi true:

- Grup filter pertama menentukan permintaan tarik yang dibuat atau diperbarui pada cabang dengan nama referensi Git yang cocok dengan ekspresi reguler `^refs/heads/main$` dan referensi kepala yang cocok`^refs/heads/myBranch$`.
- Grup filter kedua menentukan permintaan push pada cabang dengan nama referensi Git yang cocok dengan ekspresi reguler`^refs/heads/myBranch$`.

```
"filterGroups": [
  [
    {
      "type": "EVENT",
      "pattern": "PULL_REQUEST_CREATED, PULL_REQUEST_UPDATED, PULL_REQUEST_CLOSED"
    },
    {
      "type": "HEAD_REF",
      "pattern": "^refs/heads/myBranch$"
    },
    {
      "type": "BASE_REF",
      "pattern": "^refs/heads/main$"
    }
  ],
  [
    {
      "type": "EVENT",
      "pattern": "PUSH"
    },
    {
      "type": "HEAD_REF",
      "pattern": "^refs/heads/myBranch$"
    }
  ]
]
```

Anda dapat menggunakan `excludeMatchedPattern` parameter untuk menentukan peristiwa mana yang tidak memicu build. Dalam contoh ini, build dipicu untuk semua permintaan kecuali peristiwa tag.

```
"filterGroups": [
  [
    {
      "type": "EVENT",
```

```
      "pattern": "PUSH, PULL_REQUEST_CREATED, PULL_REQUEST_UPDATED,
 PULL_REQUEST_MERGED, PULL_REQUEST_CLOSED"
    },
    {
      "type": "HEAD_REF",
      "pattern": "^refs/tags/.*",
      "excludeMatchedPattern": true
    }
  ]
]
```

Anda dapat membuat filter yang memicu build hanya ketika perubahan dilakukan oleh pengguna Bitbucket dengan ID `actor-account-id` akun.

> **Note**
>
> Untuk informasi tentang cara menemukan ID akun Bitbucket Anda, lihat https://api.bitbucket.org/2.0/users/*user-name*, di *user-name* mana nama pengguna Bitbucket Anda.

```
"filterGroups": [
  [
    {
      "type": "EVENT",
      "pattern": "PUSH, PULL_REQUEST_CREATED, PULL_REQUEST_UPDATED,
 PULL_REQUEST_MERGED, PULL_REQUEST_CLOSED"
    },
    {
      "type": "ACTOR_ACCOUNT_ID",
      "pattern": "actor-account-id"
    }
  ]
]
```

Anda dapat membuat filter yang memicu build hanya ketika file dengan nama yang cocok dengan ekspresi reguler dalam `pattern` argumen berubah. Dalam contoh ini, grup filter menentukan bahwa build dipicu hanya ketika file dengan nama yang cocok dengan ekspresi reguler `^buildspec.*` berubah.

```
"filterGroups": [
```

```
  [
    {
      "type": "EVENT",
      "pattern": "PUSH"
    },
    {
      "type": "FILE_PATH",
      "pattern": "^buildspec.*"
    }
  ]
]
```

Dalam contoh ini, grup filter menentukan bahwa build dipicu hanya ketika file diubah dalam `test` folder `src` atau.

```
"filterGroups": [
    [
        {
            "type": "EVENT",
            "pattern": "PUSH"
        },
        {
            "type": "FILE_PATH",
            "pattern": "^src/.+|^test/.+"
        }
    ]
]
```

Anda dapat membuat filter yang memicu build hanya ketika pesan komit kepala cocok dengan ekspresi reguler dalam argumen pola. Dalam contoh ini, grup filter menentukan bahwa build dipicu hanya ketika pesan komit kepala dari peristiwa push cocok dengan ekspresi reguler`\[CodeBuild \]`.

```
  "filterGroups": [
    [
      {
        "type": "EVENT",
        "pattern": "PUSH"
      },
      {
        "type": "COMMIT_MESSAGE",
        "pattern": "\[CodeBuild\]"
```

```
      }
    ]
  ]
```

Filter acara webhook Bitbucket (CloudFormation)

Untuk menggunakan CloudFormation template untuk memfilter peristiwa webhook, gunakan `FilterGroups` properti AWS CodeBuild proyek. YAML-formatted Bagian berikut dari CloudFormation template membuat dua grup filter. Bersama-sama, mereka memicu pembuatan ketika salah satu atau keduanya mengevaluasi ke BETUL:

- Grup filter pertama menentukan permintaan tarik dibuat atau diperbarui pada cabang dengan nama referensi Git yang cocok dengan ekspresi reguler `^refs/heads/main$` oleh pengguna Bitbucket yang tidak memiliki ID `12345` akun.
- Grup filter kedua menentukan permintaan push dibuat pada cabang dengan nama referensi Git yang cocok dengan ekspresi reguler`^refs/heads/.*`.
- Grup filter ketiga menentukan permintaan push dengan pesan komit kepala yang cocok dengan ekspresi reguler`\[CodeBuild\]`.

```
CodeBuildProject:
  Type: AWS::CodeBuild::Project
  Properties:
    Name: MyProject
    ServiceRole: service-role
    Artifacts:
      Type: NO_ARTIFACTS
    Environment:
      Type: LINUX_CONTAINER
      ComputeType: BUILD_GENERAL1_SMALL
      Image: aws/codebuild/standard:5.0
    Source:
      Type: BITBUCKET
      Location: source-location
    Triggers:
      Webhook: true
      FilterGroups:
        - - Type: EVENT
            Pattern: PULL_REQUEST_CREATED,PULL_REQUEST_UPDATED
          - Type: BASE_REF
            Pattern: ^refs/heads/main$
```

```
            ExcludeMatchedPattern: false
          - Type: ACTOR_ACCOUNT_ID
            Pattern: 12345
            ExcludeMatchedPattern: true
        - - Type: EVENT
            Pattern: PUSH
          - Type: HEAD_REF
            Pattern: ^refs/heads/.*
          - Type: FILE_PATH
            Pattern: READ_ME
            ExcludeMatchedPattern: true
        - - Type: EVENT
            Pattern: PUSH
          - Type: COMMIT_MESSAGE
            Pattern: \[CodeBuild\]
          - Type: FILE_PATH
            Pattern: ^src/.+|^test/.+
```

GitHub webhook global dan organisasi

Anda dapat menggunakan webhook CodeBuild GitHub global atau organisasi untuk memulai membangun acara webhook dari repositori apa pun dalam GitHub organisasi atau perusahaan. Webhook global dan organisasi bekerja dengan salah satu jenis acara GitHub webhook yang ada, dan dapat dikonfigurasi dengan menambahkan konfigurasi cakupan saat membuat CodeBuild webhook. Anda juga dapat menggunakan webhook global dan organisasi untuk mengatur pelari Aksi yang di GitHub hosting sendiri CodeBuild di dalamnya untuk menerima `WORKFLOW_JOB_QUEUED` acara dari beberapa repositori dalam satu proyek.

Topik

- Siapkan GitHub webhook global atau organisasi
- Memfilter acara webhook GitHub global atau organisasi (konsol)
- Filter acara webhook GitHub organisasi (CloudFormation)

Siapkan GitHub webhook global atau organisasi

Langkah-langkah tingkat tinggi untuk mengatur GitHub webhook global atau organisasi adalah sebagai berikut. Untuk informasi selengkapnya tentang GitHub webhook global dan organisasi, lihatGitHub webhook global dan organisasi.

1. Setel lokasi sumber proyek Anda ke`CODEBUILD_DEFAULT_WEBHOOK_SOURCE_LOCATION`.
2. Dalam konfigurasi lingkup webhook, atur cakupan ke salah satu `GITHUB_ORGANIZATION` atau `GITHUB_GLOBAL` tergantung pada apakah itu harus menjadi organisasi atau web hook global. Untuk informasi selengkapnya, lihat Jenis webhook.
3. Tentukan nama sebagai bagian dari konfigurasi lingkup webhook. Untuk webhook organisasi, ini adalah nama organisasi, dan untuk webhook global ini adalah nama perusahaan.

   > **Note**
   >
   > Jika tipe sumber proyek adalah`GITHUB_ENTERPRISE`, Anda juga perlu menentukan domain sebagai bagian dari konfigurasi lingkup webhook.

4. (Opsional) Jika Anda hanya ingin menerima peristiwa webhook untuk repositori tertentu dalam organisasi atau perusahaan Anda, Anda dapat menentukan `REPOSITORY_NAME` sebagai filter saat membuat webhook.
5. Jika Anda membuat webhook organisasi, pastikan itu CodeBuild memiliki izin untuk membuat webhook tingkat organisasi di dalamnya. GitHub Anda dapat membuat token akses GitHub pribadi dengan izin webhook organisasi, atau menggunakan CodeBuild OAuth. Untuk informasi selengkapnya, lihat GitHub dan token akses Server GitHub Perusahaan.

   Perhatikan bahwa webhook organisasi bekerja dengan salah satu jenis acara GitHub webhook yang ada.
6. Jika Anda membuat webhook global, webhook perlu dibuat secara manual. Untuk informasi selengkapnya tentang cara membuat webhook secara manual di dalamnya GitHub, lihatGitHub webhook manual.

   Perhatikan bahwa webhook global hanya mendukung jenis `WORKFLOW_JOB_QUEUED` acara. Untuk informasi selengkapnya, lihat Tutorial: Konfigurasikan pelari CodeBuild-hosted GitHub Tindakan.

Memfilter acara webhook GitHub global atau organisasi (konsol)

Saat membuat GitHub proyek melalui konsol, pilih opsi berikut untuk membuat webhook GitHub global atau organisasi dalam proyek. Untuk informasi selengkapnya tentang GitHub webhook global dan organisasi, lihatGitHub webhook global dan organisasi.

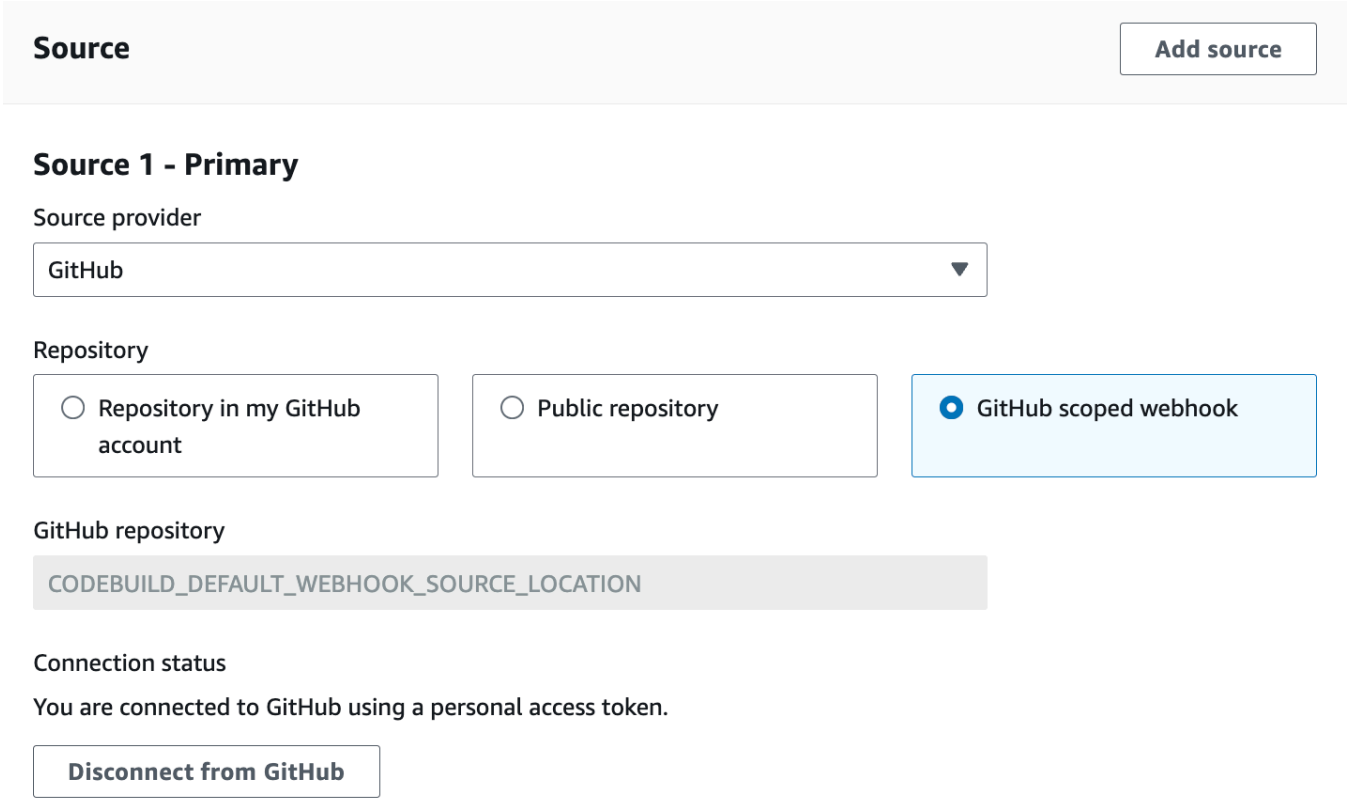
1. Buka AWS CodeBuild konsol di https://console.aws.amazon.com/codesuite/codebuild/home.

2. Buat proyek pembangunan. Untuk informasi selengkapnya, lihat Buat proyek build (konsol) dan Jalankan build (konsol).

   - Dalam Sumber:
     - Untuk Penyedia Sumber, pilih GitHub atau GitHub Perusahaan.
     - Untuk Repositori, pilih web GitHub hook bercakupan.

     GitHub Repositori akan secara otomatis diatur ke`CODEBUILD_DEFAULT_WEBHOOK_SOURCE_LOCATION`, yang merupakan lokasi sumber yang diperlukan untuk webhook global dan organisasi.

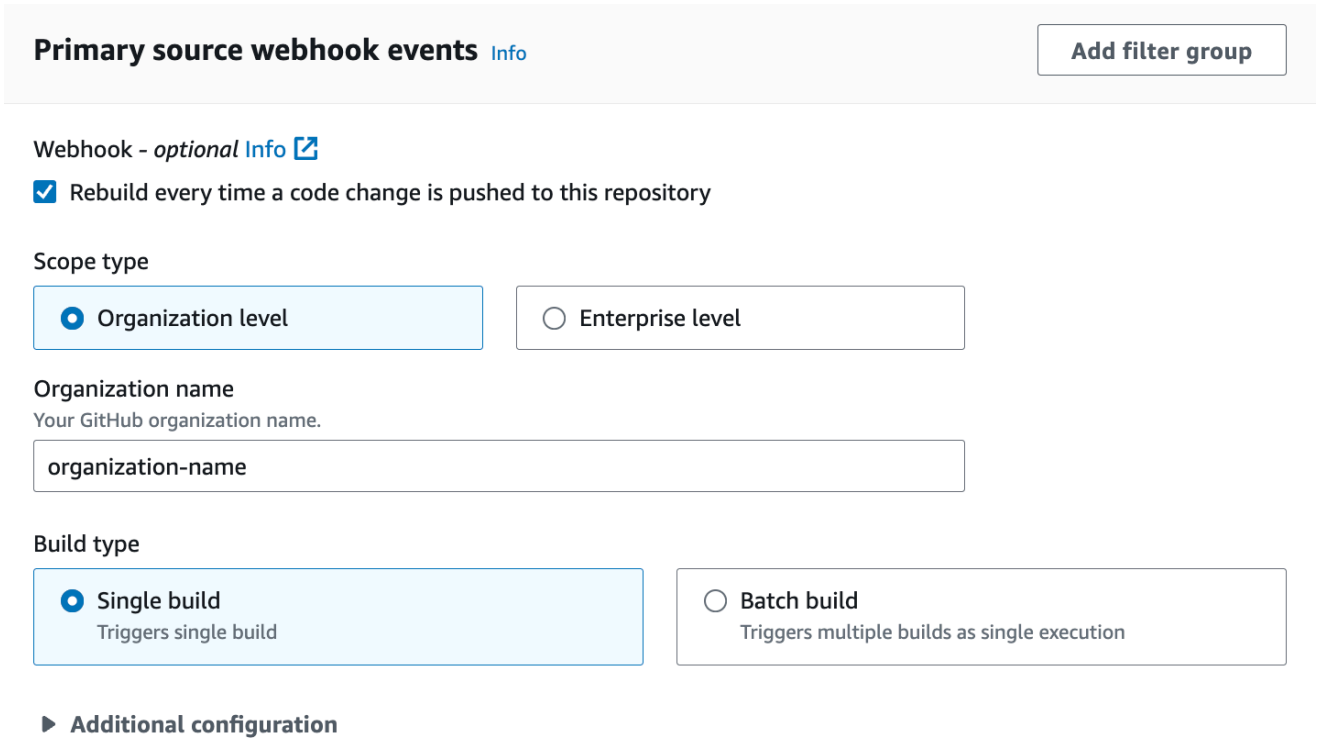
     > **Note**
     >
     > Jika Anda menggunakan webhook organisasi, pastikan itu CodeBuild memiliki izin untuk membuat webhook tingkat organisasi di dalamnya. GitHub Jika Anda menggunakan koneksi OAuth yang ada, Anda mungkin perlu membuat ulang koneksi untuk memberikan izin CodeBuild ini. Atau, Anda dapat membuat webhook secara manual menggunakan fitur webhook CodeBuild manual. Perhatikan bahwa jika Anda memiliki token GitHub OAuth yang ada dan ingin menambahkan izin organisasi tambahan, Anda dapat mencabut izin token OAuth dan menyambungkan kembali token melalui konsol. CodeBuild



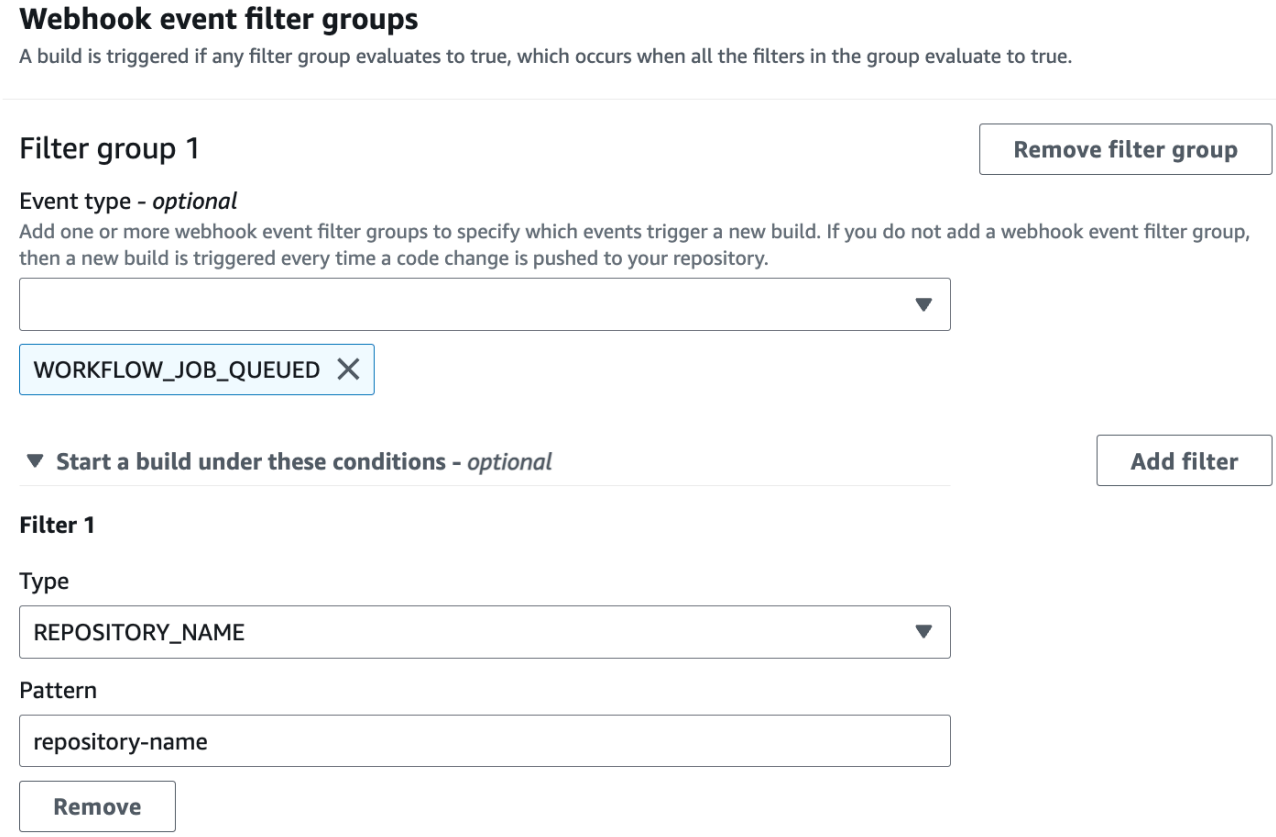
- Dalam acara webhook sumber utama:
  - Untuk Jenis Lingkup, pilih Tingkat organisasi jika Anda membuat webhook organisasi atau tingkat perusahaan jika Anda membuat webhook global.
  - Untuk Nama, masukkan nama perusahaan atau organisasi, tergantung pada apakah webhook adalah webhook global atau organisasi.

    Jika tipe sumber proyek adalah`GITHUB_ENTERPRISE`, Anda juga perlu menentukan domain sebagai bagian dari konfigurasi organisasi webhook. Misalnya, jika URL organisasi Anda adalah**`https://domain.com/orgs/org-name`**, maka domainnya adalah**`https://domain.com`**.

    > **Note**
    >
    > Nama ini tidak dapat diubah setelah webhook dibuat. Untuk mengubah nama, Anda dapat menghapus dan membuat ulang webhook. Jika Anda ingin menghapus webhook sepenuhnya, Anda juga dapat memperbarui lokasi sumber proyek ke GitHub repositori.



- (Opsional) Di grup filter peristiwa Webhook, Anda dapat menentukan peristiwa mana yang ingin Anda picu build baru. Anda juga dapat menentukan `REPOSITORY_NAME` sebagai filter untuk hanya memicu build pada peristiwa webhook dari repositori tertentu.



Anda juga dapat mengatur jenis acara `WORKFLOW_JOB_QUEUED` untuk mengatur pelari Tindakan yang dihosting GitHub sendiri. Untuk informasi selengkapnya, lihat Tutorial: Konfigurasikan pelari CodeBuild-hosted GitHub Tindakan.

3. Lanjutkan dengan nilai default dan kemudian pilih Buat proyek build.

Filter acara webhook GitHub organisasi (CloudFormation)

Untuk menggunakan CloudFormation template untuk memfilter acara webhook organisasi, gunakan `ScopeConfiguration` properti AWS CodeBuild proyek. Untuk informasi selengkapnya tentang GitHub webhook global dan organisasi, lihatGitHub webhook global dan organisasi.

> **Note**
>
> Webhook global dan web GitHub hook perusahaan tidak didukung oleh. CloudFormation

YAML-formatted Bagian berikut dari CloudFormation template membuat empat grup filter. Bersama-sama, mereka memicu pembangunan ketika satu atau semua mengevaluasi dengan benar:

- Grup filter pertama menentukan permintaan tarik dibuat atau diperbarui pada cabang dengan nama referensi Git yang cocok dengan ekspresi reguler `^refs/heads/main$` oleh GitHub pengguna yang tidak memiliki ID akun`12345`.
- Grup filter kedua menentukan permintaan push dibuat pada file dengan nama yang cocok dengan ekspresi reguler `READ_ME` di cabang dengan nama referensi Git yang cocok dengan ekspresi reguler `^refs/heads/.*`.
- Grup filter ketiga menentukan permintaan push dengan pesan komit kepala yang cocok dengan ekspresi reguler`\[CodeBuild\]`.
- Grup filter keempat menentukan permintaan pekerjaan alur kerja Ac GitHub tions dengan nama alur kerja yang cocok dengan ekspresi reguler`\[CI-CodeBuild\]`.

```
CodeBuildProject:
  Type: AWS::CodeBuild::Project
  Properties:
    Name: MyProject
    ServiceRole: service-role
    Artifacts:
      Type: NO_ARTIFACTS
    Environment:
      Type: LINUX_CONTAINER
      ComputeType: BUILD_GENERAL1_SMALL
      Image: aws/codebuild/standard:5.0
    Source:
      Type: GITHUB
      Location: source-location
    Triggers:
      Webhook: true
      ScopeConfiguration:
        Name: organization-name
        Scope: GITHUB_ORGANIZATION
      FilterGroups:
        - - Type: EVENT
            Pattern: PULL_REQUEST_CREATED,PULL_REQUEST_UPDATED
          - Type: BASE_REF
            Pattern: ^refs/heads/main$
            ExcludeMatchedPattern: false
          - Type: ACTOR_ACCOUNT_ID
            Pattern: 12345
            ExcludeMatchedPattern: true
        - - Type: EVENT
```

```
            Pattern: PUSH
          - Type: HEAD_REF
            Pattern: ^refs/heads/.*
          - Type: FILE_PATH
            Pattern: READ_ME
            ExcludeMatchedPattern: true
        - - Type: EVENT
            Pattern: PUSH
          - Type: COMMIT_MESSAGE
            Pattern: \[CodeBuild\]
          - Type: FILE_PATH
            Pattern: ^src/.+|^test/.+
        - - Type: EVENT
            Pattern: WORKFLOW_JOB_QUEUED
          - Type: WORKFLOW_NAME
            Pattern: \[CI-CodeBuild\]
```

GitHub webhook manual

Anda dapat mengonfigurasi GitHub webhook manual untuk CodeBuild mencegah secara otomatis mencoba membuat webhook di dalamnya. GitHub CodeBuild mengembalikan URL payload sebagai bagian dari panggilan untuk membuat webhook, yang dapat digunakan untuk membuat webhook secara manual di dalamnya. GitHub Bahkan jika tidak CodeBuild diizinkan untuk membuat webhook di GitHub akun Anda, Anda masih dapat membuat webhook secara manual untuk proyek build Anda.

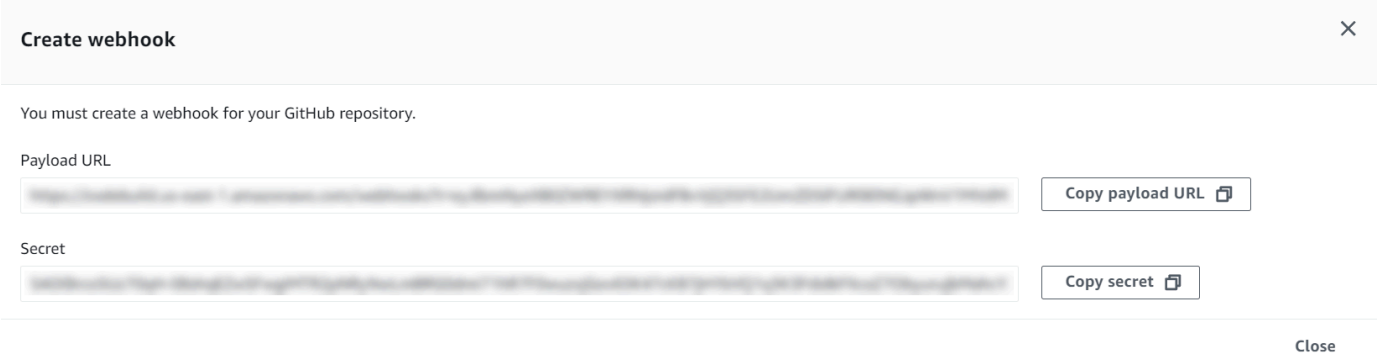
Gunakan prosedur berikut untuk membuat webhook GitHub manual.

Untuk membuat web GitHub hook manual

1. Buka AWS CodeBuild konsol di https://console.aws.amazon.com/codesuite/codebuild/home.
2. Buat proyek pembangunan. Untuk informasi selengkapnya, lihat Buat proyek build (konsol) dan Jalankan build (konsol).
   - Dalam Sumber:
     - Untuk Penyedia Sumber, pilih GitHub.
     - Untuk Repositori, pilih Repositori di GitHub akun saya.
     - Untuk URL Repositori, masukkan**`https://github.com/`*`user-name`*`/`*`repository-name`***.
   - Dalam acara webhook sumber utama:

- Untuk Webhook - opsional, pilih Bang un kembali setiap kali perubahan kode didorong ke repositori ini.
- Pilih Konfigur asi tambahan dan untuk Pembuatan manual - opsional, pilih Buat webhook secara manual untuk repositori ini di GitHub konsol. .

3. Lanjutkan dengan nilai default dan kemudian pilih Buat proyek build. Catat URL Payload dan nilai Rahasia karena Anda akan menggunakannya nanti.



4. Buka GitHub konsol di `https://github.com/`*`user-name`*`/`*`repository-name`*`/settings/hooks` dan pilih Tambahkan webhook.
   - Untuk URL Payload, masukkan nilai URL Payload yang Anda catat sebelumnya.
   - Untuk Jenis konten, pilih application/json.
   - Untuk Rah asia, masukkan nilai Rahasia yang Anda catat sebelumnya.
   - Konfigurasikan peristiwa individual yang akan mengirim muatan webhook ke. CodeBuild Untuk acara mana Anda ingin memicu webhook ini? , pilih Izinkan saya memilih acara individual, lalu pilih dari acara berikut: Dorong, P ull request, dan Releas es. Jika Anda ingin memulai build untuk `WORKFLOW_JOB_QUEUED` acara, pilih pekerjaan alur kerja. Untuk mempelajari lebih lanjut tentang GitHub Actions runner, lihatTutorial: Konfigurasikan pelari CodeBuild-hosted GitHub Tindakan. Untuk mempelajari lebih lanjut tentang jenis acara yang didukung oleh CodeBuild, lihatGitHub acara webhook.
5. Pilih Tambahkan webhook.

GitHub acara webhook

Anda dapat menggunakan grup filter webhook untuk menentukan peristiwa GitHub webhook mana yang memicu build. Misalnya, Anda dapat menentukan bahwa build hanya dipicu untuk perubahan pada cabang tertentu.

Anda dapat membuat satu atau beberapa grup filter webhook untuk menentukan peristiwa webhook mana yang memicu build. Build dipicu jika ada grup filter yang dievaluasi menjadi true, yang terjadi ketika semua filter dalam grup mengevaluasi ke true. Ketika Anda membuat grup filter, tentukan:

Sebuah acara

Untuk GitHub, Anda dapat memilih satu atau beberapa acara berikut:`PUSH`,,`PULL_REQUEST_CREATED`,`PULL_REQUEST_UPDATED`, `PULL_REQUEST_REOPENED``PULL_REQUEST_MERGED`,`PULL_REQUEST_CLOSED`,`RELEASED`,`PRERELEAS` dan`WORKFLOW_JOB_QUEUED`. Jenis acara webhook ada di `X-GitHub-Event` header di payload webhook. Di `X-GitHub-Event` header, Anda mungkin melihat `pull_request` atau`push`. Untuk acara pull request, jenisnya ada di `action` bidang payload peristiwa webhook. Tabel berikut menunjukkan bagaimana nilai `X-GitHub-Event` header dan nilai `action` bidang payload permintaan tarik webhook dipetakan ke jenis peristiwa yang tersedia.

`X-GitHub-Event` Nilai header	Nilai muatan acara Webhook **`action`**	Tipe peristiwa
`pull_request`	`opened`	`PULL_REQUEST_CREATED`
`pull_request`	`reopened`	`PULL_REQUEST_REOPE NED`
`pull_request`	`synchronize`	`PULL_REQUEST_UPDATED`
`pull_request`	`closed`Dan lap `merged` angannya adalah `true`	`PULL_REQUEST_MERGED`
`pull_request`	`closed`Dan lap `merged` angannya adalah `false`	`PULL_REQUEST_CLOSED`
`push`	n/a	`PUSH`
`release`	dirilis	`RELEASED`
`release`	pra-rilis	`PRERELEASED`
`workflow_job`	mengantri	`WORKFLOW_JOB_QUEUED`

> **Note**
>
> Jenis `PULL_REQUEST_REOPENED` acara dapat digunakan dengan GitHub dan Server GitHub Perusahaan saja. Jenis `PRERELEASED` acara `RELEASED` dan GitHub hanya dapat digunakan dengan. Untuk informasi selengkapnya tentang `WORKFLOW_JOB_QUEUED`, lihat Tutorial: Konfigurasikan pelari CodeBuild-hosted GitHub Tindakan.

Satu atau lebih filter opsional

Gunakan ekspresi reguler untuk menentukan filter. Agar suatu peristiwa memicu build, setiap filter dalam grup yang terkait dengannya harus dievaluasi menjadi true.

CodeBuild mengevaluasi pola filter menggunakan sintaks ekspresi RE2 reguler, yang tidak mendukung lookahead, lookbehind, backreference, atau grup atom. Untuk informasi selengkapnya, lihat halaman RE2 Sintaks di GitHub situs web.

`ACTOR_ACCOUNT_ID`(`ACTOR_ID`di konsol)

Peristiwa webhook memicu build saat ID akun Server GitHub Perusahaan cocok dengan pola ekspresi reguler. GitHub Nilai ini ditemukan di `id` properti `sender` objek di payload webhook.

`HEAD_REF`

Peristiwa webhook memicu build ketika referensi kepala cocok dengan pola ekspresi reguler (misalnya, `refs/heads/branch-name` atau`refs/tags/tag-name`). Untuk acara push, nama referensi ditemukan di `ref` properti di payload webhook. Untuk peristiwa pull request, nama cabang ditemukan di `ref` properti `head` objek di payload webhook.

`BASE_REF`

Peristiwa webhook memicu build ketika referensi dasar cocok dengan pola ekspresi reguler (misalnya,`refs/heads/branch-name`). `BASE_REF`Filter dapat digunakan dengan peristiwa permintaan tarik saja. Nama cabang ditemukan di `ref` properti `base` objek di payload webhook.

`FILE_PATH`

Webhook memicu build ketika jalur file yang diubah cocok dengan pola ekspresi reguler. `FILE_PATH`Filter dapat digunakan dengan peristiwa permintaan GitHub push dan pull dan acara push Server GitHub Enterprise. Ini tidak dapat digunakan dengan peristiwa permintaan tarik GitHub Enterprise Server.

> **Note**
>
> `FILE_PATH`Filter dievaluasi terhadap 100 file pertama yang diubah dalam permintaan push atau pull. Untuk memastikan filter diterapkan sebagaimana dimaksud, simpan jumlah file yang diubah dalam batas ini. Untuk peristiwa pull request, sebaiknya gunakan kebijakan build pull request untuk mengontrol permintaan tarik mana yang dapat memulai build.

`COMMIT_MESSAGE`

Webhook memicu build ketika pesan komit kepala cocok dengan pola ekspresi reguler. `COMMIT_MESSAGE`Filter dapat digunakan dengan peristiwa permintaan GitHub push dan pull dan acara push Server GitHub Enterprise. Ini tidak dapat digunakan dengan peristiwa permintaan tarik GitHub Enterprise Server.

`TAG_NAME`

Webhook memicu build ketika nama tag rilis cocok dengan pola ekspresi reguler. `TAG_NAME`Filter dapat digunakan dengan peristiwa permintaan yang GitHub dirilis dan pra-rilis.

`RELEASE_NAME`

Webhook memicu build ketika nama rilis cocok dengan pola ekspresi reguler. `RELEASE_NAME`Filter dapat digunakan dengan peristiwa permintaan yang GitHub dirilis dan pra-rilis.

`REPOSITORY_NAME`

Webhook memicu build ketika nama repositori cocok dengan pola ekspresi reguler. `REPOSITORY_NAME`Filter hanya dapat digunakan dengan web GitHub hook global atau organisasi.

`ORGANIZATION_NAME`

Webhook memicu build ketika nama organisasi cocok dengan pola ekspresi reguler. `ORGANIZATION_NAME`Filter hanya dapat digunakan dengan web GitHub hook global.

`WORKFLOW_NAME`

Webhook memicu build ketika nama alur kerja cocok dengan pola ekspresi reguler. `WORKFLOW_NAME`Filter dapat digunakan dengan GitHub acara permintaan antrian pekerjaan alur kerja Actions.

> **Note**
>
> Anda dapat menemukan payload webhook di pengaturan webhook repositori Anda. GitHub

Topik

- Filter acara GitHub webhook (konsol)
- Filter peristiwa GitHub webhook (SDK)
- Filter acara GitHub webhook (CloudFormation)

Filter acara GitHub webhook (konsol)

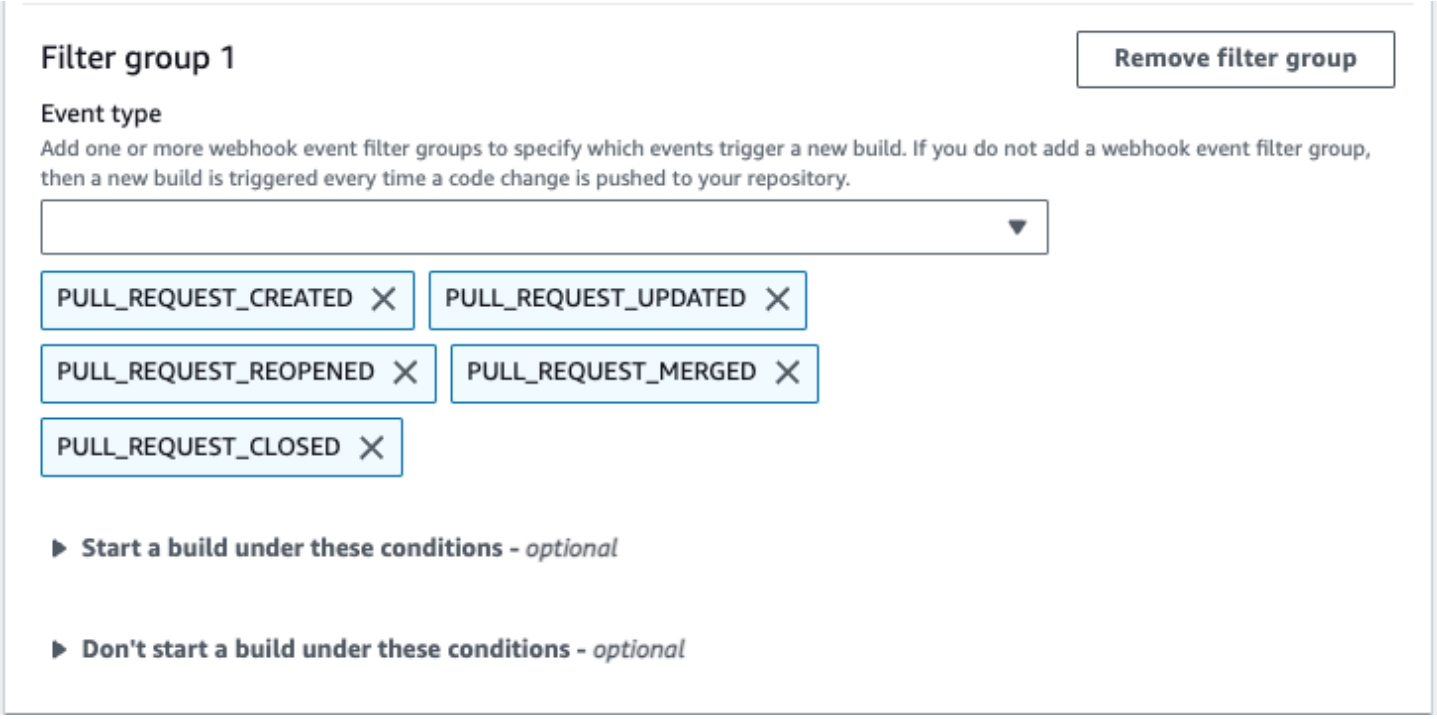
Gunakan petunjuk berikut untuk memfilter peristiwa GitHub webhook menggunakan. Konsol Manajemen AWS Untuk informasi selengkapnya tentang acara GitHub webhook, lihatGitHub acara webhook.

Di acara webhook sumber utama, pilih yang berikut ini. Bagian ini hanya tersedia ketika Anda memilih Repositori di GitHub akun saya untuk repositori sumber.

1. Pilih Bangun kembali setiap kali perubahan kode didorong ke repositori ini saat Anda membuat proyek Anda.
2. Dari Jenis acara, pilih satu atau beberapa acara.
3. Untuk memfilter saat peristiwa memicu build, di bawah Mulai build dalam kondisi ini, tambahkan satu atau beberapa filter opsional.
4. Untuk memfilter saat peristiwa tidak dipicu, di bawah Jangan memulai build dalam kondisi ini, tambahkan satu atau beberapa filter opsional.
5. Pilih Tambahkan grup filter untuk menambahkan grup filter lain, jika diperlukan.

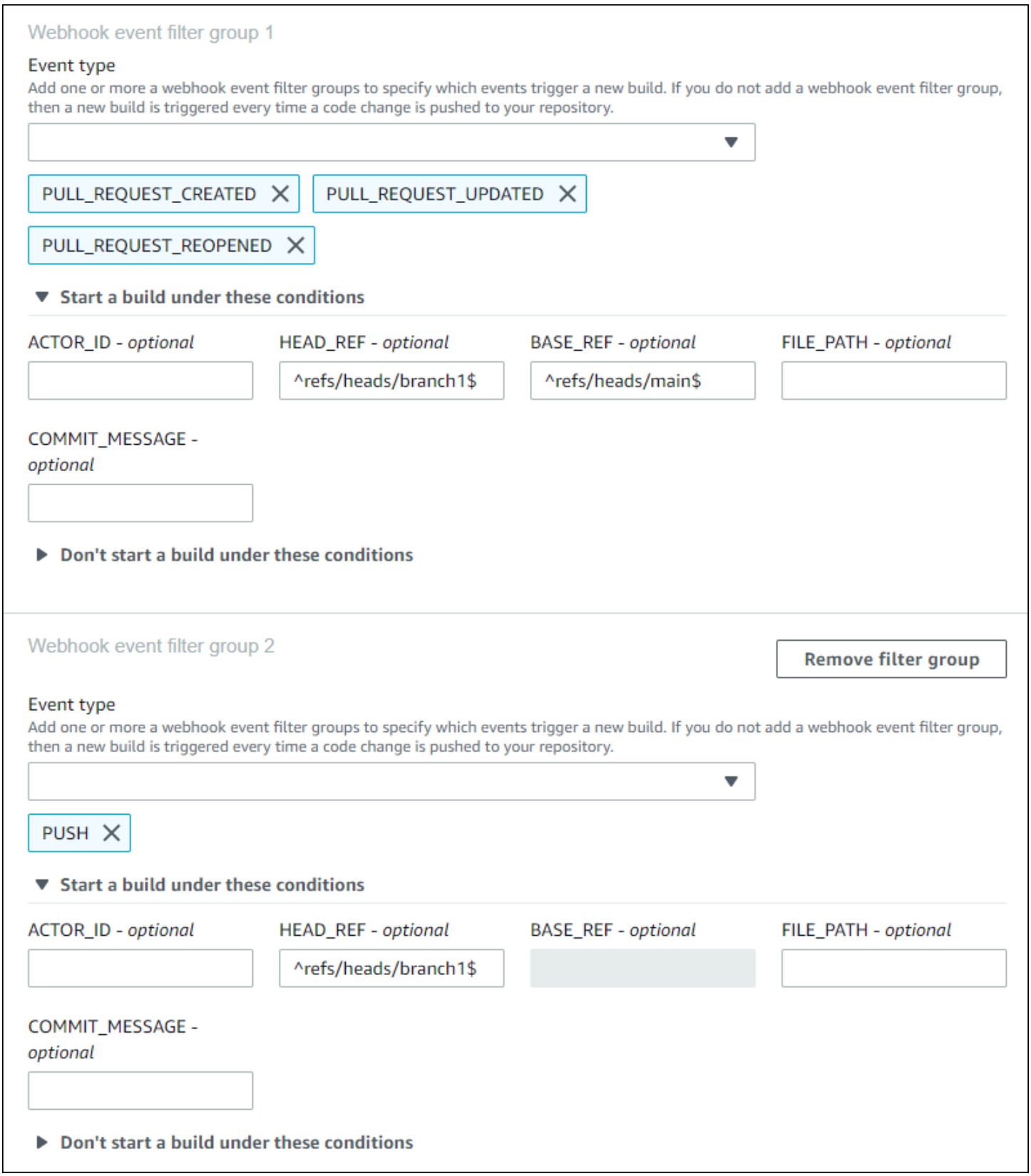
Untuk informasi selengkapnya, lihat Buat proyek build (konsol) dan WebhookFilter di Refer AWS CodeBuild ensi API.

Dalam contoh ini, grup filter webhook memicu build untuk permintaan tarik saja:

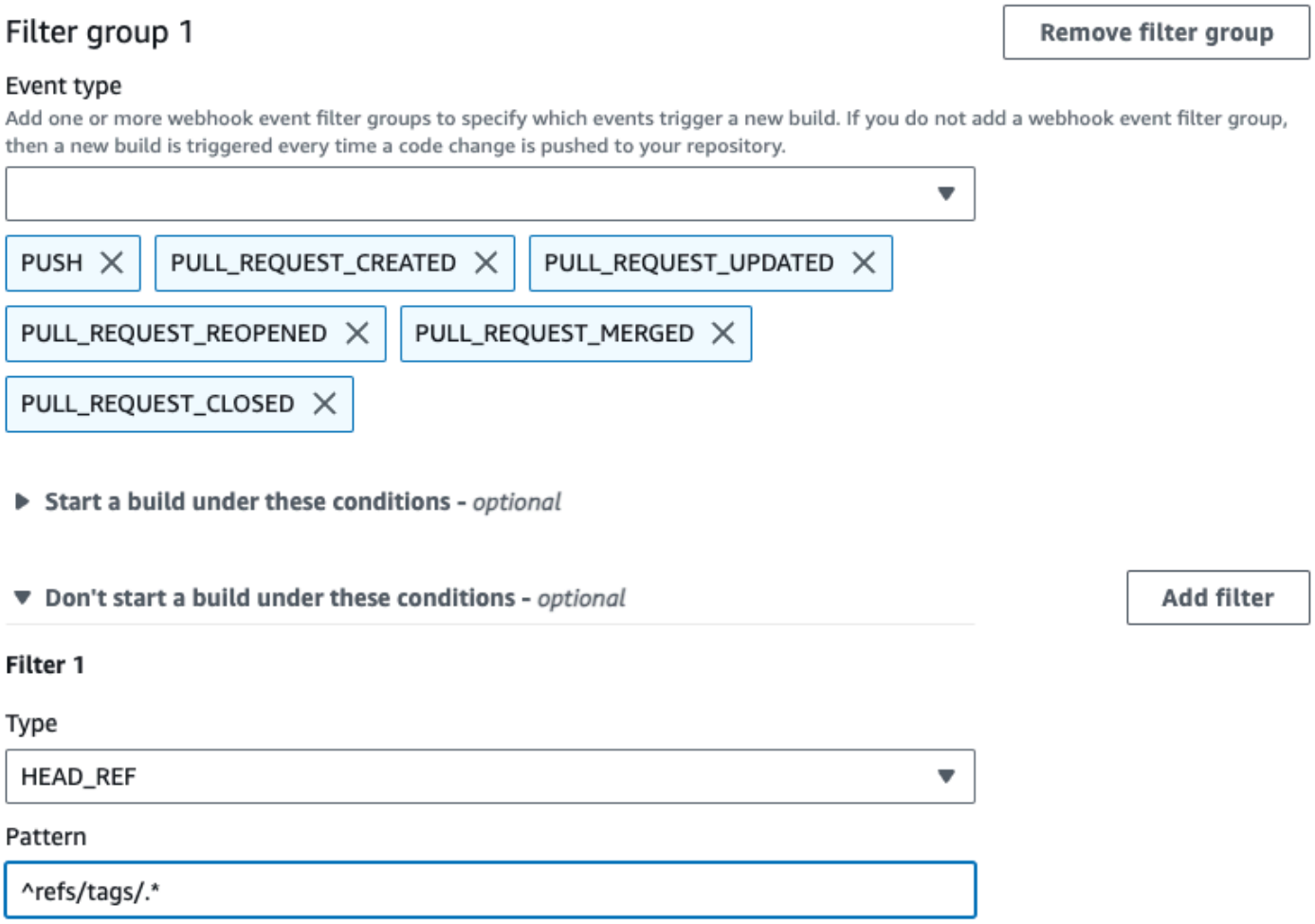


Menggunakan contoh dua grup filter webhook, build dipicu ketika salah satu atau keduanya mengevaluasi menjadi true:

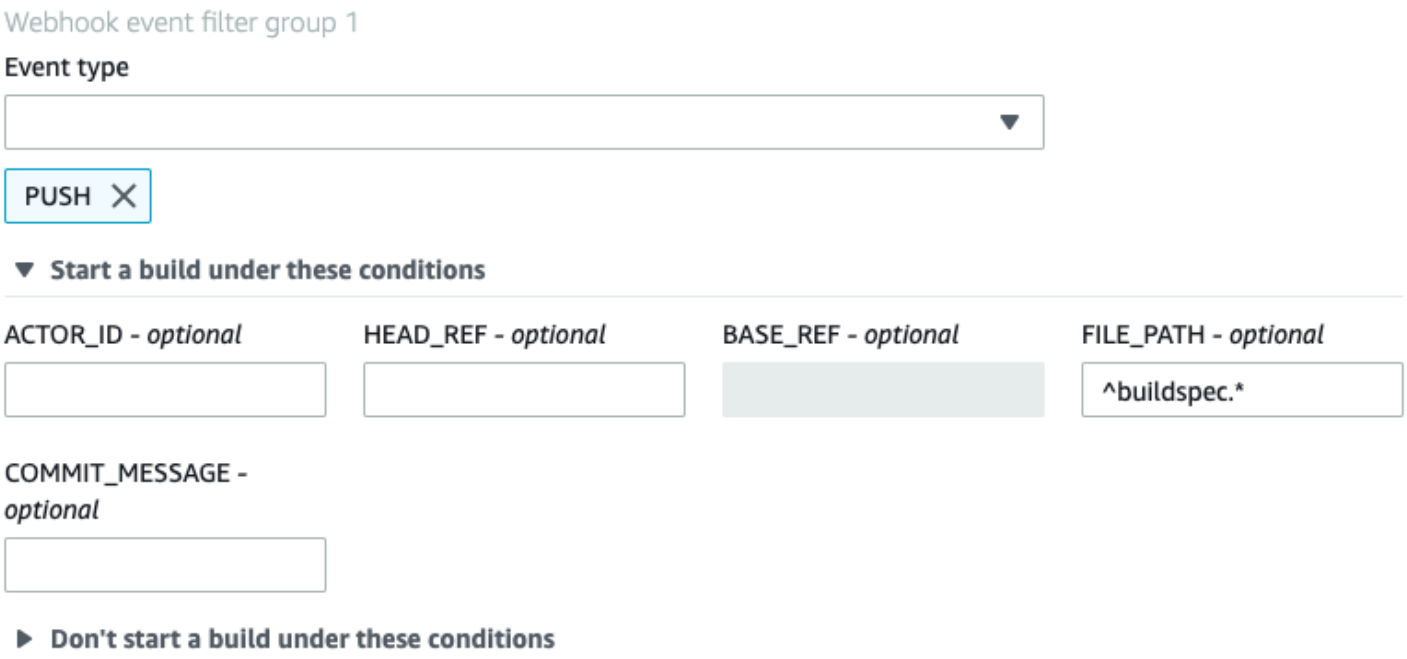
- Grup filter pertama menentukan permintaan tarik yang dibuat, diperbarui, atau dibuka kembali pada cabang dengan nama referensi Git yang cocok dengan ekspresi reguler `^refs/heads/main$` dan referensi kepala yang cocok`^refs/heads/branch1$`.
- Grup filter kedua menentukan permintaan push pada cabang dengan nama referensi Git yang cocok dengan ekspresi reguler`^refs/heads/branch1$`.



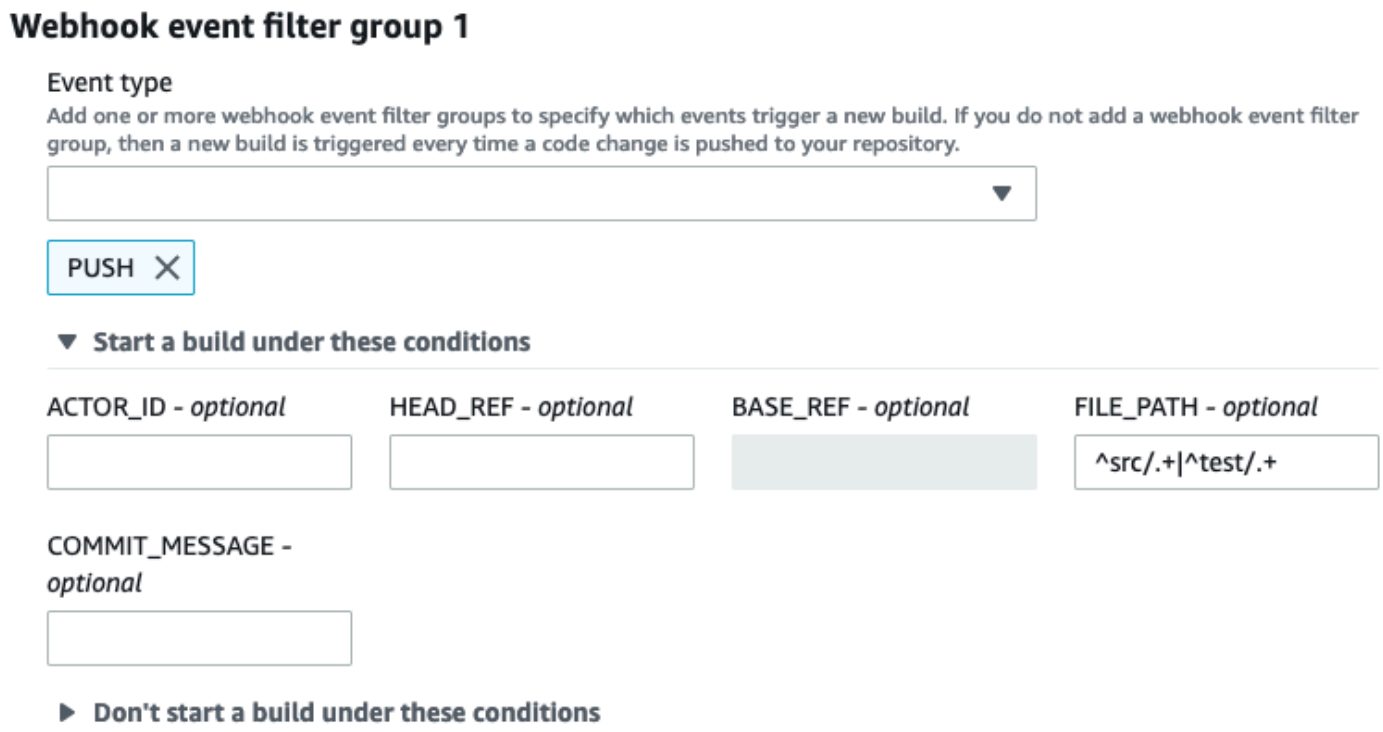
Dalam contoh ini, grup filter webhook memicu build untuk semua permintaan kecuali peristiwa tag.



Dalam contoh ini, grup filter webhook memicu build hanya ketika file dengan nama yang cocok dengan ekspresi reguler `^buildspec.*` berubah.

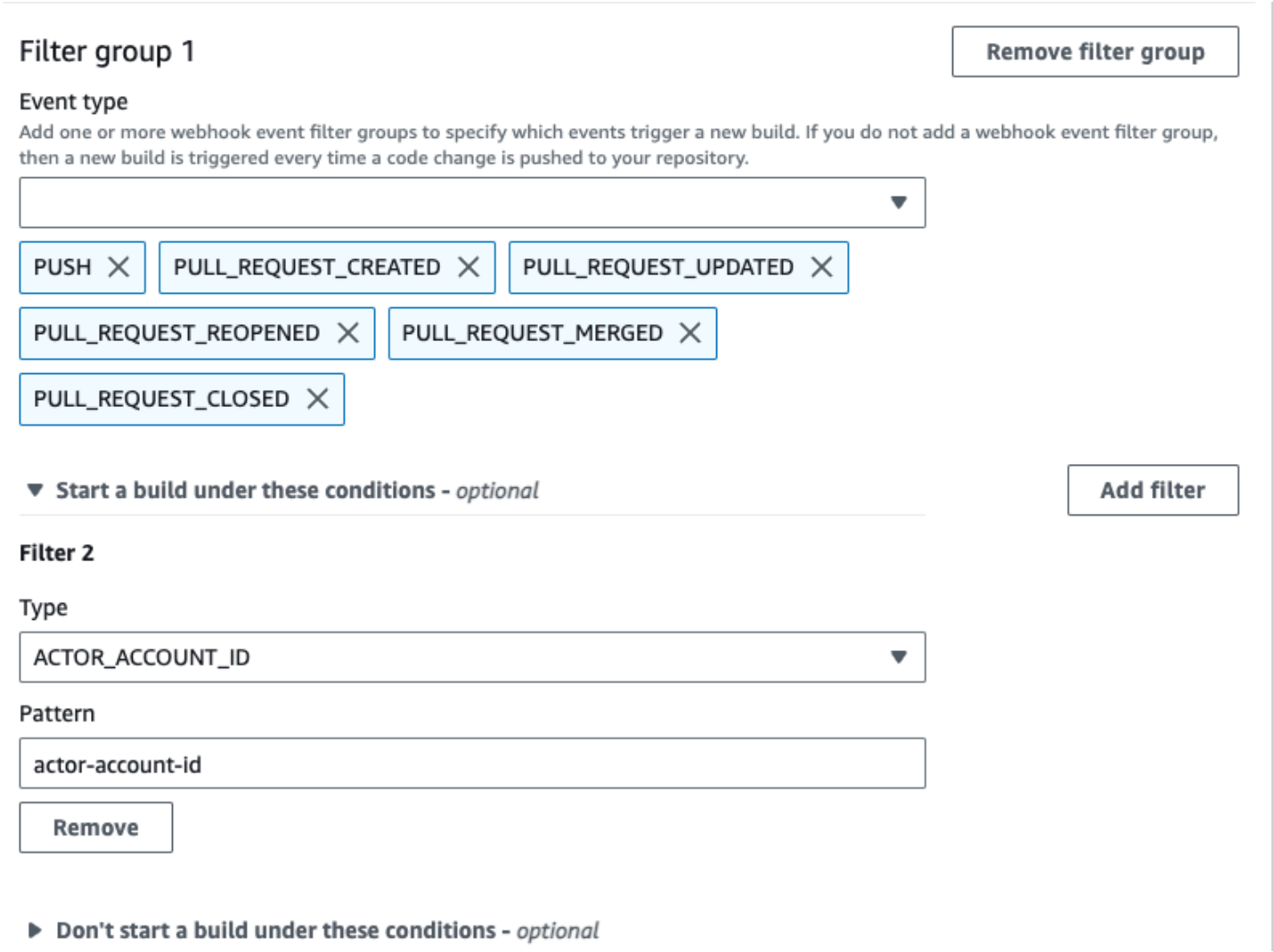


Dalam contoh ini, grup filter webhook memicu build hanya ketika file diubah di dalam `test` folder `src` atau.

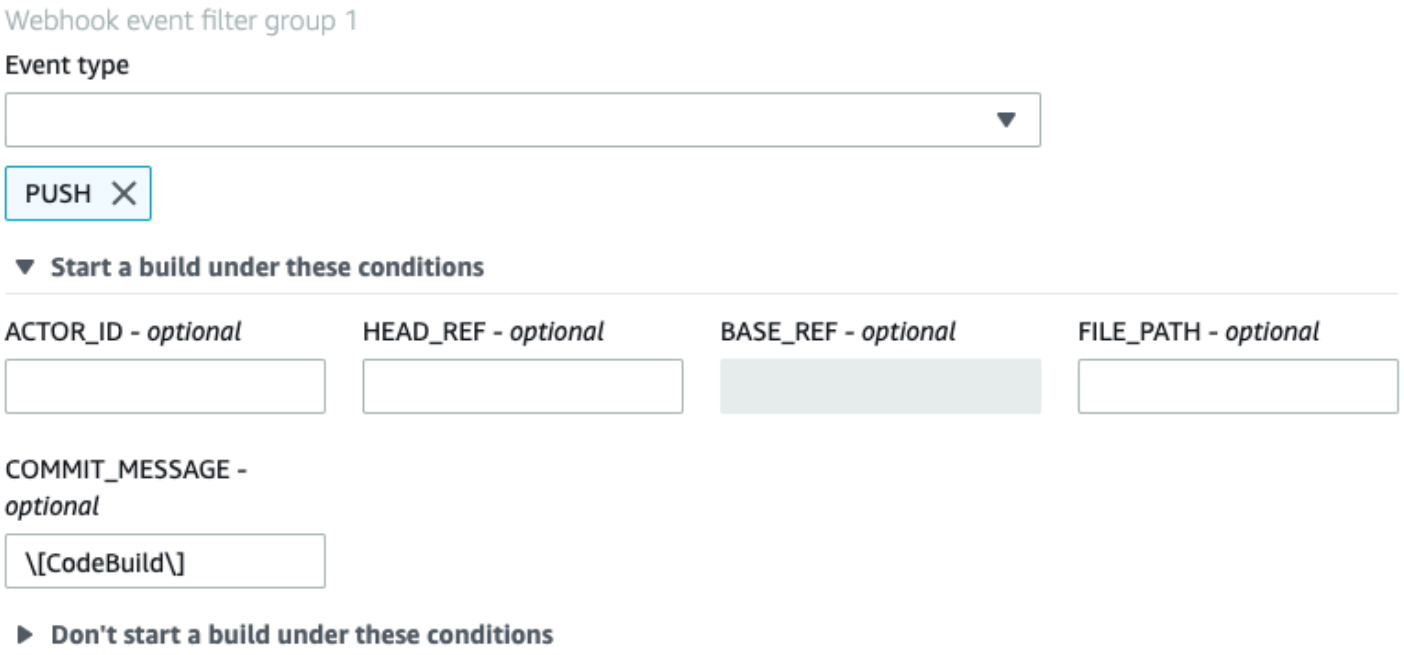


Dalam contoh ini, grup filter webhook memicu build hanya ketika perubahan dilakukan oleh pengguna Server GitHub Perusahaan GitHub atau pengguna tertentu dengan ID akun yang cocok dengan ekspresi `actor-account-id` reguler.

> **Note**
>
> Untuk informasi tentang cara menemukan ID GitHub akun Anda, lihat https://api.github.com/users/*user-name*, di *user-name* mana nama GitHub pengguna Anda.

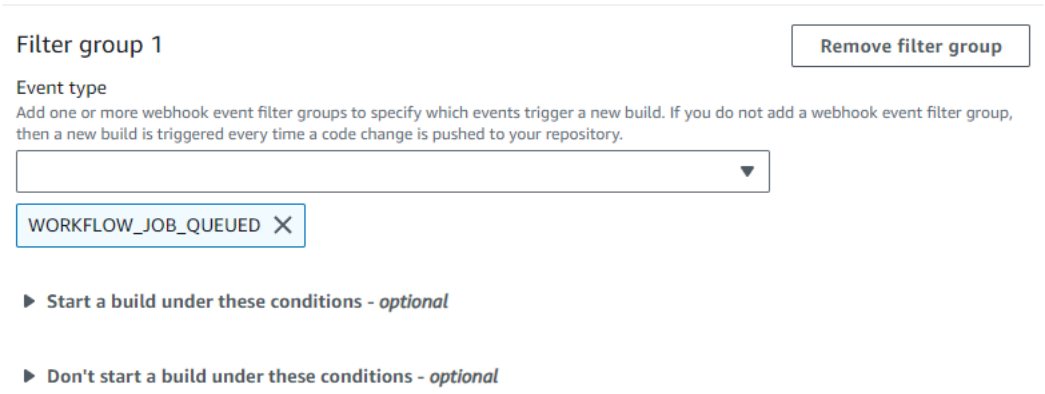


Dalam contoh ini, grup filter webhook memicu build untuk acara push ketika pesan komit kepala cocok dengan ekspresi `\[CodeBuild\]` reguler.

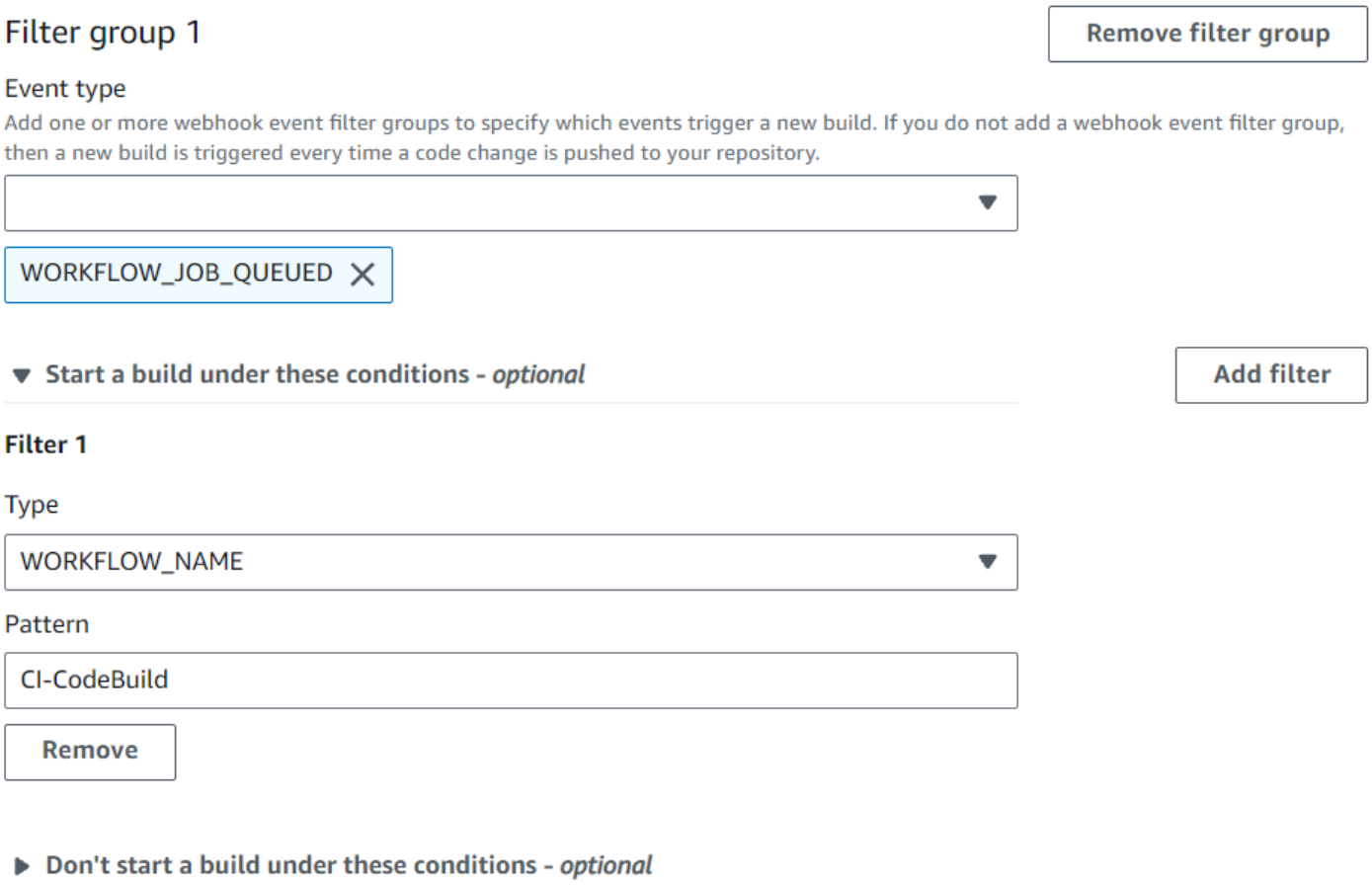


Dalam contoh ini, grup filter webhook memicu peristiwa pekerjaan alur kerja build untuk Ac GitHub tions saja.

> **Note**
>
> CodeBuild hanya akan memproses pekerjaan alur kerja Ac GitHub tions jika webhook memiliki grup filter yang berisi filter acara WORKFLOW_JOB_ QUEUED.



Dalam contoh ini, grup filter webhook memicu build untuk nama alur kerja yang cocok dengan ekspresi `CI-CodeBuild` reguler.



Filter peristiwa GitHub webhook (SDK)

Untuk menggunakan AWS CodeBuild SDK untuk memfilter peristiwa webhook, gunakan `filterGroups` bidang dalam sintaks permintaan metode `UpdateWebhook` API `CreateWebhook` atau. Untuk informasi selengkapnya, lihat WebhookFilter di dalam Referensi API CodeBuild .

Untuk informasi selengkapnya tentang acara GitHub webhook, lihatGitHub acara webhook.

Untuk membuat filter webhook yang memicu build hanya untuk permintaan tarik, masukkan yang berikut ini ke dalam sintaks permintaan:

```
"filterGroups": [
   [
        {
            "type": "EVENT",
            "pattern": "PULL_REQUEST_CREATED, PULL_REQUEST_UPDATED,
 PULL_REQUEST_REOPENED, PULL_REQUEST_MERGED, PULL_REQUEST_CLOSED"
        }
```

```
    ]
]
```

Untuk membuat filter webhook yang memicu build untuk cabang tertentu saja, gunakan `pattern` parameter untuk menentukan ekspresi reguler untuk memfilter nama cabang. Menggunakan contoh dua grup filter, build dipicu ketika salah satu atau keduanya mengevaluasi menjadi true:

- Grup filter pertama menentukan permintaan tarik yang dibuat, diperbarui, atau dibuka kembali pada cabang dengan nama referensi Git yang cocok dengan ekspresi reguler `^refs/heads/main$` dan referensi kepala yang cocok`^refs/heads/myBranch$`.
- Grup filter kedua menentukan permintaan push pada cabang dengan nama referensi Git yang cocok dengan ekspresi reguler`^refs/heads/myBranch$`.

```
"filterGroups": [
    [
        {
            "type": "EVENT",
            "pattern": "PULL_REQUEST_CREATED, PULL_REQUEST_UPDATED,
 PULL_REQUEST_REOPENED"
        },
        {
            "type": "HEAD_REF",
            "pattern": "^refs/heads/myBranch$"
        },
        {
            "type": "BASE_REF",
            "pattern": "^refs/heads/main$"
        }
    ],
    [
        {
            "type": "EVENT",
            "pattern": "PUSH"
        },
        {
            "type": "HEAD_REF",
            "pattern": "^refs/heads/myBranch$"
        }
    ]
]
```

Anda dapat menggunakan `excludeMatchedPattern` parameter untuk menentukan peristiwa mana yang tidak memicu build. Misalnya, dalam contoh ini build dipicu untuk semua permintaan kecuali peristiwa tag.

```
"filterGroups": [
    [
        {
            "type": "EVENT",
            "pattern": "PUSH, PULL_REQUEST_CREATED, PULL_REQUEST_UPDATED,
 PULL_REQUEST_REOPENED, PULL_REQUEST_MERGED, PULL_REQUEST_CLOSED"
        },
        {
            "type": "HEAD_REF",
            "pattern": "^refs/tags/.*",
            "excludeMatchedPattern": true
        }
    ]
]
```

Anda dapat membuat filter yang memicu build hanya ketika file dengan nama yang cocok dengan ekspresi reguler dalam `pattern` argumen berubah. Dalam contoh ini, grup filter menentukan bahwa build dipicu hanya ketika file dengan nama yang cocok dengan ekspresi reguler `^buildspec.*` berubah.

```
"filterGroups": [
    [
        {
            "type": "EVENT",
            "pattern": "PUSH"
        },
        {
            "type": "FILE_PATH",
            "pattern": "^buildspec.*"
        }
    ]
]
```

Dalam contoh ini, grup filter menentukan bahwa build dipicu hanya ketika file diubah dalam `test` folder `src` atau.

```
"filterGroups": [
```

```
    [
        {
            "type": "EVENT",
            "pattern": "PUSH"
        },
        {
            "type": "FILE_PATH",
            "pattern": "^src/.+|^test/.+"
        }
    ]
]
```

Anda dapat membuat filter yang memicu build hanya ketika perubahan dilakukan oleh pengguna Server GitHub Perusahaan GitHub atau pengguna tertentu dengan ID akun`actor-account-id`.

> **Note**
>
> Untuk informasi tentang cara menemukan ID GitHub akun Anda, lihat https://api.github.com/users/*`user-name`*, di *`user-name`* mana nama GitHub pengguna Anda.

```
"filterGroups": [
    [
        {
            "type": "EVENT",
            "pattern": "PUSH, PULL_REQUEST_CREATED, PULL_REQUEST_UPDATED,
 PULL_REQUEST_REOPENED, PULL_REQUEST_MERGED, PULL_REQUEST_CLOSED"
        },
        {
            "type": "ACTOR_ACCOUNT_ID",
            "pattern": "actor-account-id"
        }
    ]
]
```

Anda dapat membuat filter yang memicu build hanya ketika pesan komit kepala cocok dengan ekspresi reguler dalam argumen pola. Dalam contoh ini, grup filter menentukan bahwa build dipicu hanya ketika pesan komit kepala dari peristiwa push cocok dengan ekspresi reguler`\[CodeBuild \]`.

```
"filterGroups": [
```

```
    [
        {
            "type": "EVENT",
            "pattern": "PUSH"
        },
        {
            "type": "COMMIT_MESSAGE",
            "pattern": "\[CodeBuild\]"
        }
    ]
]
```

Untuk membuat filter webhook yang memicu pekerjaan alur kerja build for Ac GitHub tions saja, masukkan yang berikut ini ke dalam sintaks permintaan:

```
"filterGroups": [
   [
        {
            "type": "EVENT",
            "pattern": "WORKFLOW_JOB_QUEUED"
        }
    ]
]
```

Filter acara GitHub webhook (CloudFormation)

Untuk menggunakan CloudFormation template untuk memfilter peristiwa webhook, gunakan `FilterGroups` properti AWS CodeBuild proyek.

Untuk informasi selengkapnya tentang acara GitHub webhook, lihatGitHub acara webhook.

YAML-formatted Bagian berikut dari CloudFormation template membuat dua grup filter. Bersama-sama, mereka memicu pembuatan ketika salah satu atau keduanya mengevaluasi ke BETUL:

- Grup filter pertama menentukan permintaan tarik dibuat atau diperbarui pada cabang dengan nama referensi Git yang cocok dengan ekspresi reguler `^refs/heads/main$` oleh GitHub pengguna yang tidak memiliki ID akun`12345`.
- Grup filter kedua menentukan permintaan push dibuat pada file dengan nama yang cocok dengan ekspresi reguler `READ_ME` di cabang dengan nama referensi Git yang cocok dengan ekspresi reguler `^refs/heads/.*`.

- Grup filter ketiga menentukan permintaan push dengan pesan komit kepala yang cocok dengan ekspresi reguler`\[CodeBuild\]`.
- Grup filter keempat menentukan permintaan pekerjaan alur kerja Ac GitHub tions dengan nama alur kerja yang cocok dengan ekspresi reguler`\[CI-CodeBuild\]`.

```
CodeBuildProject:
  Type: AWS::CodeBuild::Project
  Properties:
    Name: MyProject
    ServiceRole: service-role
    Artifacts:
      Type: NO_ARTIFACTS
    Environment:
      Type: LINUX_CONTAINER
      ComputeType: BUILD_GENERAL1_SMALL
      Image: aws/codebuild/standard:5.0
    Source:
      Type: GITHUB
      Location: source-location
    Triggers:
      Webhook: true
      FilterGroups:
        - - Type: EVENT
            Pattern: PULL_REQUEST_CREATED,PULL_REQUEST_UPDATED
          - Type: BASE_REF
            Pattern: ^refs/heads/main$
            ExcludeMatchedPattern: false
          - Type: ACTOR_ACCOUNT_ID
            Pattern: 12345
            ExcludeMatchedPattern: true
        - - Type: EVENT
            Pattern: PUSH
          - Type: HEAD_REF
            Pattern: ^refs/heads/.*
          - Type: FILE_PATH
            Pattern: READ_ME
            ExcludeMatchedPattern: true
        - - Type: EVENT
            Pattern: PUSH
          - Type: COMMIT_MESSAGE
            Pattern: \[CodeBuild\]
          - Type: FILE_PATH
```

```
        Pattern: ^src/.+|^test/.+
    - - Type: EVENT
        Pattern: WORKFLOW_JOB_QUEUED
      - Type: WORKFLOW_NAME
        Pattern: \[CI-CodeBuild\]
```

GitLab kelompok webhooks

Anda dapat menggunakan webhook CodeBuild GitLab grup untuk memulai build pada peristiwa webhook dari repositori apa pun dalam grup. GitLab Webhook grup bekerja dengan salah satu jenis acara GitLab webhook yang ada, dan dapat dikonfigurasi dengan menambahkan konfigurasi cakupan saat membuat webhook. CodeBuild Anda juga dapat menggunakan webhook grup untuk menyiapkan GitLab pelari yang dihosting sendiri CodeBuild di dalamnya untuk menerima `WORKFLOW_JOB_QUEUED` peristiwa dari beberapa repositori dalam satu proyek.

Topik

- Siapkan GitLab webhook grup
- Filter acara webhook GitLab grup (konsol)
- Filter acara webhook GitLab grup (CloudFormation)

Siapkan GitLab webhook grup

Langkah-langkah tingkat tinggi untuk mengatur GitLab webhook grup adalah sebagai berikut. Untuk informasi selengkapnya tentang GitLab webhook grup, lihat. GitLab kelompok webhooks

1. Tetapkan lokasi sumber proyek Anda ke`CODEBUILD_DEFAULT_WEBHOOK_SOURCE_LOCATION`.
2. Dalam konfigurasi lingkup webhook, atur ruang lingkup ke`GITLAB_GROUP`.
3. Tentukan nama sebagai bagian dari konfigurasi cakupan webhook. Untuk webhook grup, ini adalah nama grup.

   > **Note**
   >
   > Jika jenis sumber proyek adalah`GITLAB_SELF_MANAGED`, Anda juga perlu menentukan domain sebagai bagian dari konfigurasi lingkup webhook.

4. (Opsional) Jika Anda hanya ingin menerima peristiwa webhook untuk repositori tertentu dalam organisasi atau perusahaan Anda, Anda dapat menentukan `REPOSITORY_NAME` sebagai filter saat membuat webhook.

5. Saat membuat webhook grup, pastikan CodeBuild memiliki izin untuk membuat webhook tingkat grup di dalamnya. GitLab Untuk melakukannya, Anda dapat menggunakan CodeBuild OAuth. AWS CodeConnections Untuk informasi selengkapnya, lihat GitLab akses di CodeBuild.

   Perhatikan bahwa webhook grup bekerja dengan salah satu jenis acara GitLab webhook yang ada.

Filter acara webhook GitLab grup (konsol)

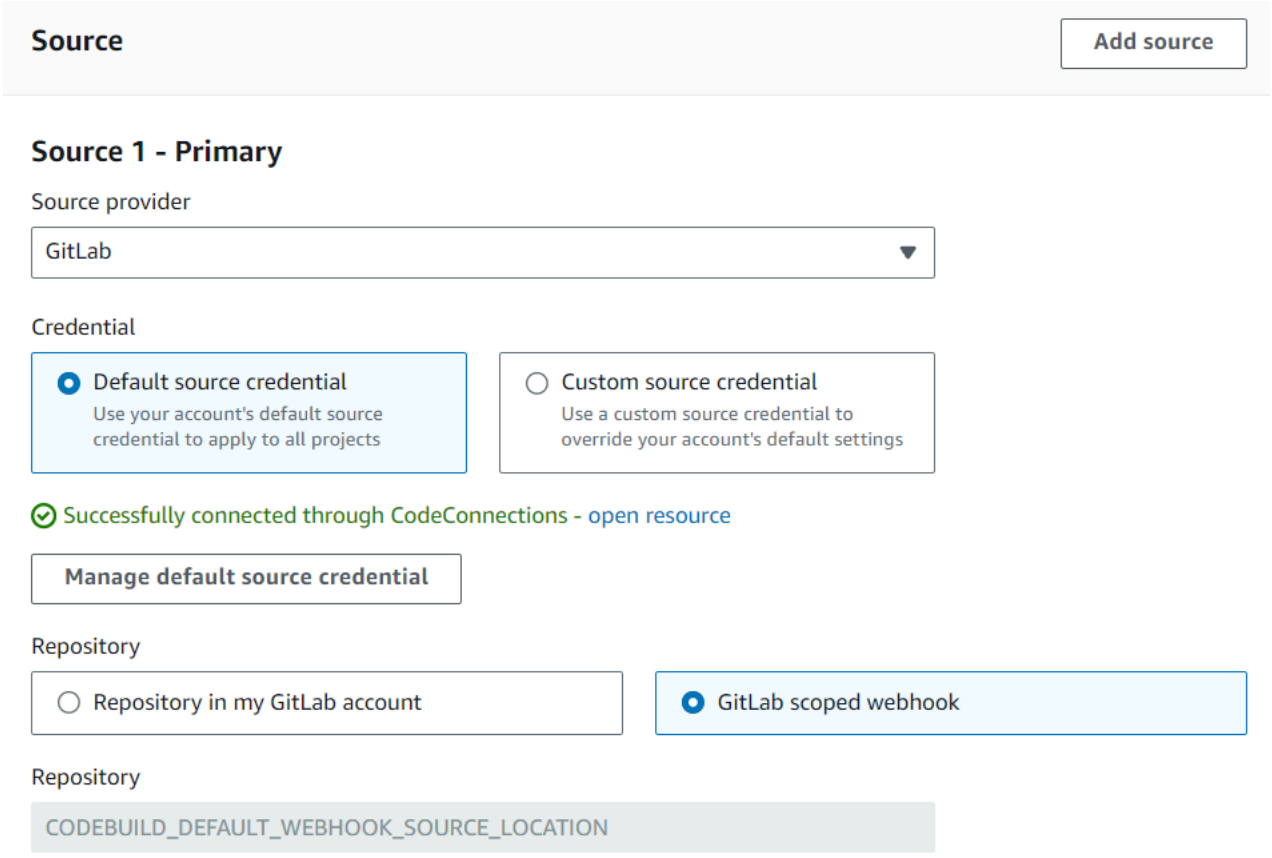
Saat membuat GitLab proyek melalui konsol, pilih opsi berikut untuk membuat webhook GitLab grup dalam proyek. Untuk informasi selengkapnya tentang GitLab webhook grup, lihat. GitLab kelompok webhooks

1. Buka AWS CodeBuild konsol di https://console.aws.amazon.com/codesuite/codebuild/home.
2. Buat proyek build. Untuk informasi selengkapnya, lihat Buat proyek build (konsol) dan Jalankan build (konsol).

   - Dalam Sumber:
     - Untuk penyedia Sumber, pilih GitLabatau Dikelola GitLab Sendiri.
     - Untuk Repositori, pilih GitLab webhook cakupan.

       GitLab Repositori akan secara otomatis diatur ke`CODEBUILD_DEFAULT_WEBHOOK_SOURCE_LOCATION`, yang merupakan lokasi sumber yang diperlukan untuk webhook grup.

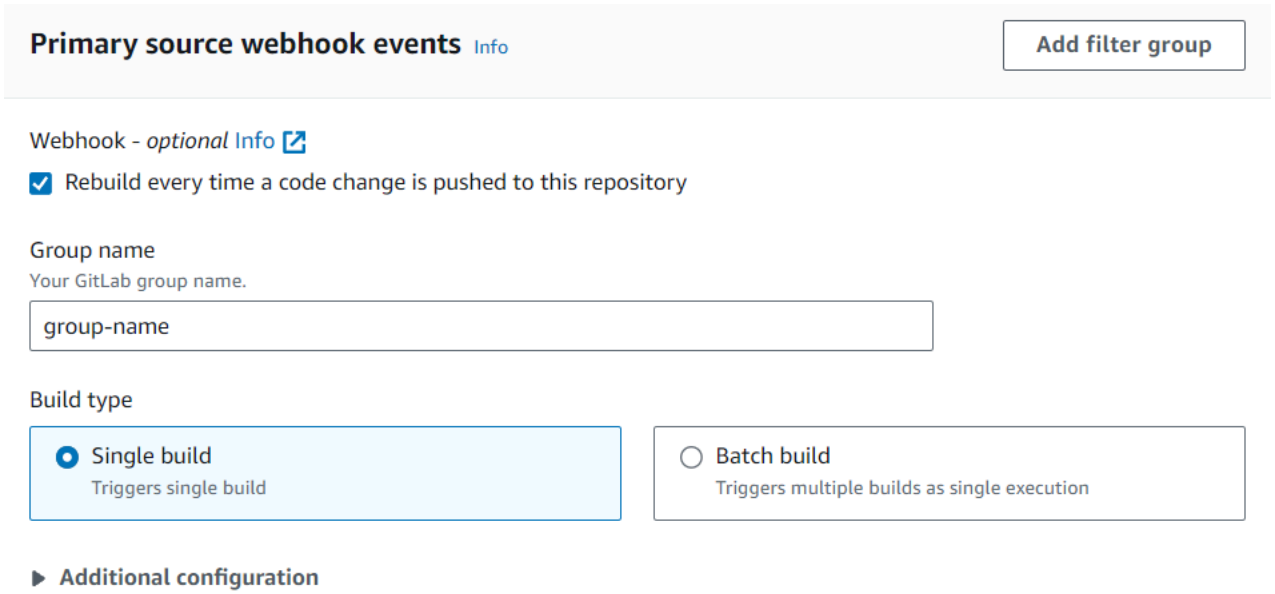
       > **Note**
       >
       > Saat menggunakan webhook grup, pastikan itu CodeBuild memiliki izin untuk membuat webhook tingkat grup di dalamnya. GitLab Jika Anda menggunakan koneksi OAuth yang ada, Anda mungkin perlu membuat ulang koneksi untuk memberikan izin CodeBuild ini.



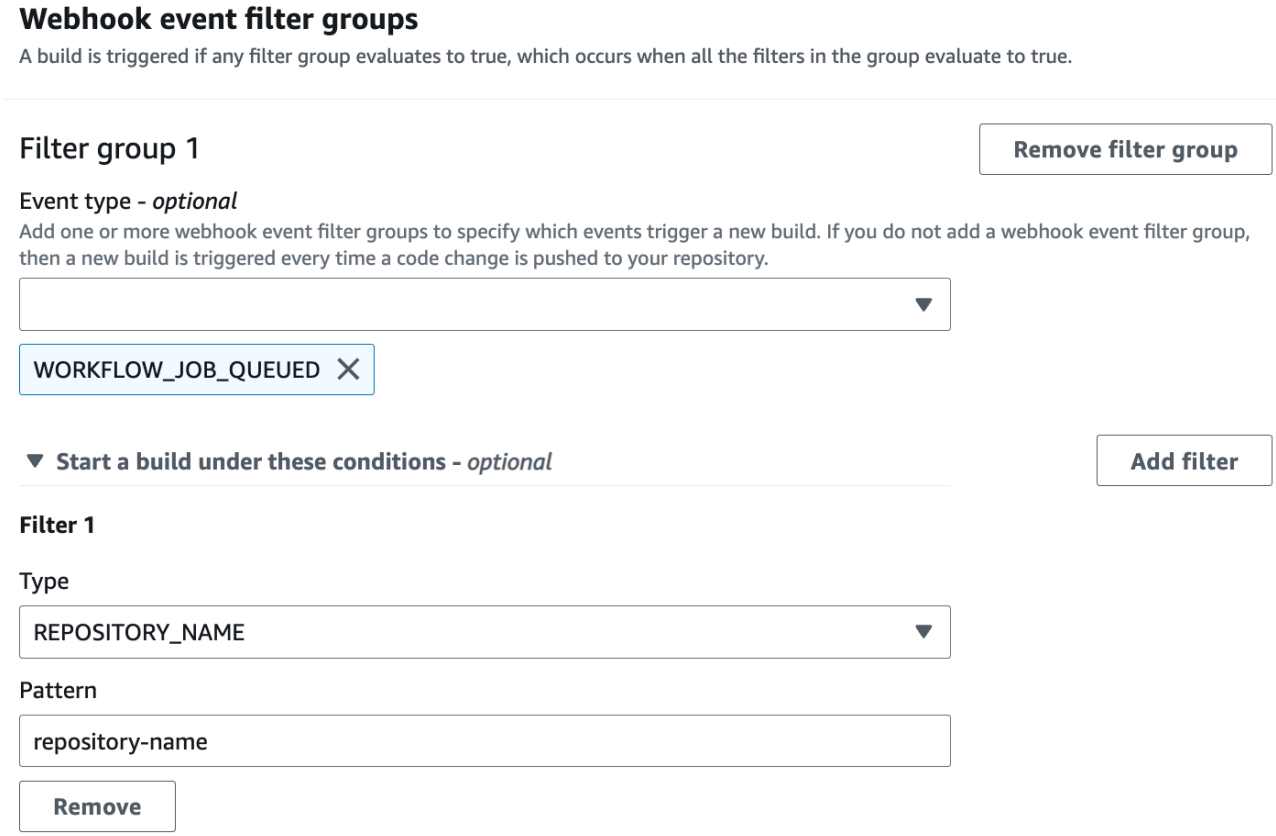
- Dalam acara webhook sumber utama:
  - Untuk nama Grup, masukkan nama grup.

    Jika jenis sumber proyek adalah`GITLAB_SELF_MANAGED`, Anda juga perlu menentukan domain sebagai bagian dari konfigurasi grup webhook. Misalnya, jika URL grup Anda adalah**`https://domain.com/group/group-name`**, maka domain tersebut**`https://domain.com`**.

    > **Note**
    >
    > Nama ini tidak dapat diubah setelah webhook dibuat. Untuk mengubah nama, Anda dapat menghapus dan membuat ulang webhook. Jika Anda ingin menghapus webhook sepenuhnya, Anda juga dapat memperbarui lokasi sumber proyek ke GitLab repositori.



- (Opsional) Di grup filter peristiwa Webhook, Anda dapat menentukan peristiwa mana yang ingin memicu build baru. Anda juga dapat menentukan `REPOSITORY_NAME` sebagai filter untuk hanya memicu build pada peristiwa webhook dari repositori tertentu.



Anda juga dapat mengatur jenis acara `WORKFLOW_JOB_QUEUED` untuk menyiapkan GitLab pelari yang dihosting sendiri. Untuk informasi selengkapnya, lihat Self-managed GitLab pelari di AWS CodeBuild.

3. Lanjutkan dengan nilai default dan kemudian pilih Create build project.

Filter acara webhook GitLab grup (CloudFormation)

Untuk menggunakan CloudFormation template untuk memfilter peristiwa webhook grup, gunakan `ScopeConfiguration` properti AWS CodeBuild proyek. Untuk informasi selengkapnya tentang GitLab webhook grup, lihat. GitLab kelompok webhooks

YAML-formatted Bagian berikut dari CloudFormation template menciptakan empat grup filter. Bersama-sama, mereka memicu build ketika satu atau semua mengevaluasi ke true:

- Grup filter pertama menentukan permintaan tarik dibuat atau diperbarui pada cabang dengan nama referensi Git yang cocok dengan ekspresi reguler `^refs/heads/main$` oleh GitLab pengguna yang tidak memiliki ID `12345` akun.
- Grup filter kedua menentukan permintaan push dibuat pada file dengan nama yang cocok dengan ekspresi reguler `README_ME` di cabang dengan nama referensi Git yang cocok dengan ekspresi reguler `^refs/heads/.*`.
- Grup filter ketiga menentukan permintaan push dengan pesan head commit yang cocok dengan ekspresi `\[CodeBuild\]` reguler.
- Grup filter keempat menentukan permintaan pekerjaan GitLab CI/CD pipeline dengan nama CI/CD pipeline yang cocok dengan ekspresi `\[CI-CodeBuild\]` reguler.

```
CodeBuildProject:
  Type: AWS::CodeBuild::Project
  Properties:
    Name: MyProject
    ServiceRole: service-role
    Artifacts:
      Type: NO_ARTIFACTS
    Environment:
      Type: LINUX_CONTAINER
      ComputeType: BUILD_GENERAL1_SMALL
      Image: aws/codebuild/standard:5.0
    Source:
```

```
      Type: GITLAB
      Location: source-location
    Triggers:
      Webhook: true
      ScopeConfiguration:
        Name: group-name
        Scope: GITLAB_GROUP
      FilterGroups:
        - - Type: EVENT
            Pattern: PULL_REQUEST_CREATED,PULL_REQUEST_UPDATED
          - Type: BASE_REF
            Pattern: ^refs/heads/main$
            ExcludeMatchedPattern: false
          - Type: ACTOR_ACCOUNT_ID
            Pattern: 12345
            ExcludeMatchedPattern: true
        - - Type: EVENT
            Pattern: PUSH
          - Type: HEAD_REF
            Pattern: ^refs/heads/.*
          - Type: FILE_PATH
            Pattern: READ_ME
            ExcludeMatchedPattern: true
        - - Type: EVENT
            Pattern: PUSH
          - Type: COMMIT_MESSAGE
            Pattern: \[CodeBuild\]
          - Type: FILE_PATH
            Pattern: ^src/.+|^test/.+
        - - Type: EVENT
            Pattern: WORKFLOW_JOB_QUEUED
          - Type: WORKFLOW_NAME
            Pattern: \[CI-CodeBuild\]
```

GitLab webhook manual

Anda dapat mengonfigurasi GitLab webhook manual untuk CodeBuild mencegah secara otomatis mencoba membuat webhook di dalamnya. GitLab CodeBuild mengembalikan URL payload sebagai bagian dari panggilan untuk membuat webhook, yang dapat digunakan untuk membuat webhook secara manual di dalamnya. GitLab Bahkan jika tidak CodeBuild diizinkan untuk membuat webhook di GitLab akun Anda, Anda masih dapat membuat webhook secara manual untuk proyek build Anda.

Gunakan prosedur berikut untuk membuat webhook GitLab manual.

Untuk membuat web GitLab hook manual

1. Buka AWS CodeBuild konsol di https://console.aws.amazon.com/codesuite/codebuild/home.
2. Buat proyek pembangunan. Untuk informasi selengkapnya, lihat Buat proyek build (konsol) dan Jalankan build (konsol).
   - Dalam Sumber:
     - Untuk Penyedia Sumber, pilih GitLab.
     - Untuk Repositori, pilih Repositori di GitLab akun saya.
     - Untuk URL Repositori, masukkan**`https://gitlab.com/`*`user-name`*`/`*`repository-name`***.
   - Dalam acara webhook sumber utama:
     - Untuk Webhook - opsional, pilih Bang un kembali setiap kali perubahan kode didorong ke repositori ini.
     - Pilih Konfigur asi tambahan dan untuk Pembuatan manual - opsional, pilih Buat webhook secara manual untuk repositori ini di GitLab konsol. .
3. Lanjutkan dengan nilai default dan kemudian pilih Buat proyek build. Catat URL Payload dan nilai Rahasia karena Anda akan menggunakannya nanti.
4. Buka GitLab konsol di `https://gitlab.com/`*`user-name`*`/`*`repository-name`*`/-/hooks` dan pilih Tambahkan webhook baru.
   - Untuk URL, masukkan nilai URL Payload yang Anda catat sebelumnya.
   - Untuk token Rah asia, masukkan nilai Rahasia yang Anda catat sebelumnya.
   - Konfigurasikan peristiwa individual yang akan mengirim muatan webhook ke. CodeBuild Untuk Pemicu, pilih dari acara berikut: Acara push, Meng gabungkan acara permintaan, acara rilis, dan acara pekerjaan. Untuk mempelajari lebih lanjut tentang jenis acara yang didukung oleh CodeBuild, lihatGitLab acara webhook.
5. Pilih Tambahkan webhook.

GitLab acara webhook

Anda dapat menggunakan grup filter webhook untuk menentukan peristiwa GitLab webhook mana yang memicu build. Misalnya, Anda dapat menentukan bahwa build hanya dipicu untuk perubahan pada cabang tertentu.

Anda dapat membuat satu atau beberapa grup filter webhook untuk menentukan peristiwa webhook mana yang memicu build. Build dipicu jika ada grup filter yang dievaluasi menjadi true, yang terjadi ketika semua filter dalam grup mengevaluasi ke true. Ketika Anda membuat grup filter, tentukan:

Sebuah acara

Untuk GitLab, Anda dapat memilih satu atau beberapa acara berikut:`PUSH`,,`PULL_REQUEST_CREATED`, `PULL_REQUEST_UPDATED``PULL_REQUEST_MERGED`,`PULL_REQUEST_REOPENED`,`PULL_REQUEST_CLOS` dan`WORKFLOW_JOB_QUEUED`.

Jenis acara webhook ada di header di `X-GitLab-Event` lapangan. Tabel berikut menunjukkan bagaimana nilai `X-GitLab-Event` header dipetakan ke jenis acara. Untuk acara `Merge Request Hook` webhook, payload `object_atttributes.action` akan berisi informasi tambahan tentang jenis permintaan gabungan.

`X-GitLab-Event` Nilai header	**`object_atttributes.action`**	Tipe peristiwa
`Push Hook`	N/A	`PUSH`
`Merge Request Hook`	terbuka	`PULL_REQUEST_CREATED`
`Merge Request Hook`	perbarui	`PULL_REQUEST_UPDATED`
`Merge Request Hook`	bermerger	`PULL_REQUEST_MERGED`
`Merge Request Hook`	buka kembali	`PULL_REQUEST_REOPENED`
`Merge Request Hook`	tutup	`PULL_REQUEST_CLOSED`
`Release Hook`	buat, perbarui	`RELEASED`
`Job Hook`	N/A	`WORKFLOW_JOB_QUEUED`

Karena`PULL_REQUEST_MERGED`, jika permintaan tarik digabungkan dengan strategi squash dan cabang permintaan tarik ditutup, komit permintaan tarik asli tidak ada lagi. Dalam hal ini, variabel

`CODEBUILD_WEBHOOK_MERGE_COMMIT` lingkungan berisi pengidentifikasi komit gabungan yang dihancurkan.

Satu atau lebih filter opsional

Gunakan ekspresi reguler untuk menentukan filter. Agar suatu peristiwa memicu build, setiap filter dalam grup yang terkait dengannya harus dievaluasi menjadi true.

CodeBuild mengevaluasi pola filter menggunakan sintaks ekspresi RE2 reguler, yang tidak mendukung lookahead, lookbehind, backreference, atau grup atom. Untuk informasi selengkapnya, lihat halaman RE2 Sintaks di GitHub situs web.

`ACTOR_ACCOUNT_ID`(`ACTOR_ID`di konsol)

Peristiwa webhook memicu build ketika ID GitLab akun cocok dengan pola ekspresi reguler. Nilai ini muncul di `account_id` properti `actor` objek di payload filter webhook.

`HEAD_REF`

Peristiwa webhook memicu build ketika referensi kepala cocok dengan pola ekspresi reguler (misalnya, `refs/heads/branch-name` dan`refs/tags/tag-name`). `HEAD_REF`Filter mengevaluasi nama referensi Git untuk cabang atau tag. Nama cabang atau tag muncul di `name` bidang `new` objek di `push` objek payload webhook. Untuk peristiwa pull request, nama cabang muncul di `name` bidang di `branch` objek `source` objek di payload webhook.

`BASE_REF`

Peristiwa webhook memicu build ketika referensi dasar cocok dengan pola ekspresi reguler. Filter `BASE_REF` dapat digunakan untuk peristiwa permintaan pull saja (misalnya, `refs/heads/branch-name`). `BASE_REF`Filter mengevaluasi nama referensi Git untuk cabang. Nama cabang muncul di `name` bidang `branch` objek di `destination` objek di payload webhook.

`FILE_PATH`

Webhook memicu build ketika jalur file yang diubah cocok dengan pola ekspresi reguler.

> **Note**
>
> `FILE_PATH`Filter dievaluasi terhadap 100 file pertama yang diubah dalam permintaan push atau pull. Untuk memastikan filter diterapkan sebagaimana dimaksud, simpan jumlah file yang diubah dalam batas ini. Untuk peristiwa pull request, sebaiknya

gunakan kebijakan build pull request untuk mengontrol permintaan tarik mana yang dapat memulai build.

`COMMIT_MESSAGE`

Webhook memicu build ketika pesan komit kepala cocok dengan pola ekspresi reguler.

`WORKFLOW_NAME`

Webhook memicu build ketika nama alur kerja cocok dengan pola ekspresi reguler.

> **Note**
>
> Anda dapat menemukan payload webhook di pengaturan webhook repositori Anda. GitLab

Topik

- Filter acara GitLab webhook (konsol)
- Filter peristiwa GitLab webhook (SDK)
- Filter acara GitLab webhook (CloudFormation)

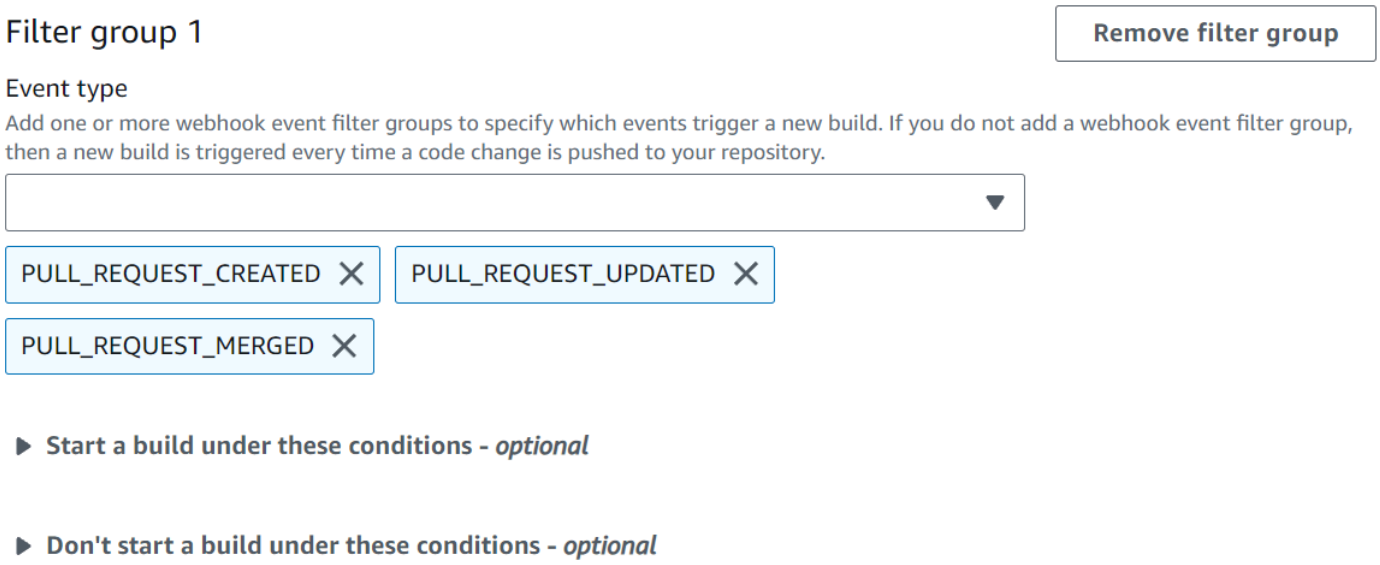
Filter acara GitLab webhook (konsol)

Gunakan petunjuk berikut untuk menggunakan untuk Konsol Manajemen AWS memfilter peristiwa webhook. Untuk informasi selengkapnya tentang acara GitLab webhook, lihatGitLab acara webhook.

1. Pilih Bangun kembali setiap kali perubahan kode didorong ke repositori ini saat Anda membuat proyek Anda.
2. Dari Jenis acara, pilih satu atau beberapa acara.
3. Untuk memfilter saat peristiwa memicu build, di bawah Mulai build dalam kondisi ini, tambahkan satu atau beberapa filter opsional.
4. Untuk memfilter saat peristiwa tidak dipicu, di bawah Jangan memulai build dalam kondisi ini, tambahkan satu atau beberapa filter opsional.
5. Pilih Tambahkan grup filter untuk menambahkan grup filter lain.

Untuk informasi selengkapnya, lihat Buat proyek build (konsol) dan WebhookFilter di Refer AWS CodeBuild ensi API.

Dalam contoh ini, grup filter webhook memicu build untuk permintaan tarik saja:



Menggunakan contoh dua grup filter, build dipicu ketika salah satu atau keduanya mengevaluasi menjadi true:

- Grup filter pertama menentukan permintaan tarik yang dibuat atau diperbarui pada cabang dengan nama referensi Git yang cocok dengan ekspresi reguler `^refs/heads/main$` dan referensi kepala yang cocok`^refs/heads/branch1!`.
- Grup filter kedua menentukan permintaan push pada cabang dengan nama referensi Git yang cocok dengan ekspresi reguler`^refs/heads/branch1$`.

Webhook event filter groups

A build is triggered if any filter group evaluates to true, which occurs when all the filters in the group evaluate to true.

Filter group 1

Remove filter group

Event type

Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

PULL_REQUEST_CREATED ✕ PULL_REQUEST_UPDATED ✕

▼ **Start a build under these conditions** - *optional*

Add filter

Filter 1

Type

HEAD_REF

Pattern

^refs/heads/branch1$

Remove

Filter 2

Type

BASE_REF

Pattern

^refs/heads/main

Remove

▶ **Don't start a build under these conditions** - *optional*

Filter group 2

Remove filter group

Event type

Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

PUSH ✕

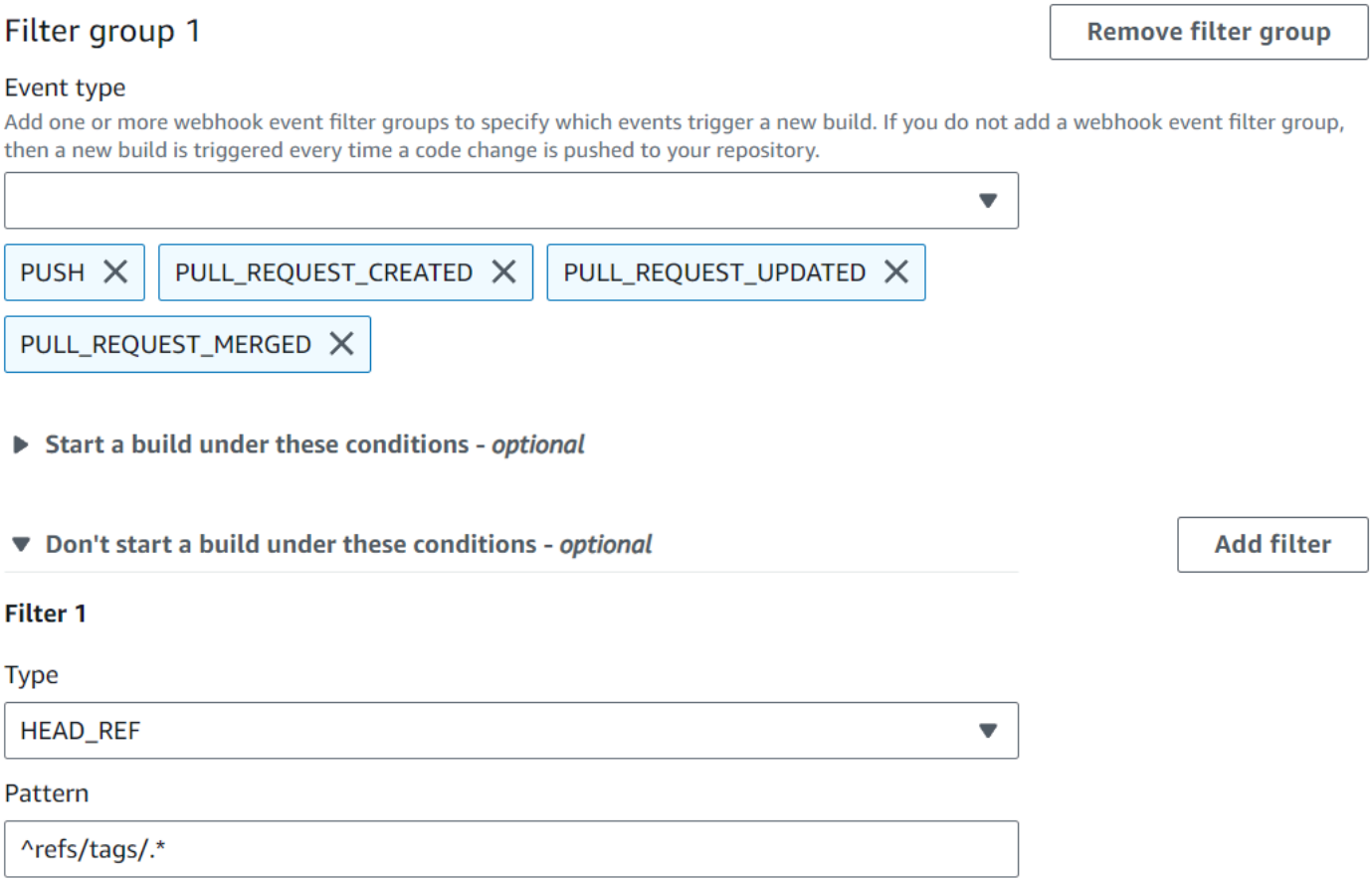
▼ **Start a build under these conditions** - *optional*

Add filter

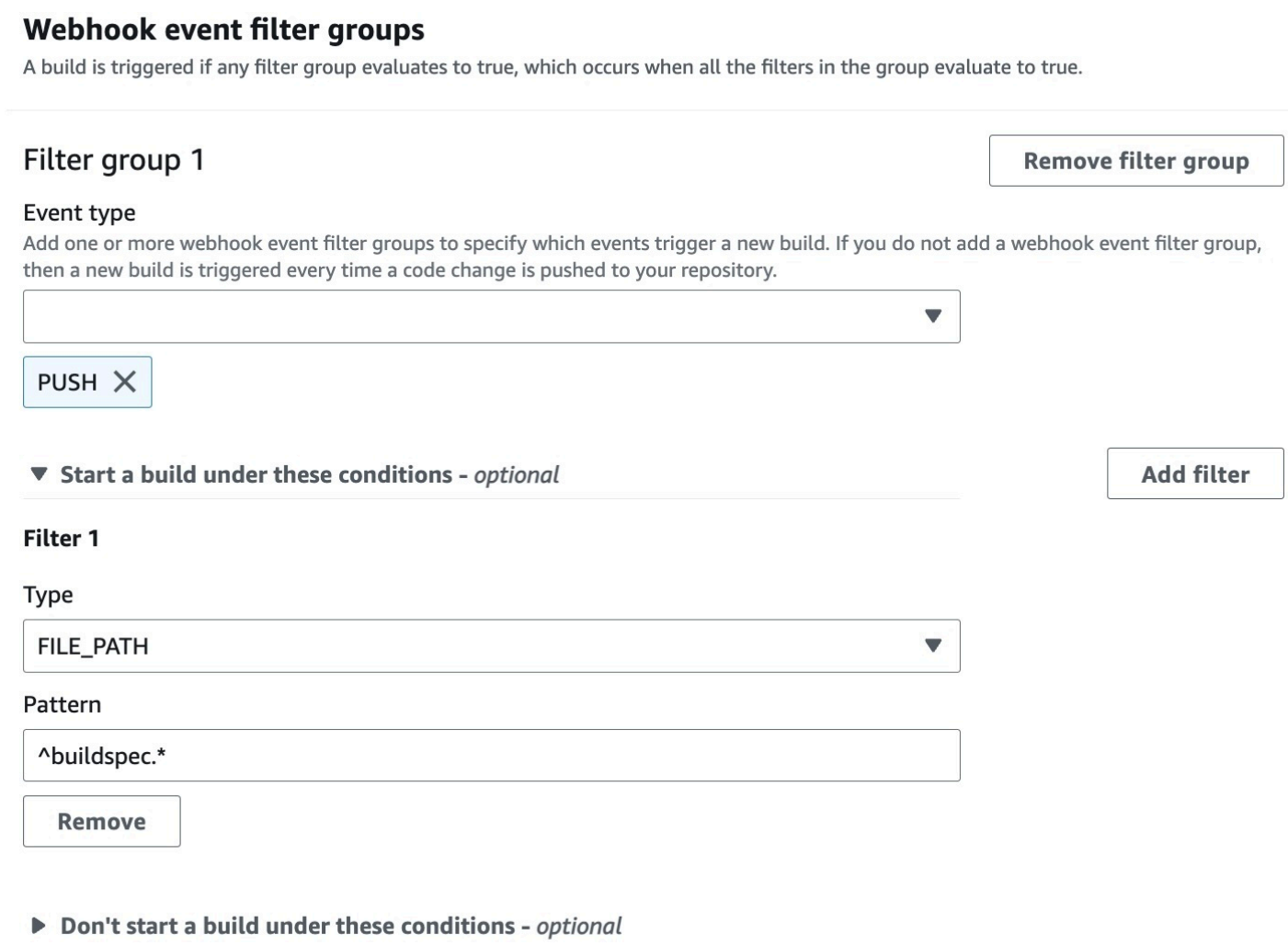
Filter 1

Type

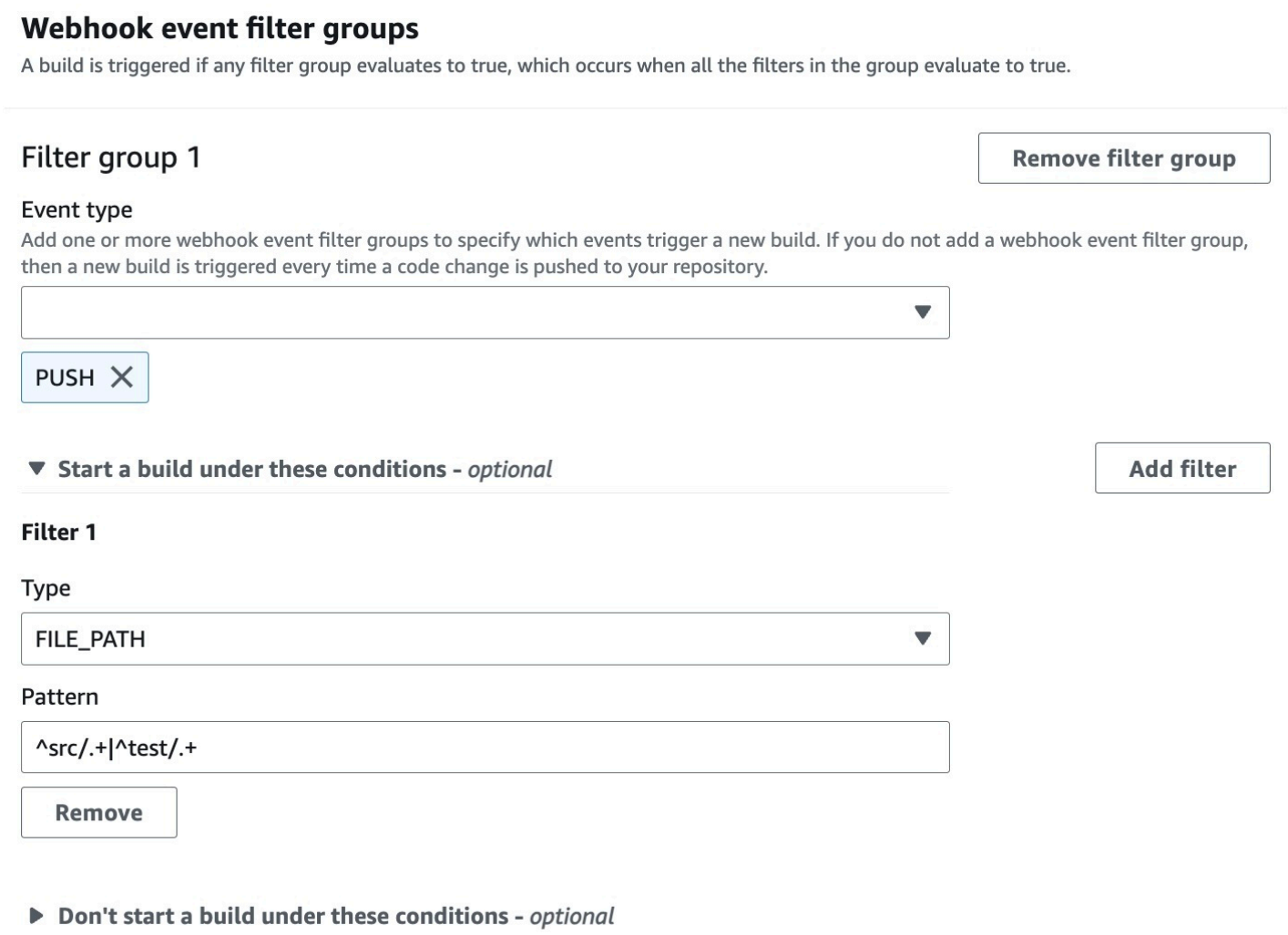
Dalam contoh ini, grup filter webhook memicu build untuk semua permintaan kecuali peristiwa tag.



Dalam contoh ini, grup filter webhook memicu build hanya ketika file dengan nama yang cocok dengan ekspresi reguler `^buildspec.*` berubah.

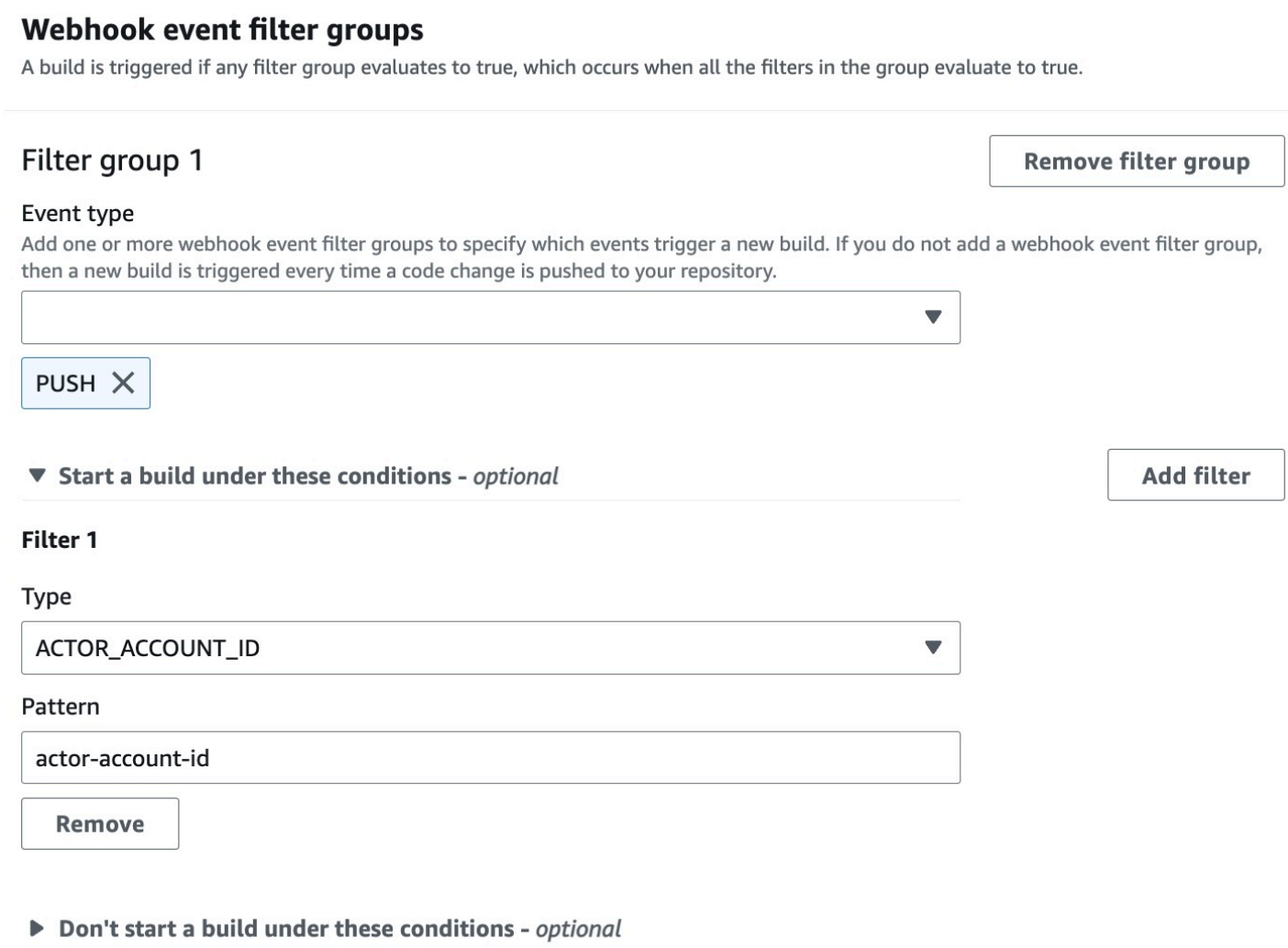


Dalam contoh ini, grup filter webhook memicu build hanya ketika file diubah di dalam `test` folder `src` atau.

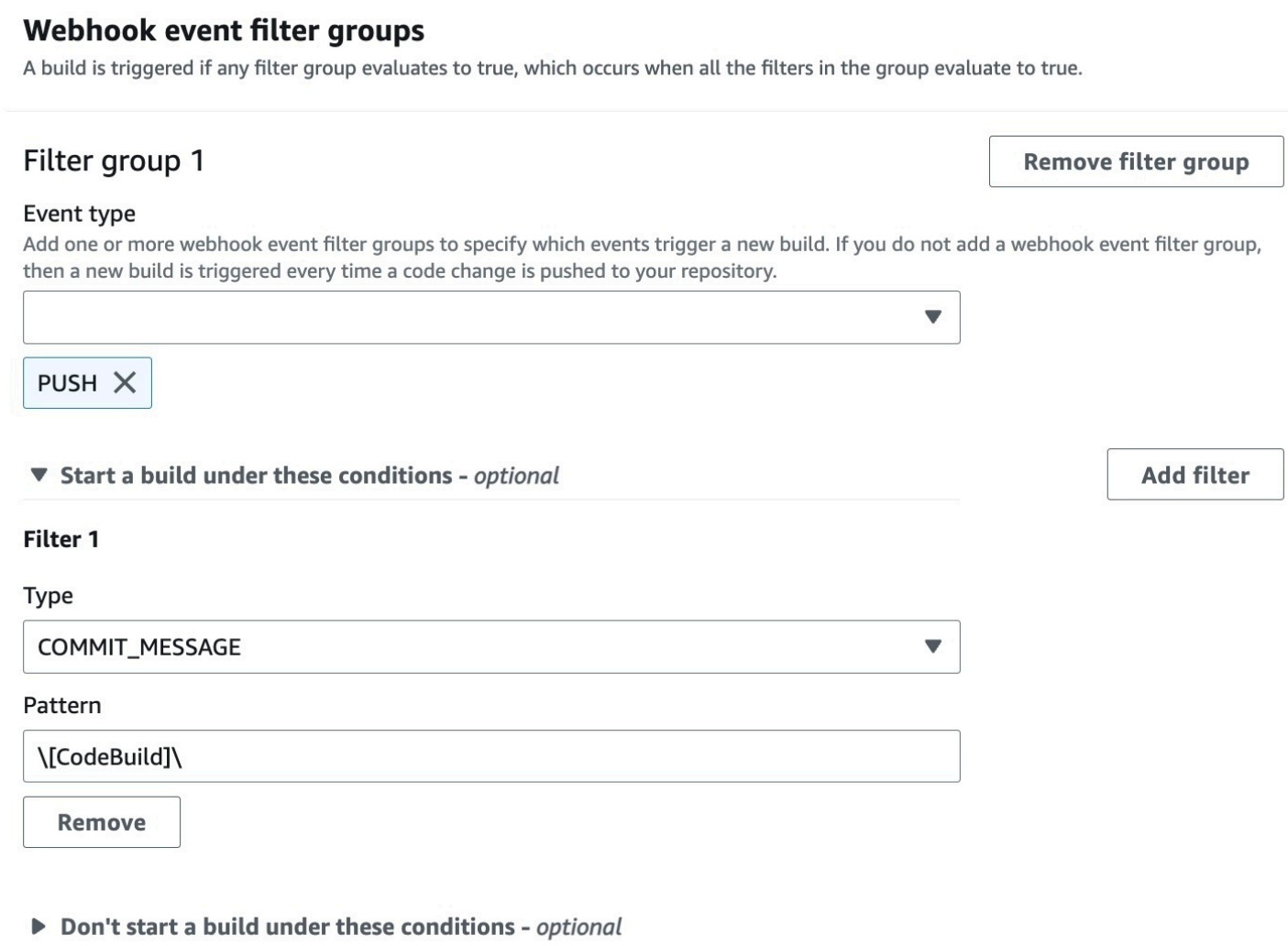


Dalam contoh ini, grup filter webhook memicu build hanya ketika perubahan dilakukan oleh GitLab pengguna yang tidak memiliki ID akun yang cocok dengan ekspresi `actor-account-id` reguler.

> **Note**
>
> Untuk informasi tentang cara menemukan ID GitLab akun Anda, lihat https://api.github.com/users/*`user-name`*, di *`user-name`* mana nama GitLab pengguna Anda.



Dalam contoh ini, grup filter webhook memicu build untuk acara push ketika pesan komit kepala cocok dengan ekspresi `\[CodeBuild\]` reguler.



Filter peristiwa GitLab webhook (SDK)

Untuk menggunakan AWS CodeBuild SDK untuk memfilter peristiwa webhook, gunakan `filterGroups` bidang dalam sintaks permintaan metode `UpdateWebhook` API `CreateWebhook` atau. Untuk informasi selengkapnya, lihat WebhookFilter di dalam Referensi API CodeBuild .

Untuk informasi selengkapnya tentang acara GitLab webhook, lihatGitLab acara webhook.

Untuk membuat filter webhook yang memicu build hanya untuk permintaan tarik, masukkan yang berikut ini ke dalam sintaks permintaan:

```
"filterGroups": [
  [
    {
      "type": "EVENT",
```

```
      "pattern": "PULL_REQUEST_CREATED, PULL_REQUEST_UPDATED, PULL_REQUEST_MERGED"
    }
  ]
]
```

Untuk membuat filter webhook yang memicu build untuk cabang tertentu saja, gunakan `pattern` parameter untuk menentukan ekspresi reguler untuk memfilter nama cabang. Menggunakan contoh dua grup filter, build dipicu ketika salah satu atau keduanya mengevaluasi menjadi true:

- Grup filter pertama menentukan permintaan tarik yang dibuat atau diperbarui pada cabang dengan nama referensi Git yang cocok dengan ekspresi reguler `^refs/heads/main$` dan referensi kepala yang cocok`^refs/heads/myBranch$`.
- Grup filter kedua menentukan permintaan push pada cabang dengan nama referensi Git yang cocok dengan ekspresi reguler`^refs/heads/myBranch$`.

```
"filterGroups": [
  [
    {
      "type": "EVENT",
      "pattern": "PULL_REQUEST_CREATED, PULL_REQUEST_UPDATED"
    },
    {
      "type": "HEAD_REF",
      "pattern": "^refs/heads/myBranch$"
    },
    {
      "type": "BASE_REF",
      "pattern": "^refs/heads/main$"
    }
  ],
  [
    {
      "type": "EVENT",
      "pattern": "PUSH"
    },
    {
      "type": "HEAD_REF",
      "pattern": "^refs/heads/myBranch$"
    }
  ]
```

```
]
```

Anda dapat menggunakan `excludeMatchedPattern` parameter untuk menentukan peristiwa mana yang tidak memicu build. Dalam contoh ini, build dipicu untuk semua permintaan kecuali peristiwa tag.

```
"filterGroups": [
  [
    {
      "type": "EVENT",
      "pattern": "PUSH, PULL_REQUEST_CREATED, PULL_REQUEST_UPDATED,
 PULL_REQUEST_MERGED"
    },
    {
      "type": "HEAD_REF",
      "pattern": "^refs/tags/.*",
      "excludeMatchedPattern": true
    }
  ]
]
```

Anda dapat membuat filter yang memicu build hanya ketika perubahan dilakukan oleh GitLab pengguna dengan ID akun`actor-account-id`.

> **Note**
>
> Untuk informasi tentang cara menemukan ID GitLab akun Anda, lihat https://api.github.com/users/*`user-name`*, di *`user-name`* mana nama GitLab pengguna Anda.

```
"filterGroups": [
  [
    {
      "type": "EVENT",
      "pattern": "PUSH, PULL_REQUEST_CREATED, PULL_REQUEST_UPDATED,
 PULL_REQUEST_MERGED"
    },
    {
      "type": "ACTOR_ACCOUNT_ID",
      "pattern": "actor-account-id"
    }
```

```
  ]
]
```

Anda dapat membuat filter yang memicu build hanya ketika file dengan nama yang cocok dengan ekspresi reguler dalam `pattern` argumen berubah. Dalam contoh ini, grup filter menentukan bahwa build dipicu hanya ketika file dengan nama yang cocok dengan ekspresi reguler `^buildspec.*` berubah.

```
"filterGroups": [
  [
    {
      "type": "EVENT",
      "pattern": "PUSH"
    },
    {
      "type": "FILE_PATH",
      "pattern": "^buildspec.*"
    }
  ]
]
```

Dalam contoh ini, grup filter menentukan bahwa build dipicu hanya ketika file diubah dalam `test` folder `src` atau.

```
"filterGroups": [
    [
        {
            "type": "EVENT",
            "pattern": "PUSH"
        },
        {
            "type": "FILE_PATH",
            "pattern": "^src/.+|^test/.+"
        }
    ]
]
```

Anda dapat membuat filter yang memicu build hanya ketika pesan komit kepala cocok dengan ekspresi reguler dalam argumen pola. Dalam contoh ini, grup filter menentukan bahwa build dipicu hanya ketika pesan komit kepala dari peristiwa push cocok dengan ekspresi reguler`\[CodeBuild \]`.

```
  "filterGroups": [
    [
      {
        "type": "EVENT",
        "pattern": "PUSH"
      },
      {
        "type": "COMMIT_MESSAGE",
        "pattern": "\[CodeBuild\]"
      }
    ]
  ]
```

Filter acara GitLab webhook (CloudFormation)

Untuk menggunakan CloudFormation template untuk memfilter peristiwa webhook, gunakan `FilterGroups` properti AWS CodeBuild proyek. Untuk informasi selengkapnya tentang acara GitLab webhook, lihatGitLab acara webhook.

YAML-formatted Bagian berikut dari CloudFormation template membuat dua grup filter. Bersama-sama, mereka memicu pembuatan ketika salah satu atau keduanya mengevaluasi ke BETUL:

- Grup filter pertama menentukan permintaan tarik dibuat atau diperbarui pada cabang dengan nama referensi Git yang cocok dengan ekspresi reguler `^refs/heads/main$` oleh GitLab pengguna yang tidak memiliki ID akun`12345`.
- Grup filter kedua menentukan permintaan push dibuat pada cabang dengan nama referensi Git yang cocok dengan ekspresi reguler`^refs/heads/.*`.
- Grup filter ketiga menentukan permintaan push dengan pesan komit kepala yang cocok dengan ekspresi reguler`\[CodeBuild\]`.
- Grup filter keempat menentukan permintaan pekerjaan alur kerja Ac GitHub tions dengan nama alur kerja yang cocok dengan ekspresi reguler`\[CI-CodeBuild\]`.

```
CodeBuildProject:
  Type: AWS::CodeBuild::Project
  Properties:
    Name: MyProject
    ServiceRole: service-role
    Artifacts:
```

```
      Type: NO_ARTIFACTS
    Environment:
      Type: LINUX_CONTAINER
      ComputeType: BUILD_GENERAL1_SMALL
      Image: aws/codebuild/standard:5.0
    Source:
      Type: GITLAB
      Location: source-location
    Triggers:
      Webhook: true
      FilterGroups:
        - - Type: EVENT
            Pattern: PULL_REQUEST_CREATED,PULL_REQUEST_UPDATED
          - Type: BASE_REF
            Pattern: ^refs/heads/main$
            ExcludeMatchedPattern: false
          - Type: ACTOR_ACCOUNT_ID
            Pattern: 12345
            ExcludeMatchedPattern: true
        - - Type: EVENT
            Pattern: PUSH
          - Type: HEAD_REF
            Pattern: ^refs/heads/.*
        - - Type: EVENT
            Pattern: PUSH
          - Type: COMMIT_MESSAGE
            Pattern: \[CodeBuild\]
        - - Type: EVENT
            Pattern: WORKFLOW_JOB_QUEUED
          - Type: WORKFLOW_NAME
            Pattern: \[CI-CodeBuild\]
```

Buildkite webhook manual

Saat ini, CodeBuild semua webhook Buildkite harus dibuat secara manual. CodeBuildmengembalikan URL payload sebagai bagian dari panggilan untuk membuat webhook, yang dapat digunakan untuk membuat webhook secara manual di Buildkite.

Gunakan prosedur berikut untuk membuat webhook manual Buildkite.

Untuk membuat CodeBuild proyek dengan webhook

1. Buka AWS CodeBuild konsol di https://console.aws.amazon.com/codesuite/codebuild/home.

2. Buat proyek pembangunan. Untuk informasi selengkapnya, lihat Buat proyek build (konsol) dan Jalankan build (konsol).
3. Di konfigurasi proyek, pilih proyek Runner.

   Dalam Pelari:

   - Untuk penyedia Runner, pilih Buildkite.
   - Untuk token agen Buildkite, pilih Buat token agen baru dengan menggunakan halaman buat rahasia. Anda akan diminta untuk membuat rahasia baru di AWS Secrets Manager dengan nilai rahasia sama dengan token agen Buildkite yang Anda buat di atas.
   - (Opsional) Jika Anda ingin menggunakan kredenSIAL ter CodeBuild kelola untuk pekerjaan Anda, pilih penyedia repositori sumber pekerjaan Anda di bawah opsi kredensi sumber Buildkite dan verifikasi bahwa kredenSIAL telah dikonfigurasi untuk akun Anda. Selain itu, verifikasi bahwa pipeline Buildkite Anda menggunakan Checkout menggunakan HTTPS.
4. • Di Lingkungan:
   - Pilih gambar Lingkungan yang didukung dan Compute. Perhatikan bahwa Anda memiliki opsi untuk mengganti setelan gambar dan instance dengan menggunakan label di ALUR KERJA Tindakan YAM GitHub L Anda. Untuk informasi selengkapnya, lihat Langkah 2: Perbarui alur kerja GitHub Tindakan Anda YAMB
   - Dalam Buildspec:
     - Perhatikan bahwa buildspec Anda akan diabaikan kecuali `buildspec-override:true` ditambahkan sebagai label. Sebagai gantinya, CodeBuild akan mengesampingkannya untuk menggunakan perintah yang akan mengatur pelari yang di-host sendiri.
5. Lanjutkan dengan nilai default dan kemudian pilih Buat proyek build.
6. Simpan URL Payload dan nilai Rahasia dari popup Buat Webhook. Ikuti petunjuk di popup untuk membuat webhook organisasi Buildkite baru.

Persetujuan komentar permintaan tarik

CodeBuild mendukung kebijakan pembuatan permintaan tarik yang memberikan kontrol tambahan atas build yang dipicu oleh permintaan tarik. Anda mungkin tidak ingin secara otomatis membuat permintaan tarik dari pengguna yang tidak dikenal sampai perubahannya dapat ditinjau. Fitur ini memungkinkan Anda untuk meminta salah satu anggota tim Anda untuk terlebih dahulu meninjau kode dan kemudian menjalankan pipeline. Ini biasanya digunakan sebagai langkah keamanan ketika membangun kode yang diajukan oleh kontributor yang tidak dikenal.

Kebijakan pembuatan permintaan tarik memungkinkan Anda mengontrol kapan CodeBuild memicu build untuk permintaan tarik berdasarkan izin dan status persetujuan kontributor. Ini sangat penting untuk repositori publik atau repositori yang menerima kontribusi dari kolaborator eksternal.

Saat diaktifkan, fitur ini memastikan bahwa build hanya dipicu untuk permintaan tarik ketika:

- Permintaan tarik dibuat oleh kontributor tepercaya.
- Kontributor tepercaya menyetujui permintaan tarik dengan memposting komentar tertentu.

Cara kerjanya

Kontributor tepercaya

Kontributor tepercaya adalah pengguna yang perannya saat ini dalam sistem kontrol sumber ditetapkan dalam kebijakan berbasis permintaan tarik sebagai peran pemberi persetujuan. Ketika kontributor tepercaya membuat permintaan tarik, CodeBuild memicu build secara otomatis, mempertahankan perilaku saat ini.

Kontributor yang tidak dipercaya

Kontributor yang tidak tepercaya adalah pengguna yang perannya tidak ditetapkan dalam daftar peran pemberi persetujuan. Saat kontributor yang tidak tepercaya membuat permintaan tarik:

1. CodeBuild menandai status build sebagai “Gagal” dengan pesan “Persetujuan permintaan tarik diperlukan untuk memulai build”.
2. Kontributor tepercaya harus meninjau perubahan dan mengirim komentar `/codebuild_run(<SHA_OF_THE_LATEST_COMMIT>)` untuk memicu build. Misalnya, `/codebuild_run(046e8b67481d53bdc86c3f6affdd5d1afae6d369)`.
3. CodeBuild memvalidasi izin komentator dan memicu build jika disetujui.
4. Hasil build dilaporkan kembali di halaman permintaan tarik.

Sintaks persetujuan komentar

Kontributor tepercaya dapat menyetujui build menggunakan format komentar berikut:

- `/codebuild_run(046e8b67481d53bdc86c3f6affdd5d1afae6d369)`- Pemicu dibangun di atas komit SHA yang ditentukan.

Konfigurasi

Perilaku default

Kebijakan pembuatan permintaan tarik diaktifkan secara default untuk semua CodeBuild proyek yang baru dibuat.

Parameter API:

Kebijakan pembuatan permintaan tarik dikonfigurasi menggunakan `PullRequestBuildPolicy` parameter dalam tindakan berikut:

- `CreateWebhook`
- `UpdateWebhook`

`PullRequestBuildPolicy`struktur

```
{
    "requiresCommentApproval": "string",
    "approverRoles": ["string", ...]
}
```

`requiresCommentApproval`

Menentukan kapan persetujuan berbasis komentar diperlukan sebelum memicu pembuatan permintaan tarik. Pengaturan ini menentukan apakah build berjalan secara otomatis atau memerlukan persetujuan eksplisit melalui komentar.

Tipe: String

Nilai valid:

- `DISABLED`- Membangun pemicu secara otomatis tanpa memerlukan persetujuan komentar.
- `FORK_PULL_REQUESTS`- Hanya permintaan tarik dari repositori bercabang yang memerlukan persetujuan komentar (kecuali kontributor adalah salah satu peran penyetujui).
- `ALL_PULL_REQUESTS`- Semua permintaan tarik memerlukan persetujuan komentar sebelum build dijalankan (kecuali kontributor adalah salah satu peran pemberi persetujuan). Ini adalah nilai default.

`approverRoles`

Daftar peran repositori yang memiliki hak persetujuan untuk pembuatan permintaan tarik saat persetujuan komentar diperlukan. Hanya pengguna dengan peran ini yang dapat memberikan

persetujuan komentar yang valid. Jika kontributor pull request adalah salah satu peran ini, build pull request mereka akan dipicu secara otomatis.

Tipe: Array string

Nilai yang valid untuk GitHub proyek (nilai-nilai dipetakan ke GitHub peran):

- `GITHUB_ADMIN`- Administrator repositori
- `GITHUB_MAINTAIN`- Pengelola repositori
- `GITHUB_WRITE`- Pengguna dengan izin menulis
- `GITHUB_TRIAGE`- Pengguna dengan izin triage
- `GITHUB_READ`- Pengguna dengan izin baca
- Default: `["GITHUB_ADMIN", "GITHUB_MAINTAIN", "GITHUB_WRITE"]`

Nilai yang valid untuk GitLab proyek (nilai-nilai dipetakan ke GitLab peran):

- `GITLAB_OWNER`- Pemilik repositori
- `GITLAB_MAINTAINER`- Pengelola repositori
- `GITLAB_DEVELOPER`- Pengguna dengan izin pengembang
- `GITLAB_REPORTER`- Pengguna dengan izin reporter
- `GITLAB_PLANNER`- Pengguna dengan izin perencana
- `GITLAB_GUEST` - Pengguna dengan izin tamu
- Default: `["GITLAB_OWNER", "GITLAB_MAINTAINER", "GITLAB_DEVELOPER"]`

Nilai yang valid untuk proyek Bitbucket (nilai-nilai dipetakan ke peran Bitbucket):

- `BITBUCKET_ADMIN` - Administrator repositori
- `BITBUCKET_WRITE`- Pengguna dengan izin menulis
- `BITBUCKET_READ` - Pengguna dengan izin baca
- Default: `["BITBUCKET_ADMIN", "BITBUCKET_WRITE"]`

Peran GitHub Perusahaan Kustom

CodeBuild memet GitHub akan peran repositori khusus Enterprise (GHE) ke `GITHUB_*` nilai standar berdasarkan tingkat izin tertinggi yang diberikan untuk peran kustom. CodeBuild mengevaluasi izin dari hak istimewa tertinggi ke terendah, dan izin pertama yang diaktifkan menentukan peran yang dipetakan.

- `admin` → `GITHUB_ADMIN`
- `maintain` → `GITHUB_MAINTAIN`

- push(tulis) → GITHUB_WRITE
- triage → GITHUB_TRIAGE
- pull(baca) → GITHUB_READ

Misalnya, peran khusus yang memberikan keduanya push dan triage izin dipetakan ke GITHUB_WRITE karena push memiliki hak istimewa yang lebih tinggi daripadatriage.

Jika peran kustom tidak memiliki izin yang diakui (admin, maintenance, push, triage, pull all dinonaktifkan), pengguna tidak cocok dengan peran penyetujui apa pun dan diperlakukan sebagai kontributor yang tidak tepercaya.

Peran GitHub standar (Admin, Maintain, Write, Triage, Read) dipetakan langsung ke GITHUB_* nilai yang sesuai tanpa resolusi berbasis izin.

Contoh

Aktifkan persetujuan komentar untuk semua permintaan tarik

Untuk menggunakan AWS CodeBuild SDK untuk mengaktifkan atau menonaktifkan kebijakan Pull Request Build untuk webhook, gunakan pullRequestBuildPolicy bidang dalam sintaks permintaan metode UpdateWebhook API CreateWebhook or. Untuk informasi selengkapnya, lihat WebhookFilter di dalam Referensi API CodeBuild .

Pengguna dengan GitHub peran Admin, Maintain, dan Write akan diperlakukan sebagai kontributor tepercaya.

```
"pullRequestBuildPolicy": {
    "requiresCommentApproval": "ALL_PULL_REQUESTS",
    "approverRoles": ["GITHUB_ADMIN", "GITHUB_MAINTAIN", "GITHUB_WRITE"]
}
```

Aktifkan persetujuan komentar hanya untuk admin dan pengelola repositori

Pengguna dengan GitHub peran Admin dan Maintain akan diperlakukan sebagai kontributor tepercaya.

```
"pullRequestBuildPolicy": {
    "requiresCommentApproval": "FORK_PULL_REQUESTS",
    "approverRoles": ["GITHUB_ADMIN", "GITHUB_MAINTAIN"]
}
```

Nonaktifkan persetujuan komentar

```
"pullRequestBuildPolicy": {
    "requiresCommentApproval": "DISABLED"
}
```

AWS CloudFormation

Untuk menggunakan AWS CloudFormation template untuk mengaktifkan atau menonaktifkan kebijakan Pull Request Build untuk webhook, gunakan PullRequestBuildPolicy properti. Bagian AWS CloudFormation template berikut YAML-formatted membuat proyek dengan webhook yang mengaktifkan Kebijakan Pembuatan Permintaan Pull untuk semua permintaan tarik. Peran Pertahankan dan Admin ditentukan sebagai pemberi persetujuan.

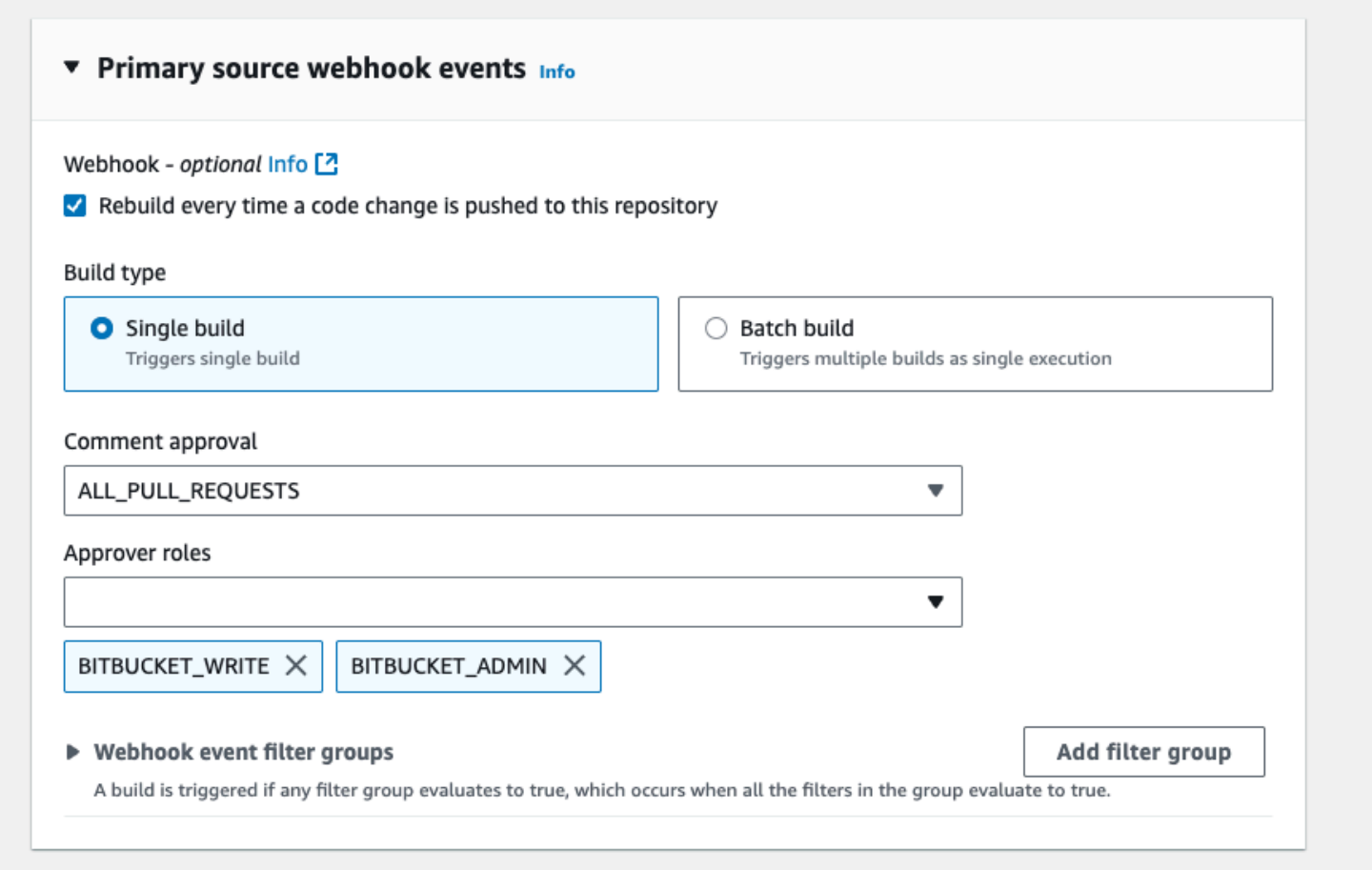
```
CodeBuildProject:
  Type: AWS::CodeBuild::Project
  Properties:
    Name: MyProject
    ServiceRole: service-role
    Artifacts:
      Type: NO_ARTIFACTS
    Environment:
      Type: LINUX_CONTAINER
      ComputeType: BUILD_GENERAL1_SMALL
      Image: aws/codebuild/standard:5.0
    Source:
      Type: BITBUCKET
      Location: source-location
    Triggers:
      Webhook: true
      FilterGroups:
        - - Type: EVENT
            Pattern: PULL_REQUEST_CREATED,PULL_REQUEST_UPDATED
          - Type: BASE_REF
            Pattern: ^refs/heads/main$
            ExcludeMatchedPattern: false
      PullRequestBuildPolicy:
        RequiresCommentApproval: ALL_PULL_REQUESTS
        ApproverRoles:
          - GITHUB_MAINTAIN
          - GITHUB_ADMIN
```

Konfigurasi konsol

Untuk menggunakan AWS Management Console untuk memfilter peristiwa webhook:

1. Untuk persetujuan Komentar, pilih dinonaktifkan atau diaktifkan untuk semua pull request (`ALL_PULL_REQUEST`) atau hanya untuk permintaan tarik dari fork (`FORK_PULL_REQUEST`).
2. Untuk peran Approver, pilih peran repositori yang memiliki hak istimewa persetujuan untuk build permintaan tarik saat persetujuan komentar diperlukan.

Untuk informasi selengkapnya, lihat Buat proyek build (konsol) dan WebhookFilter di Refer CodeBuild ensi API.



Lihat detail proyek build di AWS CodeBuild

Anda dapat menggunakan AWS CodeBuild konsol, AWS CLI, atau AWS SDK untuk melihat detail proyek build. CodeBuild

Topik

- Melihat detail proyek build (konsol)
- Lihat detail proyek build (AWS CLI)
- Melihat detail proyek build (AWS SDK)

Melihat detail proyek build (konsol)

1. Buka AWS CodeBuild konsol di https://console.aws.amazon.com/codesuite/codebuild/home.
2. Di panel navigasi, pilih Membangun proyek.

> **Note**
>
> Secara default, hanya 10 proyek build terbaru yang ditampilkan. Untuk melihat proyek build lainnya, pilih ikon roda gigi, lalu pilih nilai berbeda untuk Proyek per halaman atau gunakan panah mundur dan maju.

3. Dalam daftar proyek build, di kolom Name, pilih link untuk proyek build.
4. Pada *project-name* halaman Build project:, pilih Build details.

Lihat detail proyek build (AWS CLI)

Jalankan perintah batch-get-projects:

```
aws codebuild batch-get-projects --names names
```

Pada perintah sebelumnya, ganti placeholder berikut:

- *names*: String yang diperlukan digunakan untuk menunjukkan satu atau beberapa nama proyek build untuk melihat detailnya. Untuk menentukan lebih dari satu proyek build, pisahkan setiap nama proyek build dengan spasi. Anda dapat menentukan hingga 100 nama proyek build. Untuk mendapatkan daftar proyek pembangunan, lihatLihat daftar nama proyek build (AWS CLI).

Misalnya, jika Anda menjalankan perintah ini:

```
aws codebuild batch-get-projects --names codebuild-demo-project codebuild-demo-project2
 my-other-demo-project
```

Hasil yang mirip dengan berikut ini mungkin muncul di output. Ellipses (`...`) digunakan untuk mewakili data yang dihilangkan untuk singkatnya.

```
{
  "projectsNotFound": [
    "my-other-demo-project"
  ],
  "projects": [
    {
      ...
      "name": codebuild-demo-project,
      ...
    },
    {
      ...
      "name": codebuild-demo-project2",
      ...
    }
  ]
}
```

Pada output sebelumnya, `projectsNotFound` array mencantumkan nama proyek build apa pun yang ditentukan, tetapi tidak ditemukan. `projects`Array mencantumkan detail untuk setiap proyek build tempat informasi ditemukan. Detail proyek build telah dihilangkan dari output sebelumnya untuk singkatnya. Untuk informasi lebih lanjut, lihat output dariBuat proyek build (AWS CLI).

batch-get-projectsPerintah tidak mendukung pemfilteran untuk nilai properti tertentu, tetapi Anda dapat menulis skrip yang menghitung properti untuk proyek. Misalnya, skrip shell Linux berikut menghitung proyek di wilayah saat ini untuk akun saat ini, dan mencetak gambar yang digunakan oleh setiap proyek.

```
#!/usr/bin/sh

# This script enumerates all of the projects for the current account
# in the current region and prints out the image that each project is using.

imageName=""
```

```
function getImageName(){
  local environmentValues=(${1//$'\t'/ })
  imageName=${environmentValues[1]}
}

function processProjectInfo() {
  local projectInfo=$1

  while IFS=$'\t' read -r section value; do
    if [[ "$section" == *"ENVIRONMENT"* ]]; then
      getImageName "$value"
    fi
  done <<< "$projectInfo"
}

# Get the list of projects.
projectList=$(aws codebuild list-projects --output=text)

for projectName in $projectList
do
  if [[ "$projectName" != *"PROJECTS"* ]]; then
    echo "================================================"

    # Get the detailed information for the project.
    projectInfo=$(aws codebuild batch-get-projects --output=text --names
 "$projectName")

    processProjectInfo "$projectInfo"

    printf 'Project "%s" has image "%s"\n' "$projectName" "$imageName"
  fi
done
```

Untuk informasi lebih lanjut tentang menggunakan AWS CLI with AWS CodeBuild, lihatReferensi baris perintah.

Melihat detail proyek build (AWS SDK)

Untuk informasi selengkapnya tentang penggunaan AWS CodeBuild dengan AWS SDK, lihat. AWS Referensi SDK dan alat

Lihat nama proyek build di AWS CodeBuild

Anda dapat menggunakan AWS CodeBuild konsol AWS CLI, atau AWS SDK untuk melihat daftar proyek build di CodeBuild.

Topik

- Melihat daftar nama proyek build (konsol)
- Lihat daftar nama proyek build (AWS CLI)
- Melihat daftar nama proyek build (AWS SDK)

Melihat daftar nama proyek build (konsol)

Anda dapat melihat daftar proyek build di AWS Wilayah di konsol. Informasi mencakup nama, penyedia sumber, repositori, status build terbaru, dan deskripsi, jika ada.

1. Buka AWS CodeBuild konsol di https://console.aws.amazon.com/codesuite/codebuild/home.
2. Di panel navigasi, pilih Membangun proyek.

> **Note**
>
> Secara default, hanya 10 proyek build terbaru yang ditampilkan. Untuk melihat proyek build lainnya, pilih ikon roda gigi, lalu pilih nilai berbeda untuk Proyek per halaman atau gunakan panah mundur dan maju.

Lihat daftar nama proyek build (AWS CLI)

Jalankan perintah list-projects:

```
aws codebuild list-projects --sort-by sort-by --sort-order sort-order --next-
token next-token
```

Pada perintah sebelumnya, ganti placeholder berikut:

- *sort-by*: String opsional digunakan untuk menunjukkan kriteria yang akan digunakan untuk daftar nama proyek build. Nilai yang valid meliputi:
  - `CREATED_TIME`: Buat daftar nama proyek build berdasarkan kapan setiap proyek build dibuat.

  - `LAST_MODIFIED_TIME`: Buat daftar nama proyek build berdasarkan kapan informasi tentang setiap proyek build terakhir diubah.
  - `NAME`: Buat daftar nama proyek build berdasarkan nama masing-masing proyek build.
- *`sort-order`*: String opsional yang digunakan untuk menunjukkan urutan daftar proyek build, berdasarkan*`sort-by`*. Nilai yang valid mencakup `ASCENDING` dan `DESCENDING`.
- *`next-token`*: String opsional. Selama proses sebelumnya, jika ada lebih dari 100 item dalam daftar, hanya 100 item pertama yang dikembalikan, bersama dengan string unik yang disebut token berikutnya. Untuk mendapatkan kumpulan item berikutnya dalam daftar, jalankan perintah ini lagi, tambahkan token berikutnya ke panggilan. Untuk mendapatkan semua item dalam daftar, terus jalankan perintah ini dengan setiap token berikutnya berikutnya, sampai tidak ada lagi token berikutnya yang dikembalikan.

Misalnya, jika Anda menjalankan perintah ini:

```
aws codebuild list-projects --sort-by NAME --sort-order ASCENDING
```

Hasil yang mirip dengan berikut ini mungkin muncul di output:

```
{
  "nextToken": "Ci33ACF6...The full token has been omitted for brevity...U+AkMx8=",
  "projects": [
    "codebuild-demo-project",
    "codebuild-demo-project2",
    ... The full list of build project names has been omitted for brevity ...
    "codebuild-demo-project99"
  ]
}
```

Jika Anda menjalankan perintah ini lagi:

```
aws codebuild list-projects  --sort-by NAME --sort-order ASCENDING --next-token
 Ci33ACF6...The full token has been omitted for brevity...U+AkMx8=
```

Hasil yang mirip dengan berikut ini mungkin muncul di output:

```
{
  "projects": [
    "codebuild-demo-project100",
```

```
    "codebuild-demo-project101",
    ... The full list of build project names has been omitted for brevity ...
    "codebuild-demo-project122"
  ]
}
```

Melihat daftar nama proyek build (AWS SDK)

Untuk informasi selengkapnya tentang penggunaan AWS CodeBuild dengan AWS SDK, lihat. AWS Referensi SDK dan alat

Membangun di AWS CodeBuild

Sebuah build mewakili serangkaian tindakan yang dilakukan oleh AWS CodeBuild untuk membuat artefak keluaran (misalnya, file JAR) berdasarkan sekumpulan artefak input (misalnya, kumpulan file kelas Java).

Aturan berikut berlaku saat Anda menjalankan beberapa build:

- Jika memungkinkan, build berjalan secara bersamaan. Jumlah maksimum build yang berjalan secara bersamaan dapat bervariasi. Untuk informasi selengkapnya, lihat Kuota untuk AWS CodeBuild.
- Jika proyek build memiliki set batas build bersamaan, build mengembalikan kesalahan jika jumlah build yang berjalan mencapai batas build bersamaan untuk proyek. Untuk informasi selengkapnya, lihat Mengaktifkan batas build bersamaan.
- Jika proyek build tidak memiliki set batas build bersamaan, build akan mengantri jika jumlah build yang berjalan mencapai batas build bersamaan untuk platform dan tipe komputasi. Jumlah maksimum build dalam antrian adalah lima kali batas build bersamaan. Untuk informasi selengkapnya, lihat Kuota untuk AWS CodeBuild.

  Build dalam antrian yang tidak dimulai setelah jumlah menit yang ditentukan dalam nilai waktu habisnya dihapus dari antrian. Nilai batas waktu default adalah delapan jam. Anda dapat mengganti batas waktu antrian build dengan nilai antara lima menit dan delapan jam saat Anda menjalankan build Anda. Untuk informasi selengkapnya, lihat Jalankan AWS CodeBuild build secara manual.

  Tidak mungkin untuk memprediksi urutan di mana build yang diantri dimulai.

> **Note**
>
> Anda dapat mengakses riwayat bangunan selama satu tahun.

Anda dapat melakukan tugas-tugas ini saat bekerja dengan build:

Topik

- Jalankan AWS CodeBuild build secara manual

- Jalankan dibangun di AWS Lambda komputasi
- Jalankan build pada armada kapasitas yang dicadangkan
- Jalankan build dalam batch
- Jalankan tes paralel dalam build batch
- Cache dibangun untuk meningkatkan kinerja
- Debug dibangun di AWS CodeBuild
- Hapus build di AWS CodeBuild
- Coba ulang membangun secara manual AWS CodeBuild
- Coba lagi membangun secara otomatis di AWS CodeBuild
- Berhenti membangun AWS CodeBuild
- Hentikan pembuatan batch AWS CodeBuild
- Pemicu AWS CodeBuild dibangun secara otomatis
- Lihat detail build di AWS CodeBuild
- Melihat daftar ID build di AWS CodeBuild
- Melihat daftar ID build untuk proyek build di AWS CodeBuild

Jalankan AWS CodeBuild build secara manual

Anda dapat menggunakan AWS CodeBuild konsol, AWS CLI, atau AWS SDK untuk menjalankan build in CodeBuild.

Topik

- Jalankan build secara lokal dengan agen AWS CodeBuild
- Jalankan build (konsol)
- Jalankan build (AWS CLI)
- Jalankan build batch (AWS CLI)
- Mulai menjalankan build secara otomatis ()AWS CLI
- Berhenti menjalankan build secara otomatis ()AWS CLI
- Jalankan build (AWS SDK)

Jalankan build secara lokal dengan agen AWS CodeBuild

Anda dapat menggunakan AWS CodeBuild agen untuk menjalankan CodeBuild build di mesin lokal. Ada agen yang tersedia untuk platform x86_64 dan ARM.

Anda juga dapat berlangganan untuk menerima pemberitahuan ketika versi baru agen diterbitkan.

Prasyarat

Sebelum Anda mulai, Anda perlu melakukan hal berikut:

- Instal Git di mesin lokal Anda.
- Instal dan atur Docker di mesin lokal Anda.

Siapkan image build

Anda hanya perlu menyiapkan image build saat pertama kali menjalankan agen, atau saat gambar telah berubah.

Untuk mengatur image build

1. Jika Anda ingin menggunakan image Amazon Linux 2 yang dikurasi, Anda dapat menariknya dari repositori Amazon ECR CodeBuild publik di https://gallery.ecr.aws/codebuild/amazonlinux-x86_64-standarddengan perintah berikut:

   ```
   $ docker pull public.ecr.aws/codebuild/amazonlinux-x86_64-standard:4.0
   ```

   Atau, jika Anda ingin menggunakan image Linux lain, lakukan langkah-langkah berikut:

   a. Kloning repo CodeBuild gambar:

      ```
      $ git clone https://github.com/aws/aws-codebuild-docker-images.git
      ```

   b. Ubah ke direktori gambar. Untuk contoh ini, gunakan `aws/codebuild/standard:5.0` gambar:

      ```
      $ cd aws-codebuild-docker-images/ubuntu/standard/5.0
      ```

   c. Bangun gambar. Ini akan memakan waktu beberapa menit.

```
$ docker build -t aws/codebuild/standard:5.0 .
```

2. Unduh CodeBuild agennya.

   Untuk mengunduh agen versi x86_64, jalankan perintah berikut:

   ```
   $ docker pull public.ecr.aws/codebuild/local-builds:latest
   ```

   Untuk mengunduh versi ARM agen, jalankan perintah berikut:

   ```
   $ docker pull public.ecr.aws/codebuild/local-builds:aarch64
   ```

3. CodeBuild Agen tersedia dari https://gallery.ecr.aws/codebuild/local-builds.

   Tanda tangan Secure Hash Algorithm (SHA) untuk agen versi x86_64 adalah:

   ```
   sha256:ccb19bdd7af94e4dc761e4c58c267e9455c28ec68d938086b4dc1cf8fe6b0940
   ```

   Tanda tangan SHA untuk versi ARM agen adalah:

   ```
   sha256:7d7b5d35d2ac4e062ae7ba8c662ffed15229a52d09bd0d664a7816c439679192
   ```

   Anda dapat menggunakan SHA untuk mengidentifikasi versi agen. Untuk melihat tanda tangan SHA agen, jalankan perintah berikut dan cari SHA di bawah`RepoDigests`:

   ```
   $ docker inspect public.ecr.aws/codebuild/local-builds:latest
   ```

Jalankan CodeBuild agen

Untuk menjalankan CodeBuild agen

1. Ubah ke direktori yang berisi sumber proyek build Anda.
2. Unduh skrip codebuild_build.sh:

   ```
   $ curl -O  https://raw.githubusercontent.com/aws/aws-codebuild-docker-images/
   master/local_builds/codebuild_build.sh
   $ chmod +x codebuild_build.sh
   ```

3. Jalankan `codebuild_build.sh` skrip dan tentukan gambar kontainer Anda dan direktori output.

   Untuk menjalankan build x86_64, jalankan perintah berikut:

   ```
   $ ./codebuild_build.sh -i <container-image> -a <output directory>
   ```

   Untuk menjalankan ARM build, jalankan perintah berikut:

   ```
   $ ./codebuild_build.sh -i <container-image> -a <output directory> -l
    public.ecr.aws/codebuild/local-builds:aarch64
   ```

   Ganti *<container-image>* dengan nama gambar kontainer, seperti `aws/codebuild/standard:5.0` atau`public.ecr.aws/codebuild/amazonlinux-x86_64-standard:4.0`.

   Skrip meluncurkan image build dan menjalankan build pada proyek di direktori saat ini. Untuk menentukan lokasi proyek build, tambahkan `-s` *<build project directory>* opsi ke perintah skrip.

Menerima pemberitahuan untuk versi CodeBuild agen baru

Anda dapat berlangganan notifikasi Amazon SNS sehingga Anda akan diberi tahu saat versi baru AWS CodeBuild agen dirilis.

Untuk berlangganan pemberitahuan CodeBuild agen

1. Buka konsol Amazon SNS di https://console.aws.amazon.com/sns/v3/home.
2. Di bilah navigasi, jika belum dipilih, ubah AWS Region ke US East (Virginia N.). Anda harus memilih AWS Wilayah ini karena notifikasi Amazon SNS yang Anda berlangganan dibuat di Wilayah ini.
3. Di panel navigasi, pilih Langganan.
4. Pilih Buat langganan.
5. Di Buat langganan, lakukan hal berikut:

   a. Untuk ARN Topik, salin Amazon Resource Name (ARN) berikut:

```
arn:aws:sns:us-east-1:850632864840:AWS-CodeBuild-Local-Agent-Updates
```

b. Untuk Protokol, pilih Email atau SMS.

c. Untuk Endpoint, pilih tempat (email atau SMS) untuk menerima notifikasi. Masukkan email atau alamat atau nomor telepon, termasuk kode area.

d. Pilih Buat langganan.

e. Pilih Email untuk menerima email yang meminta Anda mengonfirmasi langganan Anda. Ikuti petunjuk di email untuk menyelesaikan langganan Anda.

Jika Anda tidak ingin lagi menerima notifikasi ini, gunakan prosedur berikut untuk berhenti berlangganan.

Untuk berhenti berlangganan pemberitahuan CodeBuild agen

1. Buka konsol Amazon SNS di https://console.aws.amazon.com/sns/v3/home.
2. Di panel navigasi, pilih Langganan.
3. Pilih langganan dan dari Tindakan, pilih Hapus langganan. Ketika diminta untuk mengonfirmasi, pilih Hapus.

Jalankan build (konsol)

Untuk menggunakan AWS CodePipeline untuk menjalankan build dengan CodeBuild, lewati langkah-langkah ini dan ikuti instruksi diGunakan CodeBuild dengan CodePipeline.

1. Buka AWS CodeBuild konsol di https://console.aws.amazon.com/codesuite/codebuild/home.
2. Di panel navigasi, pilih Membangun proyek.
3. Dalam daftar proyek build, pilih proyek build.
4. Anda dapat menjalankan build dengan setelan proyek build default, atau mengganti setelan build hanya untuk build ini.

   a. Jika Anda ingin menjalankan build dengan pengaturan proyek build default, pilih Start build. Membangun segera dimulai.

   b. Jika Anda ingin mengganti setelan proyek build default, pilih Mulai build dengan penggantian. Di halaman Mulai build, Anda dapat mengganti yang berikut ini:

- Membangun konfigurasi
- Sumber
- Pengganti variabel lingkungan

Jika Anda perlu memilih penggantian lanjutan lainnya, pilih Pengganti build lanjutan. Di halaman ini, Anda dapat mengganti yang berikut ini:

- Membangun konfigurasi
- Sumber
- Lingkungan
- Buildspec
- Artefak
- Log

Ketika Anda telah membuat pilihan penggantian, pilih Mulai build.

Untuk informasi rinci tentang build ini, lihatLihat detail build (konsol).

Jalankan build (AWS CLI)

> **Note**
>
> Untuk menggunakan CodePipeline untuk menjalankan build AWS CodeBuild, lewati langkah-langkah ini dan ikuti instruksi diBuat pipeline yang menggunakan CodeBuild (AWS CLI). Untuk informasi lebih lanjut tentang menggunakan AWS CLI with CodeBuild, lihatReferensi baris perintah.

1. Jalan `start-build` kan perintah dengan salah satu cara berikut:

   ```
   aws codebuild start-build --project-name <project-name>
   ```

   Gunakan ini jika Anda ingin menjalankan build yang menggunakan versi terbaru dari artefak input build dan pengaturan proyek build yang ada.

```
aws codebuild start-build --generate-cli-skeleton
```

Gunakan ini jika Anda ingin menjalankan build dengan versi sebelumnya dari artefak input build atau jika Anda ingin mengganti pengaturan untuk artefak keluaran build, variabel lingkungan, buildspec, atau periode batas waktu build default.

2. Jika Anda menjalankan start-build perintah dengan `--project-name` opsi, ganti *<project-name>* dengan nama proyek build, dan kemudian lewati ke langkah 6 dari prosedur ini. Untuk mendapatkan daftar proyek pembangunan, lihatLihat nama proyek build.
3. Jika Anda menjalankan start-build perintah dengan `--idempotency-token` opsi tersebut, pengenal unik yang peka huruf besar-kecil, atau token, disertakan dengan `start-build` permintaan. Token ini berlaku selama 5 menit setelah permintaan. Jika Anda mengulangi `start-build` permintaan dengan token yang sama, tetapi mengubah parameter, CodeBuild mengembalikan kesalahan ketidakcocokan parameter.
4. Jika Anda menjalankan start-build perintah dengan `--generate-cli-skeleton` opsi, JSON-formatted data muncul di output. Salin data ke file (misalnya,*start-build.json*) di lokasi di komputer lokal atau instance tempat AWS CLI diinstal. Ubah data yang disalin agar sesuai dengan format berikut, dan simpan hasil Anda:

```
{
  "projectName": "projectName",
  "sourceVersion": "sourceVersion",
  "artifactsOverride": {
    "type": "type",
    "location": "location",
    "path": "path",
    "namespaceType": "namespaceType",
    "name": "artifactsOverride-name",
    "packaging": "packaging"
  },
  "buildspecOverride": "buildspecOverride",
  "cacheOverride": {
    "location": "cacheOverride-location",
    "type": "cacheOverride-type"
  },
  "certificateOverride": "certificateOverride",
  "computeTypeOverride": "computeTypeOverride",
  "environmentTypeOverride": "environmentTypeOverride",
  "environmentVariablesOverride": {
```

```
    "name": "environmentVariablesOverride-name",
    "value": "environmentVariablesValue",
    "type": "environmentVariablesOverride-type"
  },
  "gitCloneDepthOverride": "gitCloneDepthOverride",
  "imageOverride": "imageOverride",
  "idempotencyToken": "idempotencyToken",
  "insecureSslOverride": "insecureSslOverride",
  "privilegedModeOverride": "privilegedModeOverride",
  "queuedTimeoutInMinutesOverride": "queuedTimeoutInMinutesOverride",
  "reportBuildStatusOverride": "reportBuildStatusOverride",
  "timeoutInMinutesOverride": "timeoutInMinutesOverride",
  "sourceAuthOverride": "sourceAuthOverride",
  "sourceLocationOverride": "sourceLocationOverride",
  "serviceRoleOverride": "serviceRoleOverride",
  "sourceTypeOverride": "sourceTypeOverride"
}
```

Ganti placeholder berikut:

- *projectName*: String yang diperlukan. Nama proyek build yang akan digunakan untuk build ini.
- *sourceVersion*: String opsional. Versi kode sumber yang akan dibangun, sebagai berikut:
  - Untuk Amazon S3, ID versi yang sesuai dengan versi file ZIP input yang ingin Anda buat. Jika *sourceVersion* tidak ditentukan, maka versi terbaru digunakan.
  - Untuk CodeCommit, ID komit yang sesuai dengan versi kode sumber yang ingin Anda buat. Jika tidak *sourceVersion* ditentukan, ID komit HEAD cabang default digunakan. (Anda tidak dapat menentukan nama tag untuk*sourceVersion*, tetapi Anda dapat menentukan ID komit tag.)
  - Untuk GitHub, ID komit, ID permintaan tarik, nama cabang, atau nama tag yang sesuai dengan versi kode sumber yang ingin Anda buat. Jika ID permintaan tarik ditentukan, ia harus menggunakan format `pr/`*pull-request-ID* (misalnya `pr/25`). Jika nama cabang ditentukan, ID melakukan HEAD cabang digunakan. Jika tidak *sourceVersion* ditentukan, ID komit HEAD cabang default digunakan.
  - Untuk Bitbucket, ID commit, nama cabang, atau nama tag yang sesuai dengan versi kode sumber yang ingin Anda buat. Jika nama cabang ditentukan, ID melakukan HEAD cabang digunakan. Jika tidak *sourceVersion* ditentukan, ID komit HEAD cabang default digunakan.

- Placeholder berikut adalah untuk. `artifactsOverride`
  - *`type`*: Opsional. Jenis artefak keluaran build yang menggantikan untuk build ini yang ditentukan dalam proyek build.
  - *`location`*: Opsional. Lokasi artefak keluaran build yang menggantikan untuk build ini yang ditentukan dalam proyek build.
  - *`path`*: Opsional. Jalur artefak keluaran build yang menggantikan untuk build ini yang ditentukan dalam proyek build.
  - *`namespaceType`*: Opsional. Jenis jalur artefak keluaran build yang menggantikan build ini yang ditentukan dalam proyek build.
  - *`name`*: Opsional. Nama artefak keluaran build yang menggantikan untuk build ini yang ditentukan dalam proyek build.
  - *`packaging`*: Opsional. Jenis pengemasan artefak keluaran build yang menggantikan build ini yang ditentukan dalam proyek build.
- *`buildspecOverride`*: Opsional. Deklarasi buildspec yang menggantikan build ini yang ditentukan dalam proyek build. Jika nilai ini disetel, itu bisa berupa definisi buildspec sebaris, jalur ke file buildspec alternatif relatif terhadap nilai variabel `CODEBUILD_SRC_DIR` lingkungan bawaan, atau jalur ke bucket S3. Bucket S3 harus berada di Wil AWS ayah yang sama dengan proyek build. Tentukan file buildspec menggunakan ARN-nya (misalnya,). `arn:aws:s3:::`*`<my-codebuild-sample2>`*`/buildspec.yml` Jika nilai ini tidak disediakan atau diatur ke string kosong, kode sumber harus berisi `buildspec.yml` file di direktori root. Untuk informasi selengkapnya, lihat Nama file buildspec dan lokasi penyimpanan.
- Placeholder berikut adalah untuk. `cacheOverride`
  - *`cacheOverride-location`*: Opsional. Lokasi `ProjectCache` objek untuk build ini yang menggantikan `ProjectCache` objek yang ditentukan dalam proyek build. `cacheOverride`bersifat opsional dan mengambil `ProjectCache` objek. `location`diperlukan dalam suatu `ProjectCache` objek.
  - *`cacheOverride-type`*: Opsional. Jenis `ProjectCache` objek untuk build ini yang menggantikan `ProjectCache` objek yang ditentukan dalam proyek build. `cacheOverride`bersifat opsional dan mengambil `ProjectCache` objek. `type`diperlukan dalam suatu `ProjectCache` objek.
- *`certificateOverride`*: Opsional. Nama sertifikat untuk build ini yang menggantikan yang ditentukan dalam proyek build.

- *environmentTypeOverride*: Opsional. Jenis wadah untuk build ini yang menggantikan yang ditentukan dalam proyek build. String valid saat ini adalah`LINUX_CONTAINER`.
- Placeholder berikut adalah untuk. `environmentVariablesOverride`
  - *environmentVariablesOverride-name*: Opsional. Nama variabel lingkungan dalam proyek build yang nilainya ingin Anda ganti untuk build ini.
  - *environmentVariablesOverride-type*: Opsional. Jenis variabel lingkungan dalam proyek build yang nilainya ingin Anda ganti untuk build ini.
  - *environmentVariablesValue*: Opsional. Nilai variabel lingkungan yang ditentukan dalam proyek build yang ingin Anda ganti untuk build ini.
- *gitCloneDepthOverride*: Opsional. Nilai kedalaman klon Git dalam proyek build yang nilainya ingin Anda ganti untuk build ini. Jika jenis sumber Anda adalah Amazon S3, nilai ini tidak didukung.
- *imageOverride*: Opsional. Nama gambar untuk build ini yang menggantikan yang ditentukan dalam proyek build.
- *idempotencyToken*: Opsional. String yang berfungsi sebagai token untuk menentukan bahwa permintaan build adalah idempotent. Anda dapat memilih string apa pun yang 64 karakter atau kurang. Token berlaku selama 5 menit setelah permintaan start-build. Jika Anda mengulangi permintaan start-build dengan token yang sama, tetapi mengubah parameter, CodeBuild mengembalikan kesalahan ketidakcocokan parameter.
- *insecureSslOverride*: Boolean opsional yang menentukan apakah akan mengganti pengaturan TLS tidak aman yang ditentukan dalam proyek build. Pengaturan TLS tidak aman menentukan apakah akan mengabaikan peringatan TLS saat menghubungkan ke kode sumber proyek. Penggantian ini hanya berlaku jika sumber build adalah GitHub Enterprise Server.
- *privilegedModeOverride*: Boolean opsional. Jika disetel ke true, build mengesampingkan mode istimewa dalam proyek build.
- *queuedTimeoutInMinutesOverride*: Bilangan bulat opsional yang menentukan jumlah menit build diizinkan untuk diantri sebelum waktunya habis. Nilai minimumnya adalah lima menit dan nilai maksimumnya adalah 480 menit (delapan jam).
- *reportBuildStatusOverride*: Boolean opsional yang menentukan apakah akan mengirimkan status awal dan penyelesaian build kepada penyedia sumber Anda. Jika Anda menyetel ini dengan penyedia sumber selain GitHub, GitHub Enterprise Server, atau Bitbucket, nilai yang tidak valid dilempar InputException kan.

- *sourceAuthOverride*: String opsional. Jenis otorisasi untuk build ini yang menggantikan yang ditentukan dalam proyek build. Penggantian ini hanya berlaku jika sumber proyek build adalah Bitbucket atau. GitHub
- *sourceLocationOverride*: String opsional. Lokasi yang mengganti untuk ini membangun lokasi sumber untuk yang ditentukan dalam proyek build.
- *serviceRoleOverride*: String opsional. Nama peran layanan untuk build ini yang menggantikan yang ditentukan dalam proyek build.
- *sourceTypeOverride*: String opsional. Jenis input sumber untuk build ini yang menggantikan input sumber yang ditentukan dalam proyek build. String yang valid adalah `NO_SOURCECODECOMMIT`,`CODEPIPELINE`,`GITHUB`,`S3`,,`BITBUCKET`, dan`GITHUB_ENTERPRISE`.
- *timeoutInMinutesOverride*: Nomor opsional. Jumlah menit batas waktu build yang menggantikan untuk build ini yang ditentukan dalam proyek build.

Sebaiknya simpan variabel lingkungan dengan nilai sensitif, seperti ID kunci AWS akses, kunci akses AWS rahasia, atau kata sandi sebagai parameter di Amazon EC2 Systems Manager Parameter Store. CodeBuild dapat menggunakan parameter yang disimpan di Amazon EC2 Systems Manager Parameter Store hanya jika nama parameter itu dimulai dengan `/CodeBuild/` (misalnya,`/CodeBuild/dockerLoginPassword`). Anda dapat menggunakan CodeBuild konsol untuk membuat parameter di Amazon EC2 Systems Manager. Pilih Buat parameter, lalu ikuti petunjuknya. (Dalam kotak dialog itu, untuk kunci KMS, Anda dapat secara opsional menentukan ARN AWS KMS kunci di akun Anda. Amazon EC2 Systems Manager menggunakan kunci ini untuk mengenkripsi nilai parameter selama penyimpanan dan mendekripsi selama pengambilan.) Jika Anda menggunakan CodeBuild konsol untuk membuat parameter, konsol memulai parameter dengan `/CodeBuild/` saat disimpan. Namun, jika Anda menggunakan konsol Amazon EC2 Systems Manager Parameter Store untuk membuat parameter, Anda harus memulai nama parameter dengan`/CodeBuild/`, dan Anda harus menyetel Type ke String A man. Untuk informasi selengkapnya, lihat penyimpanan AWS Systems Manager parameter dan Panduan: Membuat dan menguji parameter String (konsol) di Panduan Pengguna Amazon EC2 Systems Manager.

Jika proyek build mengacu pada parameter yang disimpan di Amazon EC2 Systems Manager Parameter Store, peran layanan proyek build harus mengizinkan `ssm:GetParameters` tindakan tersebut. Jika Anda memilih Buat peran layanan baru di akun Anda sebelumnya, ser CodeBuild takan tindakan ini dalam peran layanan default untuk proyek build Anda secara

otomatis. Namun, jika Anda memilih Pilih peran layanan yang ada dari akun Anda, maka Anda harus menyertakan tindakan ini dalam peran layanan secara terpisah.

Variabel lingkungan yang Anda tetapkan menggantikan variabel lingkungan yang ada. Misalnya, jika image Docker sudah berisi variabel lingkungan bernama `MY_VAR` dengan nilai`my_value`, dan Anda menetapkan variabel lingkungan bernama `MY_VAR` dengan nilai`other_value`, `my_value` maka diganti dengan`other_value`. Demikian pula, jika image Docker sudah berisi variabel lingkungan bernama `PATH` dengan nilai`/usr/local/sbin:/usr/local/bin`, dan Anda menetapkan variabel lingkungan bernama `PATH` dengan nilai`$PATH:/usr/share/ant/bin`, `/usr/local/sbin:/usr/local/bin` maka diganti dengan nilai `$PATH:/usr/share/ant/bin` literal.

Jangan mengatur variabel lingkungan apa pun dengan nama yang dimulai dengan`CODEBUILD_`. Awalan ini dicadangkan untuk penggunaan internal .

Jika variabel lingkungan dengan nama yang sama didefinisikan di beberapa tempat, nilai variabel lingkungan ditentukan sebagai berikut:

- Nilai dalam panggilan operasi mulai build memiliki prioritas tertinggi.
- Nilai dalam definisi proyek build akan diutamakan berikutnya.
- Nilai dalam deklarasi file buildspec memiliki prioritas terendah.

Untuk informasi tentang nilai yang valid untuk placeholder ini, lihat. Buat proyek build (AWS CLI) Untuk daftar pengaturan terbaru untuk proyek build, lihatLihat detail proyek build.

5. Beralih ke direktori yang berisi file yang baru saja Anda simpan, dan jalankan `start-build` perintah lagi.

```
aws codebuild start-build --cli-input-json file://start-build.json
```

6. Jika berhasil, data yang mirip dengan yang dijelaskan dalam Untuk menjalankan build prosedur muncul di output.

Untuk bekerja dengan informasi terperinci tentang build ini, catat `id` nilai dalam output, lalu lihatLihat detail build (AWS CLI).

Jalankan build batch (AWS CLI)

1. Jalan `start-build-batch` kan perintah dengan salah satu cara berikut:

   ```
   aws codebuild start-build-batch --project-name <project-name>
   ```

   Gunakan ini jika Anda ingin menjalankan build yang menggunakan versi terbaru dari artefak input build dan pengaturan proyek build yang ada.

   ```
   aws codebuild start-build-batch --generate-cli-skeleton > <json-file>
   ```

   Gunakan ini jika Anda ingin menjalankan build dengan versi sebelumnya dari artefak input build atau jika Anda ingin mengganti pengaturan untuk artefak keluaran build, variabel lingkungan, buildspec, atau periode batas waktu build default.
2. Jika Anda menjalankan start-build-batch perintah dengan `--project-name` opsi, ganti *<project-name>* dengan nama proyek build, dan kemudian lewati ke langkah 6 dari prosedur ini. Untuk mendapatkan daftar proyek pembangunan, lihatLihat nama proyek build.
3. Jika Anda menjalankan start-build-batch perintah dengan `--idempotency-token` opsi tersebut, pengenal unik yang peka huruf besar-kecil, atau token, disertakan dengan `start-build-batch` permintaan. Token ini berlaku selama 5 menit setelah permintaan. Jika Anda mengulangi `start-build-batch` permintaan dengan token yang sama, tetapi mengubah parameter, CodeBuild mengembalikan kesalahan ketidakcocokan parameter.
4. Jika Anda menjalankan start-build-batch perintah dengan `--generate-cli-skeleton` opsi, JSON-formatted data dikeluarkan ke *<json-file>* file. File ini mirip dengan kerangka yang dihasilkan oleh start-build perintah, dengan penambahan objek berikut. Untuk informasi lebih lanjut tentang objek umum, lihatJalankan build (AWS CLI).

   Ubah file ini untuk menambahkan penggantian build apa pun, dan simpan hasil Anda.

   ```
   "buildBatchConfigOverride": {
     "combineArtifacts": combineArtifacts,
     "restrictions": {
       "computeTypesAllowed": [
         allowedComputeTypes
       ],
       "maximumBuildsAllowed": maximumBuildsAllowed
     },
     "serviceRole": "batchServiceRole",
   ```

```
  "timeoutInMins": batchTimeout
}
```

Ob `buildBatchConfigOverride` jek adalah ProjectBuildBatchConfig struktur yang berisi penggantian konfigurasi build batch untuk build ini.

combineArtifacts

Sebuah boolean yang menentukan apakah artefak build untuk build batch harus digabungkan ke dalam satu lokasi artefak.

allowedComputeTypes

Array string yang menentukan jenis komputasi yang diizinkan untuk pembangunan batch. Lihat Mem bangun tipe komputasi lingkungan untuk nilai-nilai ini.

maximumBuildsAllowed

Menentukan jumlah maksimum pembangunan yang diizinkan.

batchServiceRole

Menentukan ARN peran layanan untuk proyek build batch.

batchTimeout

Menentukan jumlah waktu maksimum, dalam menit, bahwa build batch harus diselesaikan.

5. Beralih ke direktori yang berisi file yang baru saja Anda simpan, dan jalankan `start-build-batch` perintah lagi.

```
aws codebuild start-build-batch --cli-input-json file://start-build.json
```

6. Jika berhasil, representasi JSON dari BuildBatch objek muncul di output konsol. Lihat Sintaks StartBuildBatch Respon untuk contoh data ini.

Mulai menjalankan build secara otomatis ()AWS CLI

Jika kode sumber Anda disimpan dalam GitHub atau repositori Server GitHub Perusahaan, Anda dapat menggunakan GitHub webhook untuk AWS CodeBuild membangun kembali kode sumber Anda setiap kali perubahan kode didorong ke repositori.

Jalankan create-webhook perintah sebagai berikut:

```
aws codebuild create-webhook --project-name <project-name>
```

*<project-name>*adalah nama proyek build yang berisi kode sumber yang akan dibangun kembali.

Untuk GitHub, informasi yang mirip dengan berikut ini muncul di output:

```
{
  "webhook": {
    "url": "<url>"
  }
}
```

*<url>*adalah URL ke GitHub webhook.

Untuk GitHub Enterprise Server, informasi yang mirip dengan berikut ini muncul di output:



1. Salin kunci rahasia dan URL payload dari output. Anda membutuhkan mereka untuk menambahkan webhook di GitHub Enterprise Server.
2. Di GitHub Enterprise Server, pilih repositori tempat CodeBuild proyek Anda disimpan. Pilih Pengaturan, pilih Hooks & services, lalu pilih Add webhook.
3. Masukkan URL payload dan kunci rahasia, terima default untuk bidang lainnya, lalu pilih Tambahkan webhook.

Berhenti menjalankan build secara otomatis ()AWS CLI

Jika kode sumber Anda disimpan dalam GitHub atau repositori Server GitHub Perusahaan, Anda dapat mengatur GitHub webhook untuk AWS CodeBuild membangun kembali kode sumber Anda setiap kali perubahan kode didorong ke repositori. Untuk informasi selengkapnya, lihat Mulai menjalankan build secara otomatis ()AWS CLI.

Jika Anda telah mengaktifkan perilaku ini, Anda dapat mematikannya dengan menjalankan `delete-webhook` perintah sebagai berikut:

```
aws codebuild delete-webhook --project-name <project-name>
```

- di mana *<project-name>* adalah nama proyek build yang berisi kode sumber yang akan dibangun kembali.

Jika perintah ini berhasil, tidak ada informasi dan tidak ada kesalahan yang muncul di output.

> **Note**
>
> Ini menghapus webhook dari CodeBuild proyek Anda saja. Anda juga harus menghapus webhook dari repositori Server GitHub atau GitHub Enterprise Anda.

Jalankan build (AWS SDK)

Untuk menggunakan CodePipeline untuk menjalankan build dengan AWS CodeBuild, lewati langkah-langkah ini dan ikuti instruksi Gunakan AWS CodeBuild dengan AWS CodePipeline untuk menguji kode dan menjalankan build sebagai gantinya.

Untuk informasi tentang penggunaan CodeBuild dengan AWS SDK, lihat. AWS Referensi SDK dan alat

Jalankan dibangun di AWS Lambda komputasi

AWS Lambda compute menawarkan kecepatan start-up yang dioptimalkan untuk build Anda. AWS Lambda mendukung build yang lebih cepat karena latensi start-up yang lebih rendah. AWS Lambda juga secara otomatis menskalakan, jadi build tidak menunggu dalam antrian untuk dijalankan. Namun, ada beberapa kasus penggunaan yang AWS Lambda tidak mendukung, dan jika itu memengaruhi Anda, gunakan komputasi EC2. Untuk informasi selengkapnya, lihat Pembatasan AWS Lambda komputasi.

Topik

- Alat dan runtime mana yang akan disertakan dalam gambar docker lingkungan runtime yang dikuratori yang dijalankan AWS Lambda?
- Bagaimana jika gambar yang dikuratori tidak menyertakan alat yang saya butuhkan?

- Daerah mana yang mendukung AWS Lambda menghitung di? CodeBuild
- Pembatasan AWS Lambda komputasi
- Menyebarkan fungsi Lambda menggunakan AWS SAM dengan CodeBuild Lambda Java
- Buat aplikasi React satu halaman dengan CodeBuild Lambda Node.js
- Perbarui konfigurasi fungsi Lambda dengan CodeBuild Lambda Python

Alat dan runtime mana yang akan disertakan dalam gambar docker lingkungan runtime yang dikuratori yang dijalankan AWS Lambda?

AWS Lambda mendukung alat berikut: AWS CLI v2, AWS SAM CLI, git, go, Java,, Python Node.js, pip, Ruby, dan.NET.

Bagaimana jika gambar yang dikuratori tidak menyertakan alat yang saya butuhkan?

Jika gambar yang dikuratori tidak menyertakan alat yang Anda butuhkan, Anda dapat menyediakan image Docker lingkungan khusus yang menyertakan alat yang diperlukan.

> **Note**
>
> Lambda tidak mendukung fungsi yang menggunakan gambar kontainer multi-arsitektur. Untuk informasi selengkapnya, lihat Membuat fungsi Lambda menggunakan image wadah di Panduan Peng AWS Lambda embang.

Perhatikan bahwa Anda memerlukan izin Amazon ECR berikut untuk menggunakan gambar khusus untuk komputasi Lambda:

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ecr:GetAuthorizationToken"
```

```
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ecr:BatchCheckLayerAvailability",
                "ecr:GetDownloadUrlForLayer",
                "ecr:BatchGetImage"
            ],
            "Resource": "arn:aws:ecr:us-east-1:111122223333:repository/image-
repo"
        }
    ]
}
```

Perhatikan juga bahwa `curl` atau `wget` harus diinstal untuk menggunakan gambar khusus.

Daerah mana yang mendukung AWS Lambda menghitung di? CodeBuild

Di CodeBuild, AWS Lambda komputasi didukung dalam hal berikut Wilayah AWS: AS Timur (Virginia Utara), AS Timur (Ohio), AS Barat (Oregon), Asia Pasifik (Mumbai), Asia Pasifik (Singapura), Asia Pasifik (Sydney), Asia Pasifik (Tokyo), Eropa (Frankfurt), Eropa (Irlandia), dan Amerika Selatan (São Paulo). Untuk informasi selengkapnya tentang Wilayah AWS tempat CodeBuild tersedia, lihat AWS Layanan berdasarkan Wilayah.

Pembatasan AWS Lambda komputasi

Ada beberapa kasus penggunaan yang AWS Lambda tidak mendukung, dan jika berdampak pada Anda, gunakan komputasi EC2:

- AWS Lambda tidak mendukung alat yang memerlukan izin root. Untuk alat seperti `yum` atau`rpm`, gunakan tipe komputasi EC2 atau alat lain yang tidak memerlukan izin root.
- AWS Lambda tidak mendukung build atau menjalankan Docker.
- AWS Lambda tidak mendukung penulisan ke file di luar`/tmp`. Manajer paket yang disertakan dikonfigurasi untuk menggunakan `/tmp` direktori secara default untuk mengunduh dan mereferensikan paket.
- AWS Lambda tidak mendukung jenis lingkungan `LINUX_GPU_CONTAINER` dan tidak didukung pada Windows Server Core 2019.

- AWS Lambda tidak mendukung caching, batas waktu build khusus, batas waktu antrian, lencana build, mode istimewa, lingkungan runtime khusus, atau runtime lebih lama dari 15 menit.
- AWS Lambda tidak mendukung konektivitas VPC, rentang tetap alamat IP CodeBuild sumber, EFS, menginstal sertifikat, atau akses SSH dengan Manajer Sesi.

Menyebarkan fungsi Lambda menggunakan AWS SAM dengan CodeBuild Lambda Java

AWS Serverless Application Model (AWS SAM) adalah kerangka kerja open-source untuk membangun aplikasi tanpa server. Untuk informasi selengkapnya, lihat AWS Serverless Application Model repositori pada GitHub. Contoh Java berikut menggunakan Gradle untuk membangun dan menguji AWS Lambda fungsi. Setelah itu, AWS SAM CLI digunakan untuk menerapkan CloudFormation template dan bundel penerapan. Dengan menggunakan CodeBuild Lambda, langkah-langkah pembuatan, pengujian, dan penerapan semuanya ditangani secara otomatis, memungkinkan infrastruktur diperbarui dengan cepat tanpa intervensi manual dalam satu build.

Siapkan AWS SAM repositori

Buat AWS SAM `Hello World` proyek menggunakan AWS SAM CLI.

Untuk membuat AWS SAM Proyek

1. Ikuti petunjuk dalam Panduan Peng AWS Serverless Application Model embang untuk Meng instal AWS SAM CLI di mesin lokal Anda.
2. J `sam init` alankan dan pilih konfigurasi proyek berikut.

```
Which template source would you like to use?: 1 - AWS Quick Start Templates
Choose an AWS Quick Start application template: 1 - Hello World Example
Use the most popular runtime and package type? (Python and zip) [y/N]: N
Which runtime would you like to use?: 8 - java21
What package type would you like to use?: 1 - Zip
Which dependency manager would you like to use?: 1 - gradle
Would you like to enable X-Ray tracing on the function(s) in your application? [y/
N]: N
Would you like to enable monitoring using CloudWatch Application Insights? [y/N]: N
Would you like to set Structured Logging in JSON format on your Lambda functions?
 [y/N]:  N
Project name [sam-app]: <insert project name>
```

3. Unggah folder AWS SAM proyek ke repositori sumber yang didukung. Untuk daftar jenis sumber yang didukung, lihat ProjectSource.

Buat proyek CodeBuild Lambda Java

Buat proyek AWS CodeBuild Lambda Java dan atur izin IAM yang diperlukan untuk build.

Untuk membuat proyek CodeBuild Lambda Java

1. Buka AWS CodeBuild konsol di https://console.aws.amazon.com/codesuite/codebuild/home.
2. Jika halaman CodeBuild informasi ditampilkan, pilih Buat proyek build. Jika tidak, pada panel navigasi, per luas Bang un, pilih Bang un proyek, lalu pilih Buat proyek build.
3. Di Nama proyek, masukkan nama untuk proyek pembangunan ini. Nama proyek build harus unik di setiap AWS akun. Anda juga dapat menyertakan deskripsi opsional proyek build untuk membantu pengguna lain memahami untuk apa proyek ini digunakan.
4. Di Sumber, pilih repositori sumber tempat AWS SAM proyek Anda berada.
5. Di Lingkungan:
   - Untuk Compute, pilih Lambda.
   - Untuk Runtime (s), pilih Java.
   - Untuk Gambar, pilih aws/codebuild /amaz onlinux-x86_64-lambda-standard:corretto21.
   - Untuk Peran Layanan, biarkan Peran layanan baru dipilih. Catat nama Peran. Anda memerlukan ini saat memperbarui izin IAM proyek nanti dalam contoh ini.
6. Pilih Buat proyek pembangunan.
7. Buka konsol IAM di https://console.aws.amazon.com/iam/.
8. Di panel navigasi, pilih Per an dan pilih peran layanan yang terkait dengan proyek Anda. Anda dapat menemukan peran proyek Anda CodeBuild dengan memilih proyek build Anda, memilih Edit, Lingkungan, dan kemudian peran Layanan.
9. Pilih tab Relasi kepercayaan, lalu pilih Edit kebijakan kepercayaan.
10. Tambahkan kebijakan sebaris berikut ke peran IAM Anda. Ini akan digunakan untuk menyebarkan AWS SAM infrastruktur Anda nanti. Untuk informasi lebih lanjut, lihat Menambahkan dan menghapus izin identitas IAM dalam Panduan Pengguna IAM.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "",
            "Effect": "Allow",
            "Action": [
                "cloudformation:*",
                "lambda:*",
                "iam:*",
                "apigateway:*",
                "s3:*"
            ],
            "Resource": "arn:aws:iam::*:role/Service*"
        }
    ]
}
```

Siapkan proyek buildspec

Untuk membangun, menguji, dan menerapkan fungsi Lambda Anda, CodeBuild membaca dan mengeksekusi perintah build dari buildspec.

Untuk mengatur buildspec proyek Anda

1. Di CodeBuild konsol, pilih proyek build Anda, lalu pilih Edit dan Build spec.
2. Di Buildspec, pilih S isipkan perintah build dan kemudian Ber alih ke editor.
3. Hapus perintah build yang sudah diisi sebelumnya dan tempel di buildspec berikut.

```
version: 0.2
env:
  variables:
    GRADLE_DIR: "HelloWorldFunction"
phases:
  build:
    commands:
      - echo "Running unit tests..."
```

```
      - cd $GRADLE_DIR; gradle test; cd ..
      - echo "Running build..."
      - sam build --template-file template.yaml
      - echo "Running deploy..."
      - sam package --output-template-file packaged.yaml --resolve-s3 --template-
file template.yaml
      - yes | sam deploy
```

4. Pilih Per barui buildspec.

Menyebarkan milik Anda AWS SAM Infrastruktur Lambda

Gunakan CodeBuild Lambda untuk secara otomatis menerapkan infrastruktur Lambda Anda

Untuk menyebarkan infrastruktur Lambda

1. Pilih Mulai membangun. Ini akan secara otomatis membangun, menguji, dan menyebarkan AWS SAM aplikasi Anda untuk AWS Lambda menggunakan CloudFormation.
2. Setelah build selesai, navigasikan ke AWS Lambda konsol dan cari fungsi Lambda baru Anda di bawah nama AWS SAM proyek.
3. Uji fungsi Lambda Anda dengan memilih API Gateway di bawah I kh tisar Fungsi, lalu klik URL titik akhir API. Anda akan melihat halaman terbuka dengan pesan`"message": "hello world"`.

Bersihkan infrastruktur Anda

Untuk menghindari biaya lebih lanjut untuk sumber daya yang Anda gunakan selama tutorial ini, hapus sumber daya yang dibuat oleh AWS SAM template Anda dan CodeBuild.

Untuk membersihkan infrastruktur Anda

1. Arahkan ke CloudFormation konsol dan pilih`aws-sam-cli-managed-default`.
2. Di Resources, kosongkan bucket penerapan`SamCliSourceBucket`.
3. Hapus tum `aws-sam-cli-managed-default` pukan.
4. Hapus tum CloudFormation pukan yang terkait dengan AWS SAM proyek Anda. Tumpukan ini harus memiliki nama yang sama dengan AWS SAM proyek Anda.
5. Arahkan ke CloudWatch konsol dan hapus grup CloudWatch log yang terkait dengan CodeBuild proyek Anda.

6. Arahkan ke CodeBuild konsol dan hapus CodeBuild proyek Anda dengan memilih H apus proyek build.

Buat aplikasi React satu halaman dengan CodeBuild Lambda Node.js

Create React App adalah cara untuk membuat aplikasi React satu halaman. Node.js Contoh berikut digunakan Node.js untuk membangun artefak sumber dari Create React App dan mengembalikan artefak build.

Siapkan ember repositori dan artefak sumber Anda

Buat repositori sumber untuk proyek Anda menggunakan yarn dan Create React App.

Untuk mengatur repositori sumber dan ember artefak

1. Di mesin lokal Anda, jalankan `yarn create react-app <app-name>` untuk membuat aplikasi React sederhana.
2. Unggah folder proyek aplikasi React ke repositori sumber yang didukung. Untuk daftar jenis sumber yang didukung, lihat ProjectSource.

Buat proyek CodeBuild Lambda Node.js

Buat proyek AWS CodeBuild Lambda Node.js .

Untuk membuat proyek CodeBuild Lambda Node.js Anda

1. Buka AWS CodeBuild konsol di https://console.aws.amazon.com/codesuite/codebuild/home.
2. Jika halaman CodeBuild informasi ditampilkan, pilih Buat proyek build. Jika tidak, pada panel navigasi, perluas Build, pilih Build projects, lalu pilih Create build project.
3. Di Nama proyek, masukkan nama untuk proyek pembangunan ini. Membangun nama proyek harus unik di setiap AWS akun. Anda juga dapat menyertakan deskripsi opsional proyek build untuk membantu pengguna lain memahami tujuan proyek ini.
4. Di Sumber, pilih repositori sumber tempat AWS SAM proyek Anda berada.
5. Di Lingkungan:
   - Untuk Compute, pilih Lambda.
   - Untuk Runtime, pilih Node.js.

- Untuk Gambar, pilih aws/codebuild/amazonlinux-x86_64-lambda-standard:nodejs20.

6. Dalam Artefak:
   - Untuk Jenis, pilih Amazon S3.
   - Untuk nama Bucket, pilih bucket artefak proyek yang Anda buat sebelumnya.
   - Untuk kemasan Artefak, pilih Zip.
7. Pilih Buat proyek build.

Siapkan proyek buildspec

Untuk membangun aplikasi React Anda, CodeBuild membaca dan mengeksekusi perintah build dari file buildspec.

Untuk menyiapkan buildspec proyek Anda

1. Di CodeBuild konsol, pilih proyek build Anda, lalu pilih Edit dan Buildspec.
2. Di Buildspec, pilih Sisipkan perintah build dan kemudian Beralih ke editor.
3. Hapus perintah build yang telah diisi sebelumnya dan tempel di buildspec berikut.

```
version: 0.2
phases:
  build:
    commands:
      - yarn
      - yarn add --dev jest-junit @babel/plugin-proposal-private-property-in-object
      - yarn run build
      - yarn run test -- --coverage --watchAll=false --testResultsProcessor="jest-
junit" --detectOpenHandles
artifacts:
  name: "build-output"
  files:
    - "**/*"
reports:
  test-report:
    files:
      - 'junit.xml'
    file-format: 'JUNITXML'
  coverage-report:
    files:
      - 'coverage/clover.xml'
```

```
    file-format: 'CLOVERXML'
```

4. Pilih Perbarui buildspec.

Membangun dan menjalankan aplikasi React Anda

Bangun aplikasi React di CodeBuild Lambda, unduh artefak build, dan jalankan aplikasi React secara lokal.

Untuk membangun dan menjalankan aplikasi React Anda

1. Pilih Mulai membangun.
2. Setelah build selesai, navigasikan ke bucket artefak proyek Amazon S3 Anda dan unduh artefak aplikasi React.
3. Buka zip artefak build React dan `run npm install -g serve && serve -s build` di folder proyek.
4. `serve`Perintah akan melayani situs statis pada port lokal dan mencetak output ke terminal Anda. Anda dapat mengunjungi URL localhost `Local:` di bawah output terminal untuk melihat aplikasi React Anda.

Untuk mempelajari lebih lanjut tentang cara menangani deployment untuk server berbasis React, lihat Membuat Penerapan Aplikasi React.

Bersihkan infrastruktur Anda

Untuk menghindari biaya lebih lanjut untuk sumber daya yang Anda gunakan selama tutorial ini, hapus sumber daya yang dibuat untuk CodeBuild proyek Anda.

Untuk membersihkan infrastruktur Anda

1. Hapus artefak proyek Anda Amazon S3 bucket
2. Arahkan ke CloudWatch konsol dan hapus grup CloudWatch log yang terkait dengan CodeBuild proyek Anda.
3. Arahkan ke CodeBuild konsol dan hapus CodeBuild project Anda dengan memilih Delete build project.

Perbarui konfigurasi fungsi Lambda dengan CodeBuild Lambda Python

Contoh Python berikut menggunakan Boto3 dan CodeBuild Lambda Python untuk memperbarui konfigurasi fungsi Lambda. Sampel ini dapat diperluas untuk mengelola AWS sumber daya lain secara terprogram. Untuk informasi selengkapnya, lihat dokument asi Boto3.

Prasyarat

Buat atau temukan fungsi Lambda di akun Anda.

Contoh ini mengasumsikan bahwa Anda telah membuat fungsi Lambda di akun Anda dan akan digunakan CodeBuild untuk memperbarui variabel lingkungan fungsi Lambda. Untuk informasi lebih lanjut tentang pengaturan fungsi Lambda melalui CodeBuild, lihat Menyebarkan fungsi Lambda menggunakan AWS SAM dengan CodeBuild Lambda Java sampel atau kunjungi AWS Lambda.

Siapkan repositori sumber Anda

Buat repositori sumber untuk menyimpan skrip python Boto3 Anda.

Untuk mengatur repositori sumber

1. Salin skrip python berikut ke file baru yang disebut`update_lambda_environment_variables.py`.

```
import boto3
from os import environ


def update_lambda_env_variable(lambda_client):
    lambda_function_name = environ['LAMBDA_FUNC_NAME']
    lambda_env_variable = environ['LAMBDA_ENV_VARIABLE']
    lambda_env_variable_value = environ['LAMBDA_ENV_VARIABLE_VALUE']
    print("Updating lambda function " + lambda_function_name + " environment
 variable "
          + lambda_env_variable + " to " + lambda_env_variable_value)
    lambda_client.update_function_configuration(
        FunctionName=lambda_function_name,
        Environment={
            'Variables': {
                lambda_env_variable: lambda_env_variable_value
            }
```

```
        },
    )


if __name__ == "__main__":
    region = environ['AWS_REGION']
    client = boto3.client('lambda', region)
    update_lambda_env_variable(client)
```

2. Unggah file python ke repositori sumber yang didukung. Untuk daftar jenis sumber yang didukung, lihat ProjectSource.

Buat proyek CodeBuild Lambda Python

Buat proyek CodeBuild Lambda Python.

Untuk membuat proyek CodeBuild Lambda Java Anda

1. Buka AWS CodeBuild konsol di https://console.aws.amazon.com/codesuite/codebuild/home.
2. Jika halaman CodeBuild informasi ditampilkan, pilih Buat proyek build. Jika tidak, pada panel navigasi, per luas Bang un, pilih Bang un proyek, lalu pilih Buat proyek build.
3. Di Nama proyek, masukkan nama untuk proyek pembangunan ini. Nama proyek build harus unik di setiap AWS akun. Anda juga dapat menyertakan deskripsi opsional proyek build untuk membantu pengguna lain memahami untuk apa proyek ini digunakan.
4. Di Sumber, pilih repositori sumber tempat AWS SAM proyek Anda berada.
5. Di Lingkungan:
   - Untuk Compute, pilih Lambda.
   - Untuk Runtime, pilih Python.
   - Untuk Gambar, pilih aws/codebuild /amaz onlinux-x86_64-lambda-standard:python3.12.
   - Untuk Peran Layanan, biarkan Peran layanan baru dipilih. Catat nama Peran. Anda memerlukan ini saat memperbarui izin IAM proyek nanti dalam contoh ini.
6. Pilih Buat proyek pembangunan.
7. Buka konsol IAM di https://console.aws.amazon.com/iam/.
8. Di panel navigasi, pilih Per an dan pilih peran layanan yang terkait dengan proyek Anda. Anda dapat menemukan peran proyek Anda CodeBuild dengan memilih proyek build Anda, memilih Edit, Lingkungan, dan kemudian peran Layanan.

9. Pilih tab Relasi kepercayaan, lalu pilih Edit kebijakan kepercayaan.
10. Tambahkan kebijakan sebaris berikut ke peran IAM Anda. Ini akan digunakan untuk menyebarkan AWS SAM infrastruktur Anda nanti. Untuk informasi lebih lanjut, lihat Menambahkan dan menghapus izin identitas IAM dalam Panduan Pengguna IAM.

    JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "UpdateLambdaPermissions",
            "Effect": "Allow",
            "Action": [
                "lambda:UpdateFunctionConfiguration"
            ],
            "Resource": [
                "*"
            ]
        }
    ]
}
```

Siapkan proyek buildspec

Untuk memperbarui fungsi Lambda, skrip membaca variabel lingkungan dari buildspec untuk menemukan nama fungsi Lambda, nama variabel lingkungan, dan nilai variabel lingkungan.

Untuk mengatur buildspec proyek Anda

1. Di CodeBuild konsol, pilih proyek build Anda, lalu pilih Edit dan Build spec.
2. Di Buildspec, pilih S isipkan perintah build dan kemudian Ber alih ke editor.
3. Hapus perintah build yang sudah diisi sebelumnya dan tempel di buildspec berikut.

```
version: 0.2
env:
  variables:
    LAMBDA_FUNC_NAME: "<lambda-function-name>"
    LAMBDA_ENV_VARIABLE: "FEATURE_ENABLED"
```

```
    LAMBDA_ENV_VARIABLE_VALUE: "true"
phases:
  install:
    commands:
       - pip3 install boto3
  build:
    commands:
       - python3 update_lambda_environment_variables.py
```

4. Pilih Per barui buildspec.

Perbarui konfigurasi Lambda Anda

Gunakan CodeBuild Lambda Python untuk memperbarui konfigurasi fungsi Lambda Anda secara otomatis.

Untuk memperbarui konfigurasi fungsi Lambda Anda

1. Pilih Mulai membangun.
2. Setelah build selesai, navigasikan ke fungsi Lambda Anda.
3. Pilih Konfigurasi dan kemudian Vari abel lingkungan. Anda harus melihat variabel lingkungan baru dengan kunci `FEATURE_ENABLED` dan nilai`true`.

Bersihkan infrastruktur Anda

Untuk menghindari biaya lebih lanjut untuk sumber daya yang Anda gunakan selama tutorial ini, hapus sumber daya yang dibuat untuk CodeBuild proyek Anda.

Untuk membersihkan infrastruktur Anda

1. Arahkan ke CloudWatch konsol dan hapus grup CloudWatch log yang terkait dengan CodeBuild proyek Anda.
2. Arahkan ke CodeBuild konsol dan hapus CodeBuild proyek Anda dengan memilih H apus proyek build.
3. Jika Anda membuat fungsi Lambda untuk tujuan sampel ini, pilih fungsi Tindakan dan H apus untuk membersihkan fungsi Lambda Anda.

Ekstensi

Jika Anda ingin memperluas sampel ini untuk mengelola AWS sumber daya lain menggunakan AWS CodeBuild Lambda Python:

- Perbarui skrip Python untuk memodifikasi sumber daya baru menggunakan Boto3.
- Perbarui peran IAM yang terkait dengan CodeBuild proyek Anda untuk memiliki izin untuk sumber daya baru.
- Tambahkan variabel lingkungan baru yang terkait dengan sumber daya baru ke buildspec Anda.

Jalankan build pada armada kapasitas yang dicadangkan

CodeBuild menawarkan armada komputasi berikut:

- On-demand armada
- Armada kapasitas cadangan

Dengan armada sesuai permintaan, CodeBuild menyediakan komputasi untuk build Anda. Mesin-mesin hancur ketika pembangunan selesai. On-demand armada dikelola sepenuhnya, dan mencakup kemampuan penskalaan otomatis untuk menangani lonjakan permintaan.

> **Note**
>
> On-demand armada tidak mendukung macOS.

CodeBuild juga menawarkan armada kapasitas cadangan yang berisi instans yang didukung oleh Amazon EC2 yang dikelola oleh. CodeBuild Dengan armada kapasitas cadangan, Anda mengonfigurasi satu set instans khusus untuk lingkungan build Anda. Mesin-mesin ini tetap diam, siap untuk memproses build atau pengujian segera, dan mengurangi durasi pembuatan. Dengan armada kapasitas cadangan, mesin Anda selalu berjalan dan akan terus mengeluarkan biaya selama disediakan.

> **Important**
>
> Terlepas dari berapa lama Anda menjalankan instance, armada kapasitas cadangan dikenakan biaya awal per instans, setelah itu mungkin ada biaya terkait tambahan. Untuk informasi selengkapnya, lihat https://aws.amazon.com/codebuild/pricing/.

Topik

- Buat armada kapasitas yang dicadangkan
- Praktik terbaik
- Dapatkah saya berbagi armada kapasitas yang dipesan di beberapa CodeBuild proyek?
- Bagaimana cara kerja komputasi berbasis atribut?
- Dapatkah saya menentukan instans Amazon EC2 secara manual untuk armada saya?
- Wilayah mana yang mendukung armada kapasitas cadangan?
- Bagaimana cara mengonfigurasi armada macOS kapasitas cadangan?
- Bagaimana cara mengonfigurasi Image Mesin Amazon (AMI) khusus untuk armada kapasitas cadangan?
- Keterbatasan armada kapasitas cadangan
- Properti armada kapasitas cadangan
- Sampel kapasitas cadangan dengan AWS CodeBuild

Buat armada kapasitas yang dicadangkan

Gunakan petunjuk berikut untuk membuat armada kapasitas cadangan.

Untuk membuat armada kapasitas cadangan

1. Masuk ke Konsol Manajemen AWS dan buka AWS CodeBuild konsol di https://console.aws.amazon.com/codesuite/codebuild/home.
2. Di panel navigasi, pilih Ar mada komputasi, lalu pilih Buat Armada.
3. Di bidang teks Komputasi nama armada, masukkan nama untuk armada Anda.
4. Dari menu drop-down Sistem operasi, pilih sistem operasi.
5. Dari menu drop-down Arsitektur, pilih arsitektur.

6. (Opsional) Pilih Gunakan mode berjalan instans - opsional untuk dijalankan di instans Amazon EC2 secara langsung alih-alih wadah Docker. Kemudian pilih versi Major dan versi Minor.
7. (Opsional) Dalam konfigurasi tambahan lakukan hal berikut:

   - Pilih Konfigurasi VPC - opsional untuk menghubungkan armada Anda ke VPC untuk mengakses sumber daya pribadi selama penggunaan.
     - Dari menu drop-down VPC, pilih VPC yang akan diakses CodeBuild armada Anda.
     - Dari menu drop-down Subnets, pilih subnet yang CodeBuild harus digunakan untuk mengatur konfigurasi VPC Anda.
     - Dari menu tarik -turun Grup keamanan, pilih grup keamanan yang CodeBuild harus digunakan untuk bekerja dengan VPC Anda.
     - Di bidang Peran Layanan Ar mada, pilih peran layanan yang ada.

     > **Note**
     >
     > Pastikan peran armada Anda memiliki izin yang diperlukan. Untuk informasi selengkapnya, lihat Izinkan pengguna menambahkan kebijakan izin untuk peran layanan armada.

     - Jika Anda memilih sistem operasi Amazon Linux, pilih T entukan konfigurasi proxy - opsional untuk menerapkan kontrol akses jaringan untuk instans kapasitas cadangan Anda.
     - Untuk Perilaku default, pilih untuk mengizinkan atau menolak lalu lintas keluar ke semua tujuan secara default.
     - Untuk Aturan Proxy, pilih Tambahkan aturan proxy untuk menentukan domain tujuan atau IP untuk mengizinkan atau menolak kontrol akses jaringan.
   - Pilih Konfigurasi AMI kustom - opsional untuk menggunakan Image Mesin Amazon (AMI) kustom.
     - Dari menu tar ik-turun AMI, pilih Amazon Machine Image (AMI) untuk armada Anda.
     - Di bidang Peran Layanan Ar mada, pilih peran layanan yang ada.

     > **Note**
     >
     > Pastikan peran armada Anda memiliki izin yang diperlukan. Untuk informasi selengkapnya, lihat Izinkan pengguna menambahkan kebijakan izin untuk peran layanan armada.

8. Di Konfigur asi Kapasitas, dari mode pemilihan Compute, pilih salah satu dari berikut ini:

   - Jika Anda memilih Pilihan terpandu, lakukan hal berikut:
     - Untuk Compute, pilih jenis instans yang disertakan dalam armada ini.
     - Di bidang teks Kapasitas, masukkan jumlah minimum instance dalam armada.
     - (Opsional) Dalam konfigurasi tambahan lakukan hal berikut:
       - Pilih Konfigurasi penskalaan - opsional untuk menskalakan armada Anda secara otomatis berdasarkan konfigurasi ini. Dari mode pen skalaan - menu drop-down opsional, pilih perilaku ketika permintaan melebihi kapasitas armada.
   - Jika Anda memilih Instans kustom, lakukan hal berikut:
     - Dari menu drop-down tipe instance Compute, pilih jenis instans yang disertakan dalam armada ini.
     - Di bidang Ukuran volume EBS Tambahan - teks opsional, masukkan volume tambahan ke ruang disk 64GB yang disediakan.
     - Di bidang teks Kapasitas, masukkan jumlah minimum instance dalam armada.
     - (Opsional) Dalam konfigurasi tambahan lakukan hal berikut:
       - Pilih Konfigurasi penskalaan - opsional untuk menskalakan armada Anda secara otomatis berdasarkan konfigurasi ini. Dari mode pen skalaan - menu drop-down opsional, pilih perilaku ketika permintaan melebihi kapasitas armada.
9. Pilih Buat armada komputasi.
10. Setelah armada komputasi dibuat, buat CodeBuild proyek baru atau edit yang sudah ada. Dari Lingkungan, pilih Kapasitas Cadangan di bawah Model Penyediaan, lalu pilih armada yang ditentukan di bawah Nama Armada.

Praktik terbaik

Saat menggunakan armada kapasitas cadangan, kami sarankan Anda mengikuti praktik terbaik ini.

- Sebaiknya gunakan mode cache sumber untuk membantu meningkatkan kinerja build dengan menyimpan sumber dalam cache.
- Sebaiknya gunakan caching lapisan Docker untuk membantu meningkatkan kinerja build dengan men-cache lapisan Docker yang ada.

Dapatkah saya berbagi armada kapasitas yang dipesan di beberapa CodeBuild proyek?

Ya, Anda dapat memaksimalkan pemanfaatan kapasitas armada dengan menggunakannya di beberapa proyek.

> **Important**
>
> Saat menggunakan fitur kapasitas cadangan, data yang di-cache pada instance armada, termasuk file sumber, lapisan Docker, dan direktori cache yang ditentukan dalam buildspec, dapat diakses oleh proyek lain dalam akun yang sama. Ini berdasarkan desain dan memungkinkan proyek dalam akun yang sama untuk berbagi instance armada.

Bagaimana cara kerja komputasi berbasis atribut?

Jika Anda memilih `ATTRIBUTE_BASED_COMPUTE` sebagai armada Anda`computeType`, Anda dapat menentukan atribut di bidang baru yang disebut`computeConfiguration`. Atribut ini termasuk vCPU, memori, ruang disk, dan. `machineType` Ini `machineType` adalah salah satu `GENERAL` atau`NVME`. Setelah menentukan satu atau beberapa atribut yang tersedia, CodeBuild akan memilih tipe komputasi dari jenis instans yang didukung yang tersedia sebagai yang diselesaikan`computeConfiguration`.

> **Note**
>
> CodeBuild akan memilih instance termurah yang cocok dengan semua persyaratan input. Memori instans yang dipilih, vCPU, dan ruang disk semuanya akan lebih besar dari atau sama dengan persyaratan input. Anda dapat memeriksa yang diselesaikan `computeConfiguration` di armada yang dibuat atau diperbarui.

Jika Anda memasukkan a `computeConfiguration` yang tidak mungkin dipenuhi CodeBuild, Anda akan menerima pengecualian validasi. Perhatikan juga bahwa perilaku luapan armada sesuai permintaan akan diganti ke perilaku antrian jika tidak `computeConfiguration` tersedia untuk sesuai permintaan.

Dapatkah saya menentukan instans Amazon EC2 secara manual untuk armada saya?

Ya, Anda dapat langsung memasukkan instans Amazon EC2 yang Anda inginkan di konsol dengan memilih Instans khusus atau dengan mengonfigurasi parameter API,`InstanceType`. Bidang ini digunakan dalam API berikut: CreateFleet, UpdateFleet, CreateProject, UpdateProject dan StartBuild. Untuk informasi selengkapnya, lihat Compute instance type.

Wilayah mana yang mendukung armada kapasitas cadangan?

Kapasitas cadangan armada Amazon Linux dan Windows didukung di Wilayah AWS: AS Timur (Virginia Utara), AS Timur (Ohio), AS Barat (Oregon), Asia Pasifik (Mumbai), Asia Pasifik (Singapura), Asia Pasifik (Sydney), Asia Pasifik (Tokyo), Eropa (Frankfurt), Eropa (Irlandia), dan Amerika Selatan (São Paulo). Untuk informasi selengkapnya tentang Wilayah AWS tempat CodeBuild tersedia, lihat AWS Layanan berdasarkan Wilayah.

Armada macOS Medium kapasitas cadangan didukung di Wilayah AWS: AS Timur (Virginia Utara), AS Timur (Ohio), AS Barat (Oregon), Asia Pasifik (Sydney), dan Eropa (Frankfurt). Kapasitas cadangan macOS Armada besar didukung di Wilayah AWS: AS Timur (Virginia Utara), AS Timur (Ohio), AS Barat (Oregon), dan Asia Pasifik (Sydney).

Bagaimana cara mengonfigurasi armada macOS kapasitas cadangan?

Untuk mengonfigurasi armada macOS kapasitas cadangan

1. Masuk ke Konsol Manajemen AWS dan buka AWS CodeBuild konsol di https://console.aws.amazon.com/codesuite/codebuild/home.
2. Di panel navigasi, pilih Ar mada komputasi, lalu pilih Buat Armada.
3. Di bidang teks Komputasi nama armada, masukkan nama untuk armada Anda.
4. Dari menu drop-down Sistem operasi, pilih macOS.
5. Di bidang Komputasi, pilih salah satu jenis mesin komputasi berikut: Apple M2, memori 24 GB, 8 vCPU atau Apple M2, memori 32 GB, 12 vCPU.
6. Di bidang teks Kapasitas, masukkan jumlah minimum instance dalam armada.
7. (Opsional) Untuk menggunakan gambar khusus untuk armada Anda, lihat Bagaimana cara mengonfigurasi Image Mesin Amazon (AMI) khusus untuk armada kapasitas cadangan? untuk memastikan bahwa Amazon Machine Image (AMI) Anda memiliki prasyarat yang diperlukan.

8. (Opsional) Untuk mengonfigurasi VPC dengan armada Anda, di Konfigurasi tambahan lakukan hal berikut:

   - Dari menu drop-down VPC - opsional, pilih VPC yang akan diakses CodeBuild armada Anda.
   - Dari menu drop-down Subnets, pilih subnet yang CodeBuild harus digunakan untuk mengatur konfigurasi VPC Anda.
   - Dari menu tarik -turun Grup keamanan, pilih grup keamanan yang CodeBuild harus digunakan untuk bekerja dengan VPC Anda.
   - Di bidang Peran layanan Ar mada, pilih peran layanan yang ada.

   > **Note**
   >
   > Pastikan peran armada Anda memiliki izin yang diperlukan. Untuk informasi selengkapnya, lihat Izinkan pengguna menambahkan kebijakan izin untuk peran layanan armada.

9. Pilih Buat armada komput asi dan tunggu instans armada diluncurkan. Setelah diluncurkan kapasitasnya akan berada*n*/*n*, di *n* mana kapasitas yang disediakan.
10. Setelah armada komputasi diluncurkan, buat CodeBuild proyek baru atau edit yang sudah ada. Dari Lingkungan, pilih Kapasitas Cadangan di bawah Model Penyediaan, lalu pilih armada yang ditentukan di bawah Nama Armada.

Bagaimana cara mengonfigurasi Image Mesin Amazon (AMI) khusus untuk armada kapasitas cadangan?

Cara mengonfigurasi Image Mesin Amazon (AMI) khusus untuk armada kapasitas cadangan

1. Masuk ke Konsol Manajemen AWS dan buka AWS CodeBuild konsol di https://console.aws.amazon.com/codesuite/codebuild/home.
2. Di panel navigasi, pilih Ar mada komputasi, lalu pilih Buat Armada.
3. Di bidang teks Komputasi nama armada, masukkan nama untuk armada Anda.
4. Pilih Gambar kustom untuk armada Anda dan pastikan bahwa Amazon Machine Image (AMI) Anda memiliki prasyarat berikut:

   - Jika jenis lingkungan Anda adalah`MAC_ARM`, pastikan bahwa Arsit ektur AMI Anda adalah 64-bit`Mac-Arm`.

- Jika jenis lingkungan Anda adalah`LINUX_EC2`, pastikan bahwa Arsit ektur AMI Anda adalah 64-bit`x86`.
- Jika jenis lingkungan Anda adalah`ARM_EC2`, pastikan bahwa Arsit ektur AMI Anda adalah 64-bit`Arm`.
- Jika jenis lingkungan Anda adalah`WINDOWS_EC2`, pastikan bahwa Arsit ektur AMI Anda adalah 64-bit`x86`.
- AMI memungkinkan CodeBuild layanan Organisasi ARN. Untuk daftar ARN Organisasi, lihatAmazon Machine Images (AMI).
- Jika AMI dienkripsi dengan AWS KMS kunci, AWS KMS kunci tersebut juga harus mengizinkan ID Organisasi CodeBuild layanan. Untuk daftar ID Organisasi, lihatAmazon Machine Images (AMI). Untuk informasi selengkapnya tentang AWS KMS kunci, lihat Mengiz inkan organisasi dan OU menggunakan kunci KMS di Panduan Pengguna Amazon EC2. Untuk memberikan izin CodeBuild organisasi untuk menggunakan kunci KMS, tambahkan pernyataan berikut ke kebijakan kunci:

```
{
    "Sid": "Allow access for organization root",
    "Effect": "Allow",
    "Principal": "*",
    "Action": [
        "kms:Describe*",
        "kms:List*",
        "kms:Get*",
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
        "kms:CreateGrant"
    ],
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "aws:PrincipalOrgID": "o-123example"
        }
    }
}
```

- Di bidang peran layanan Ar mada, berikan izin Amazon EC2 berikut:

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
               "ec2:DescribeImages",
               "ec2:DescribeSnapshots"
            ],
            "Resource": "*"
        }
    ]
}
```

Keterbatasan armada kapasitas cadangan

Ada beberapa kasus penggunaan yang tidak didukung oleh armada kapasitas cadangan, dan jika hal itu berdampak pada Anda, gunakan armada sesuai permintaan sebagai gantinya:

- Armada kapasitas cadangan tidak mendukung metrik pemanfaatan build.
- Armada macOS kapasitas cadangan tidak mendukung sesi debug.

Untuk informasi lebih lanjut tentang batas dan kuota, lihatHitung armada.

Properti armada kapasitas cadangan

Armada kapasitas cadangan berisi properti berikut. Untuk informasi selengkapnya tentang armada kapasitas yang dipesan, lihatJalankan build pada armada kapasitas yang dicadangkan.

Sistem operasi

Sistem operasi. Sistem operasi berikut tersedia:

- Amazon Linux
- macOS
- Windows Server 2019
- Windows Server 2022

Arsitektur

Arsitektur prosesor. Arsitektur berikut tersedia:

- x86_64
- Lengan64

Tipe lingkungan

Jenis lingkungan yang tersedia saat Amazon Linux dipilih. Jenis lingkungan berikut tersedia:

- Linux EC2
- GPU Linux

Jenis instans komputasi

Konfigurasi komputasi untuk instance armada.

Pemilihan yang dipandu

Tentukan jenis komputasi yang berbeda dengan memilih pengaturan vCPU, memori dan ruang disk. Untuk informasi tentang ketersediaan tipe komputasi menurut wilayah, lihatTentang jenis lingkungan kapasitas yang dicadangkan.

Instans khusus

Tentukan jenis instance yang diinginkan secara manual.

Kapasitas

Jumlah awal mesin yang dialokasikan ke armada, yang menentukan jumlah build yang dapat berjalan secara paralel.

Perilaku meluap

Mendefinisikan perilaku ketika jumlah build melebihi kapasitas armada.

On-demand

Build overflow berjalan sesuai permintaan CodeBuild .

> **Note**
>
> Jika Anda memilih untuk menyetel perilaku overflow ke on-demand saat membuat VPC-connected armada, pastikan Anda menambahkan izin VPC yang diperlukan ke peran layanan proyek Anda. Untuk informasi selengkapnya, lihat Contoh pernyataan

kebijakan untuk mengiz CodeBuild inkan akses ke AWS layanan yang diperlukan untuk membuat antarmuka jaringan VPC.

> **Important**
>
> Jika Anda memilih untuk menyetel perilaku overflow ke on-demand, perhatikan bahwa build overflow akan ditagih secara terpisah, mirip dengan Amazon EC2 sesuai permintaan. Untuk informasi selengkapnya, lihat https://aws.amazon.com/codebuild/pricing/.

Antrian

Proses build ditempatkan dalam antrian sampai mesin tersedia. Ini membatasi biaya tambahan karena tidak ada mesin tambahan yang dialokasikan.

Gambar Mesin Amazon (AMI)

Properti Amazon Machine Image (AMI) untuk armada Anda. Properti berikut didukung oleh CodeBuild:

Wilayah AWS	Organisasi ARN	ID Organisasi
us-east-1	arn:aws:organizations::851725618577:organization/o-c6wcu152r1	o-c6wcu152r1
us-east-2	arn:aws:organizations::992382780434:organization/o-seufr2suvq	o-seufr2suvq
us-west-2	arn:aws:organizations::381491982620:organization/o-0412o99a4r	o-0412o99a4r

Wilayah AWS	Organisasi ARN	ID Organisasi
ap-northeast-1	arn:aws:organizations::891376993293:organization/o-b6k3sjqavm	o-b6k3sjqavm
ap-south-1	arn:aws:organizations::891376924779:organization/o-krtah1lkeg	o-krtah1lkeg
ap-southeast-1	arn:aws:organizations::654654522137:organization/o-mcn8uvc3tp	o-mcn8uvc3tp
ap-southeast-2	arn:aws:organizations::767398067170:organization/o-6crt0f6bu4	o-6crt0f6bu4
eu-central-1	arn:aws:organizations::590183817084:organization/o-lb2lne3te6	o-lb2lne3te6
eu-west-1	arn:aws:organizations::891376938588:organization/o-ullrrg5qf0	o-ullrrg5qf0
sa-east-1	arn:aws:organizations::533267309133:organization/o-db63c45ozw	o-db63c45ozw

Konfigurasi tambahan

VPC - opsional

VPC yang akan diakses CodeBuild armada Anda. Untuk informasi selengkapnya, lihat Gunakan AWS CodeBuild dengan Amazon Virtual Private Cloud.

> **Note**
>
> Jika override armada ditentukan saat memanggil StartBuild API, CodeBuild akan mengabaikan konfigurasi VPC proyek.

Subnet

Subnet VPC yang digunakan untuk CodeBuild mengatur konfigurasi VPC Anda. Perhatikan bahwa armada kapasitas cadangan hanya mendukung satu subnet dalam satu Zona Ketersediaan. Juga, pastikan bahwa subnet Anda menyertakan gateway NAT.

Grup keamanan

Grup keamanan VPC yang digunakan dengan CodeBuild VPC Anda. Pastikan grup keamanan Anda mengizinkan koneksi keluar.

Peran Layanan Armada

Mendefinisikan peran layanan untuk armada Anda dari peran layanan yang ada di akun Anda.

Tentukan konfigurasi proxy - opsional

Konfigurasi proxy yang menerapkan kontrol akses jaringan ke instans kapasitas cadangan Anda. Untuk informasi selengkapnya, lihat Gunakan AWS CodeBuild dengan server proxy terkelola.

> **Note**
>
> Konfigurasi proxy tidak mendukung VPC, Windows, atau macOS.

Perilaku default

Mendefinisikan perilaku lalu lintas keluar.

Izinkan

Memungkinkan lalu lintas keluar ke semua tujuan secara default.

Menolak

Menolak lalu lintas keluar ke semua tujuan secara default.

Aturan proxy

Menentukan domain tujuan atau IP untuk mengizinkan atau menolak kontrol akses jaringan.

Sampel kapasitas cadangan dengan AWS CodeBuild

Sampel ini dapat digunakan untuk bereksperimen dengan armada kapasitas cadangan di CodeBuild dalamnya.

Topik

- Caching dengan sampel kapasitas cadangan

Caching dengan sampel kapasitas cadangan

Cache dapat menyimpan bagian yang dapat digunakan kembali dari lingkungan build Anda dan menggunakannya di beberapa build. Contoh ini menunjukkan cara mengaktifkan caching dalam proyek build Anda menggunakan kapasitas cadangan. Untuk informasi selengkapnya, lihat Cache dibangun untuk meningkatkan kinerja.

Anda dapat memulai dengan menentukan satu atau lebih mode cache di pengaturan proyek Anda:

```
Cache:
        Type: LOCAL
        Modes:
          - LOCAL_CUSTOM_CACHE
          - LOCAL_DOCKER_LAYER_CACHE
          - LOCAL_SOURCE_CACHE
```

> **Note**
>
> Pastikan untuk mengaktifkan mode istimewa untuk menggunakan cache lapisan Docker.

Setelan buildspec proyek Anda akan terlihat seperti berikut:

```
version: 0.2
      phases:
        build:
          commands:
            - echo testing local source cache
            - touch /codebuild/cache/workspace/foobar.txt
            - git checkout -b cached_branch
            - echo testing local docker layer cache
            - docker run alpine:3.14 2>&1 | grep 'Pulling from' || exit 1
            - echo testing local custom cache
            - touch foo
            - mkdir bar && ln -s foo bar/foo2
            - mkdir bar/bar && touch bar/bar/foo3 && touch bar/bar/foo4
            - "[ -f foo ] || exit 1"
            - "[ -L bar/foo2 ] || exit 1"
            - "[ -f bar/bar/foo3 ] || exit 1"
            - "[ -f bar/bar/foo4 ] || exit 1"
      cache:
        paths:
           - './foo'
           - './bar/**/*'
           - './bar/bar/foo3'
```

Anda dapat memulai dengan menjalankan build dengan proyek baru untuk menyemai cache. Setelah selesai, Anda harus memulai build lain dengan buildspec yang mengesampingkan, mirip dengan yang berikut ini:

```
version: 0.2
      phases:
        build:
          commands:
            - echo testing local source cache
            - git branch | if grep 'cached_branch'; then (exit 0); else (exit 1); fi
            - ls /codebuild/cache/workspace | if grep 'foobar.txt'; then (exit 0); else
 (exit 1); fi
            - echo testing local docker layer cache
            - docker run alpine:3.14 2>&1 | if grep 'Pulling from'; then (exit 1); else
 (exit 0); fi
            - echo testing local custom cache
            - "[ -f foo ] || exit 1"
            - "[ -L bar/foo2 ] || exit 1"
```

```
            - "[ -f bar/bar/foo3 ] || exit 1"
            - "[ -f bar/bar/foo4 ] || exit 1"
      cache:
        paths:
           - './foo'
           - './bar/**/*'
           - './bar/bar/foo3'
```

Jalankan build dalam batch

Anda dapat menggunakan AWS CodeBuild untuk menjalankan build proyek secara bersamaan dan terkoordinasi dengan build batch.

Topik

- Peran keamanan
- Jenis build batch
- Mode laporan batch
- Informasi selengkapnya

Peran keamanan

Build batch memperkenalkan peran keamanan baru dalam konfigurasi batch. Peran baru ini diperlukan karena CodeBuild harus dapat memanggil`StartBuild`,`StopBuild`, dan `RetryBuild` tindakan atas nama Anda untuk menjalankan build sebagai bagian dari batch. Pelanggan harus menggunakan peran baru, dan bukan peran yang sama yang mereka gunakan dalam build mereka, karena dua alasan:

- Memberikan izin `StartBuild`, `StopBuild`, dan `RetryBuild` kepada peran build akan memungkinkan satu build untuk memulai lebih banyak build melalui buildspec.
- CodeBuild build batch memberikan batasan yang membatasi jumlah build dan tipe komputasi yang dapat digunakan untuk build dalam batch. Jika peran build memiliki izin ini, build dapat melewati batasan ini sendiri.

Jenis build batch

CodeBuild mendukung jenis build batch berikut:

Jenis build batch

- Bangun grafik
- Buat daftar
- Bangun matriks
- Bangun fanout

Bangun grafik

Grafik build mendefinisikan serangkaian tugas yang memiliki dependensi pada tugas lain dalam batch.

Contoh berikut mendefinisikan grafik build yang membuat rantai ketergantungan.

```
batch:
  fast-fail: false
  build-graph:
    - identifier: build1
      env:
        variables:
          BUILD_ID: build1
      ignore-failure: false
    - identifier: build2
      buildspec: build2.yml
      env:
        variables:
          BUILD_ID: build2
      depend-on:
        - build1
    - identifier: build3
      env:
        variables:
          BUILD_ID: build3
      depend-on:
        - build2
    - identifier: build4
      env:
        compute-type: ARM_LAMBDA_1GB
    - identifier: build5
      env:
        fleet: fleet_name
```

Dalam contoh ini:

- `build1`berjalan lebih dulu karena tidak memiliki dependensi.
- `build2`memiliki ketergantungan pada`build1`, jadi `build2` berjalan setelah `build1` selesai.
- `build3`memiliki ketergantungan pada`build2`, jadi `build3` berjalan setelah `build2` selesai.

Untuk informasi selengkapnya tentang sintaks buildspec build graph, lihat. batch/build-grafik

Buat daftar

Daftar build mendefinisikan sejumlah tugas yang berjalan secara paralel.

Contoh berikut mendefinisikan daftar build. Buil `build1` d `build2` dan akan berjalan secara paralel.

```
batch:
  fast-fail: false
  build-list:
    - identifier: build1
      env:
        variables:
          BUILD_ID: build1
      ignore-failure: false
    - identifier: build2
      buildspec: build2.yml
      env:
        variables:
          BUILD_ID: build2
      ignore-failure: true
    - identifier: build3
      env:
        compute-type: ARM_LAMBDA_1GB
    - identifier: build4
      env:
        fleet: fleet_name
    - identifier: build5
      env:
        compute-type: GENERAL_LINUX_XLAGRE
```

Untuk informasi selengkapnya tentang sintaks buildspec daftar build, lihat. batch/build-daftar

Bangun matriks

Matriks build mendefinisikan tugas dengan konfigurasi berbeda yang berjalan secara paralel. CodeBuild membuat build terpisah untuk setiap kombinasi konfigurasi yang mungkin.

Contoh berikut menunjukkan matriks build dengan dua file buildspec dan tiga nilai untuk variabel lingkungan.

```
batch:
  build-matrix:
    static:
      ignore-failure: false
    dynamic:
      buildspec:
        - matrix1.yml
        - matrix2.yml
      env:
        variables:
          MY_VAR:
            - VALUE1
            - VALUE2
            - VALUE3
```

Dalam contoh ini, CodeBuild buat enam build:

- `matrix1.yml` dengan `$MY_VAR=VALUE1`
- `matrix1.yml` dengan `$MY_VAR=VALUE2`
- `matrix1.yml` dengan `$MY_VAR=VALUE3`
- `matrix2.yml` dengan `$MY_VAR=VALUE1`
- `matrix2.yml` dengan `$MY_VAR=VALUE2`
- `matrix2.yml` dengan `$MY_VAR=VALUE3`

Setiap build akan memiliki pengaturan berikut:

- `ignore-failure`setel ke `false`
- `env/type`setel ke `LINUX_CONTAINER`
- `env/image`setel ke `aws/codebuild/amazonlinux-x86_64-standard:4.0`

- `env/privileged-mode`setel ke `true`

Build ini berjalan secara paralel.

Untuk informasi selengkapnya tentang sintaks buildspec matriks build, lihat. [batch/build-matriks]

Bangun fanout

Fanout build mendefinisikan tugas yang akan dibagi menjadi beberapa build dalam batch. Ini dapat digunakan untuk menjalankan tes secara paralel. CodeBuild membuat build terpisah untuk setiap pecahan kasus uji berdasarkan nilai yang ditetapkan di `parallelism` lapangan.

Contoh berikut mendefinisikan build fanout yang membuat lima build yang berjalan secara paralel.

```
version: 0.2

batch:
   fast-fail: false
   build-fanout:
     parallelism: 5
     ignore-failure: false

phases:
  install:
    commands:
      - npm install
   build:
    commands:
      - mkdir -p test-results
      - cd test-results
      - |
        codebuild-tests-run \
         --test-command 'npx jest --runInBand --coverage' \
         --files-search "codebuild-glob-search '**/test/**/*.test.js'" \
         --sharding-strategy 'equal-distribution'
```

Dalam contoh ini, dengan asumsi bahwa ada 100 tes yang perlu dijalankan, CodeBuild membuat lima build yang masing-masing menjalankan 20 pengujian secara paralel.

Untuk informasi selengkapnya tentang sintaks buildspec build graph, lihat. [batch/build-fanout]

Mode laporan batch

Jika penyedia sumber untuk proyek Anda adalah Bitbucket, GitHub, atau GitHub Enterprise, dan proyek Anda dikonfigurasi untuk melaporkan status build ke penyedia sumber, Anda dapat memilih bagaimana Anda ingin status build batch dikirim ke penyedia sumber. Anda dapat memilih agar status dikirim sebagai laporan status agregat tunggal untuk batch, atau memiliki status setiap build dalam batch yang dilaporkan secara individual.

Untuk informasi selengkapnya, lihat topik berikut:

- Konfigurasi batch (buat)
- Konfigurasi batch (pembaruan)

Informasi selengkapnya

Untuk informasi selengkapnya, lihat topik berikut:

- Referensi buildspec build batch
- Konfigurasi batch
- Jalankan build batch (AWS CLI)
- Hentikan pembuatan batch AWS CodeBuild

Jalankan tes paralel dalam build batch

Anda dapat menggunakan AWS CodeBuild untuk menjalankan pengujian paralel dalam build batch. Eksekusi pengujian paralel adalah pendekatan pengujian di mana beberapa kasus uji berjalan secara bersamaan di lingkungan, mesin, atau browser yang berbeda, daripada mengeksekusi secara berurutan. Pendekatan ini dapat secara signifikan mengurangi waktu eksekusi pengujian secara keseluruhan dan meningkatkan efisiensi pengujian. Di CodeBuild, Anda dapat membagi pengujian Anda di beberapa lingkungan dan menjalankannya secara bersamaan.

Keuntungan utama dari eksekusi uji paralel meliputi:

1. Mengurangi waktu eksekusi - Tes yang akan memakan waktu berjam-jam secara berurutan dapat diselesaikan dalam hitungan menit.
2. Pemanfaatan sumber daya yang lebih baik - Membuat penggunaan sumber daya komputasi yang tersedia secara efisien.

3. Umpan balik sebelumnya - Penyelesaian pengujian yang lebih cepat berarti umpan balik yang lebih cepat kepada pengembang.
4. Cost-effective- Menghemat waktu dan biaya komputasi dalam jangka panjang.

Saat menerapkan eksekusi uji paralel, dua pendekatan utama umumnya dipertimbangkan: lingkungan terpisah dan multithreading. Sementara kedua metode bertujuan untuk mencapai eksekusi pengujian bersamaan, mereka berbeda secara signifikan dalam implementasi dan efektivitasnya. Lingkungan terpisah membuat instance terisolasi di mana setiap rangkaian pengujian berjalan secara independen, sementara multithreading mengeksekusi beberapa pengujian secara bersamaan dalam ruang proses yang sama menggunakan utas yang berbeda.

Keuntungan utama dari lingkungan terpisah dibandingkan multithreading meliputi:

1. Isol asi - Setiap pengujian berjalan di lingkungan yang benar-benar terisolasi, mencegah gangguan antar pengujian.
2. Konflik sumber daya - Tidak ada persaingan untuk sumber daya bersama yang sering terjadi dalam multithreading.
3. Stabilitas - Kurang rentan terhadap kondisi balapan dan masalah sinkronisasi.
4. Debugging yang lebih mudah - Ketika pengujian gagal, lebih mudah untuk mengidentifikasi penyebabnya karena setiap lingkungan independen.
5. Manajemen negara - Mudah mengelola masalah status bersama yang mengganggu tes multithread.
6. Skalabilitas yang lebih baik - Dapat dengan mudah menambahkan lebih banyak lingkungan tanpa kerumitan.

Topik

- Dukungan di AWS CodeBuild
- Aktifkan eksekusi pengujian paralel dalam build batch
- Gunakan perintah codebuild- tests-run CLI
- Gunakan perintah codebuild- glob-search CLI
- Tentang pemisahan tes
- Gabungkan laporan build individual secara otomatis
- Eksekusi pengujian paralel untuk berbagai contoh kerangka kerja pengujian

Dukungan di AWS CodeBuild

AWS CodeBuild memberikan dukungan kuat untuk eksekusi pengujian paralel melalui fitur pembuatan batch, yang dirancang khusus untuk memanfaatkan eksekusi lingkungan yang terpisah. Implementasi ini selaras dengan manfaat lingkungan pengujian terisolasi.

Bangun batch dengan distribusi uji

CodeBuildfungsionalitas batch build memungkinkan pembuatan beberapa lingkungan build yang berjalan secara bersamaan. Setiap lingkungan beroperasi sebagai unit yang sepenuhnya terisolasi, dengan sumber daya komputasi sendiri, lingkungan runtime, dan dependensi. Melalui konfigurasi build batch, Anda dapat menentukan berapa banyak lingkungan paralel yang mereka butuhkan dan bagaimana pengujian harus didistribusikan di seluruh lingkungan tersebut.

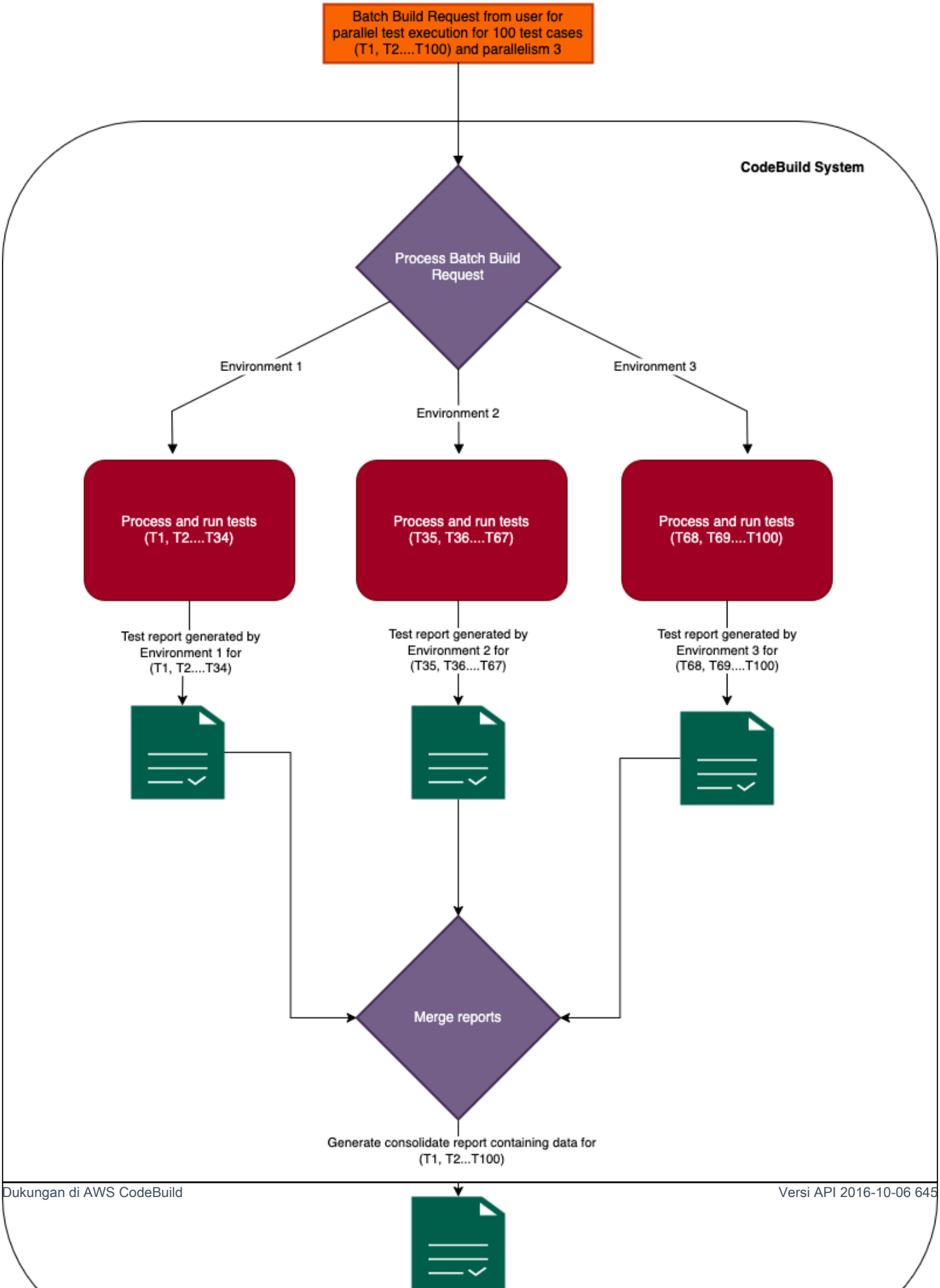
Uji sharding CLI

CodeBuild termasuk mekanisme distribusi pengujian bawaan melalui alat CLI-nya,`codebuild-tests-run`, yang secara otomatis membagi tes ke dalam lingkungan yang berbeda.

Agregasi laporan

Salah satu kekuatan utama implementasi CodeBuild adalah kemampuannya untuk menangani agregasi hasil tes dengan mulus. Sementara pengujian dijalankan di lingkungan terpisah, CodeBuild secara otomatis mengumpulkan dan menggabungkan laporan pengujian dari setiap lingkungan ke dalam laporan pengujian terpadu di tingkat build batch. Konsolidasi ini memberikan pandangan komprehensif tentang hasil pengujian sambil mempertahankan manfaat efisiensi eksekusi paralel.

Diagram berikut menjelaskan konsep lengkap eksekusi uji paralel di AWS CodeBuild.


Batch Build Request from user for parallel test execution for 100 test cases (T1, T2....T100) and parallelism 3
CodeBuild System
Process Batch Build Request
Environment 1
Environment 2
Environment 3
Process and run tests (T1, T2....T34)
Process and run tests (T35, T36....T67)
Process and run tests (T68, T69....T100)
Test report generated by Environment 1 for (T1, T2....T34)
Test report generated by Environment 2 for (T35, T36....T67)
Test report generated by Environment 3 for (T68, T69....T100)
Merge reports
Generate consolidate report containing data for (T1, T2...T100)

Aktifkan eksekusi pengujian paralel dalam build batch

Untuk menjalankan pengujian secara paralel, perbarui file buildspec build batch untuk menyertakan bidang build-fanout dan jumlah build paralel untuk membagi rangkaian pengujian di bidang seperti yang ditunjukkan di bawah ini. `parallelism` Bid `parallelism` ang menentukan berapa banyak eksekutor independen yang diatur untuk menjalankan rangkaian pengujian.

Untuk menjalankan pengujian di beberapa lingkungan eksekusi paralel, atur `parallelism` bidang ke nilai yang lebih besar dari nol. Dalam contoh di bawah ini, `parallelism` diatur ke lima, artinya CodeBuild memulai lima build identik yang menjalankan sebagian dari rangkaian pengujian secara paralel.

Anda dapat menggunakan perintah codebuild-tests-run CLI untuk membagi dan menjalankan pengujian Anda. File pengujian Anda akan dibagi, dan sebagian pengujian Anda dijalankan di setiap build. Ini mengurangi waktu keseluruhan yang dibutuhkan untuk menjalankan rangkaian pengujian lengkap. Dalam contoh berikut, tes akan dibagi menjadi lima dan titik pemisahan dihitung berdasarkan nama tes.

```
version: 0.2

batch:
  fast-fail: false
  build-fanout:
    parallelism: 5
    ignore-failure: false

phases:
  install:
    commands:
      - npm install jest-junit --save-dev
  pre_build:
    commands:
      - echo 'prebuild'
  build:
    commands:
      - |
        codebuild-tests-run \
         --test-command 'npx jest --runInBand --coverage' \
         --files-search "codebuild-glob-search '**/_tests_/**/*.test.js'" \
         --sharding-strategy 'equal-distribution'

  post_build:
```

```
    commands:
      - codebuild-glob-search '**/*.xml'
      - echo "Running post-build steps..."
      - echo "Build completed on `date`"

reports:
  test-reports:
    files:
      - '**/junit.xml'
    base-directory: .
    discard-paths: yes
    file-format: JUNITXML
```

Jika laporan dikonfigurasi untuk build-fanout build, maka laporan pengujian dibuat untuk setiap build secara terpisah, yang dapat dilihat di bawah tab Laporan dari build yang sesuai di konsol. AWS CodeBuild

Untuk informasi lebih lanjut tentang cara menjalankan pengujian paralel dalam batch, lihatEksekusi pengujian paralel untuk berbagai contoh kerangka kerja pengujian.

Gunakan perintah **`codebuild-`** tests-run CLI

AWS CodeBuild menyediakan CLI yang akan mengambil perintah pengujian dan lokasi file uji sebagai input. CLI dengan input ini akan membagi tes menjadi jumlah pecahan seperti yang ditentukan di `parallelism` bidang berdasarkan nama file uji. Penugasan file uji ke shard ditentukan oleh strategi sharding.

```
codebuild-tests-run \
    --files-search "codebuild-glob-search '**/__tests__/*.js'" \
    --test-command 'npx jest --runInBand --coverage' \
    --sharding-strategy 'equal-distribution'
```

Tabel berikut menjelaskan bidang untuk perintah `codebuild-tests-run` CLI.

Nama bidang	Tipe	Diperlukan atau opsional	Definisi
`test-command`	String	Diperlukan	Perintah ini digunakan untuk menjalankan tes.

Nama bidang	Tipe	Diperlukan atau opsional	Definisi
`files-search`	String	Diperlukan	Perintah ini memberikan daftar file uji. Anda dapat menggunakan perintah codebuild -glob-search CLI yang AWS CodeBuild disediakan atau alat pencarian file lain pilihan Anda. **Note** Pastikan bahwa `files-search` perintah mengeluarkan nama file, masing-masing dipisahkan oleh baris baru.

Nama bidang	Tipe	Diperlukan atau opsional	Definisi
`sharding-strategy`	Enum	Opsional	Nilai yang valid: `equal-distribution` (default), `stability` • `equal-distribution` : Pecahkan file uji secara merata berdasarkan nama file uji. • `stability` : File uji Shard menggunakan hashing yang konsisten dari nama file. Untuk informasi selengkapnya, lihat Tentang pemisahan tes.

`codebuild-tests-run`CLI bekerja terlebih dahulu untuk mengidentifikasi daftar file uji menggunakan perintah yang disediakan dalam `files-search` parameter. Kemudian menentukan subset file uji yang ditunjuk untuk pecahan saat ini (lingkungan) menggunakan strategi sharding yang ditentukan. Akhirnya, subset file uji ini diformat ke dalam daftar yang dipisahkan spasi dan ditambahkan ke akhir perintah yang disediakan dalam `test-command` parameter sebelum dieksekusi.

Untuk kerangka pengujian yang tidak menerima daftar yang dipisahkan spasi, `codebuild-tests-run` CLI menyediakan alternatif fleksibel melalui variabel `CODEBUILD_CURRENT_SHARD_FILES`

lingkungan. Variabel ini berisi daftar jalur file uji yang dipisahkan baris baru yang ditunjuk untuk pecahan build saat ini. Dengan memanfaatkan variabel lingkungan ini, Anda dapat dengan mudah beradaptasi dengan berbagai persyaratan kerangka pengujian, mengakomodasi mereka yang mengharapkan format input berbeda dari daftar yang dipisahkan spasi. Selain itu, Anda juga dapat memformat nama file pengujian sesuai kebutuhan kerangka pengujian. Berikut ini adalah contoh penggunaan `CODEBUILD_CURRENT_SHARD_FILES` di Linux dengan kerangka Django. Berikut `CODEBUILD_CURRENT_SHARD_FILES` ini digunakan untuk mendapatkan jalur file notasi titik yang didukung oleh Django:

```
codebuild-tests-run \
    –files-search "codebuild-glob-search '/tests/test_.py'" \
    –test-command 'python3 manage.py test $(echo "$CODEBUILD_CURRENT_SHARD_FILES" | sed
 -E "s/\//__/g; s/\.py$//; s/__/./g")' \
    –sharding-strategy 'equal-distribution'
```

> **Note**
>
> Perhatikan bahwa variabel `CODEBUILD_CURRENT_SHARD_FILES` lingkungan hanya dapat digunakan di dalam lingkup `codebuild-tests-run` CLI.
> Juga, jika Anda menggunakan perintah tes `CODEBUILD_CURRENT_SHARD_FILES` di dalam, masukkan tanda kutip ganda `CODEBUILD_CURRENT_SHARD_FILES` seperti yang ditunjukkan pada contoh di atas.

Gunakan perintah **`codebuild-`** glob-search CLI

AWS CodeBuild menyediakan alat CLI bawaan `codebuild-glob-search` yang disebut yang memungkinkan Anda untuk mencari file di direktori kerja Anda berdasarkan satu atau lebih pola glob. Alat ini dapat sangat berguna ketika Anda ingin menjalankan pengujian pada file atau direktori tertentu dalam proyek Anda.

Penggunaan

`codebuild-glob-search`CLI memiliki sintaks penggunaan berikut:

```
codebuild-glob-search <glob_pattern1> [<glob_pattern2> ...]
```

- *<glob_pattern1>*,*<glob_pattern2>*, dll.: Satu atau lebih pola glob untuk dicocokkan dengan file di direktori kerja Anda.
- *: Cocokkan urutan karakter apa pun (tidak termasuk pemisah jalur).
- **: Cocokkan urutan karakter apa pun (termasuk pemisah jalur).

> **Note**
>
> Pastikan bahwa string glob memiliki tanda kutip. Untuk memeriksa hasil pencocokan pola, gunakan perintah`echo`.
>
> ```
> version: 0.2
>
> phases:
>   build:
>     commands:
>       - echo $(codebuild-glob-search '**/__tests__/*.js')
>       - codebuild-glob-search '**/__tests__/*.js' | xargs -n 1 echo
> ```

Output

CLI akan menampilkan daftar jalur file yang dipisahkan baris baru yang cocok dengan pola glob yang disediakan. Jalur file yang dikembalikan akan relatif terhadap direktori kerja.

Jika tidak ada file yang ditemukan cocok dengan pola yang disediakan, CLI akan menampilkan pesan yang menunjukkan bahwa tidak ada file yang ditemukan.

Perhatikan bahwa direktori yang ditemukan karena pola tertentu akan dikecualikan dari hasil pencarian.

Contoh

Jika Anda ingin mencari hanya file di dalam direktori pengujian dan subdirektori dengan `.js` ekstensi, Anda dapat menggunakan perintah berikut dengan `codebuild-glob-search` CLI:

```
codebuild-glob-search '**/__tests__/*.js'
```

Perintah ini akan mencari semua file dengan `.js` ekstensi di dalam `__tests__` direktori dan subdirektorinya, seperti yang dilambangkan dengan pola.

Tentang pemisahan tes

AWS CodeBuild fitur pemisahan pengujian memungkinkan Anda untuk memparalelkan eksekusi rangkaian pengujian Anda di beberapa instance komputasi, mengurangi waktu uji coba secara keseluruhan. Fitur ini diaktifkan melalui konfigurasi batch di pengaturan CodeBuild proyek Anda dan `codebuild-tests-run` utilitas di file buildspec Anda.

Pengujian dibagi berdasarkan strategi sharding yang ditentukan. CodeBuild menyediakan dua strategi sharding seperti yang ditentukan di bawah ini:

Equal-distribution

Strategi `equal-distribution` sharding membagi pengujian di seluruh build paralel berdasarkan urutan abjad dari nama file pengujian. Pendekatan ini pertama-tama mengurutkan file pengujian dan kemudian menggunakan metode berbasis potongan untuk mendistribusikannya, memastikan bahwa file serupa dikelompokkan bersama untuk pengujian. Disarankan ketika berhadapan dengan satu set file uji yang relatif kecil. Meskipun metode ini bertujuan untuk mengalokasikan jumlah file yang kira-kira sama untuk setiap pecahan, dengan perbedaan maksimum satu, itu tidak menjamin stabilitas. Ketika file uji ditambahkan atau dihapus di build berikutnya, distribusi file yang ada dapat berubah, berpotensi menyebabkan penugasan ulang di seluruh pecahan.

Stabilitas

Strategi `stability` sharding menggunakan algoritma hashing yang konsisten untuk membagi pengujian di antara pecahan, memastikan bahwa distribusi file tetap stabil. Ketika file baru ditambahkan atau dihapus, pendekatan ini memastikan bahwa penugasan file-to-shard yang ada sebagian besar tetap tidak berubah. Untuk rangkaian pengujian besar, disarankan untuk menggunakan opsi stabilitas untuk mendistribusikan tes secara merata di seluruh pecahan. Mekanisme ini bertujuan untuk memberikan distribusi yang hampir sama, memastikan bahwa setiap pecahan menerima jumlah file yang sama, dengan hanya varians minimal. Meskipun strategi stabilitas tidak menjamin distribusi setara yang ideal, strategi ini menawarkan distribusi yang hampir sama yang mempertahankan konsistensi dalam penetapan file di seluruh build, bahkan saat file ditambahkan atau dihapus.

Untuk mengaktifkan pemisahan pengujian, Anda perlu mengonfigurasi bagian batch dalam pengaturan CodeBuild proyek Anda, menentukan `parallelism` level yang diinginkan dan parameter relevan lainnya. Selain itu, Anda harus menyertakan `codebuild-tests-run` utilitas dalam file buildspec Anda, bersama dengan perintah pengujian dan metode pemisahan yang sesuai.

Gabungkan laporan build individual secara otomatis

Dalam pembuatan batch fanout, AWS CodeBuild mendukung penggabungan otomatis laporan build individual ke dalam laporan tingkat batch konsolidasi. Fitur ini memberikan tampilan komprehensif hasil pengujian dan cakupan kode di semua build dalam satu batch.

Cara kerjanya

Saat menjalankan `fanout` batch build, setiap build individu menghasilkan laporan pengujian. CodeBuild kemudian secara otomatis mengkonsolidasikan laporan identik dari build yang berbeda ke dalam laporan terpadu, yang dilampirkan ke build batch. Laporan konsolidasi ini mudah diakses melalui `reportArns` bidang BatchGetBuildBatches API, dan juga dapat dilihat di tab Laporan di konsol. Kemampuan penggabungan ini meluas ke laporan yang ditemukan secara otomatis juga.

Laporan konsolidasi dibuat di bawah grup laporan yang ditentukan dalam spesifikasi build atau ditemukan secara otomatis oleh. CodeBuild Anda dapat menganalisis tren laporan gabungan langsung di bawah grup laporan ini, memberikan wawasan berharga tentang kinerja build secara keseluruhan dan metrik kualitas di seluruh build historis dari proyek batch build yang sama.

Untuk setiap build individu dalam batch, CodeBuild secara otomatis membuat grup laporan terpisah. Ini mengikuti konvensi penamaan tertentu, menggabungkan nama grup laporan build batch dengan akhiran`BuildFanoutShard<shard_number>`, di mana `shard_number` mewakili jumlah pecahan tempat grup laporan dibuat. Organisasi ini memungkinkan Anda untuk melacak dan menganalisis tren di tingkat pembangunan konsolidasi dan individu, memberikan fleksibilitas dalam cara Anda memantau dan mengevaluasi proses pembangunan mereka.

Laporan batch build mengikuti struktur yang sama dengan laporan build individual. Bidang kunci berikut di tab Laporan khusus untuk laporan pembuatan batch:

Status laporan pembuatan batch

Status laporan build batch mengikuti aturan tertentu tergantung pada jenis laporan:

- Laporan uji:
  - Berhasil: Status disetel menjadi berhasil ketika semua laporan build individual telah berhasil.
  - Gagal: Status disetel ke gagal jika ada laporan build individual yang gagal.
  - Tidak Lengkap: Status ditandai sebagai tidak lengkap jika ada laporan build individu yang hilang atau memiliki status tidak lengkap.
- Laporan cakupan kode:

- Selesai: Status diatur untuk selesai ketika semua laporan build individual selesai.
- Gagal: Status disetel ke gagal jika ada laporan build individual yang gagal.
- Tidak Lengkap: Status ditandai sebagai tidak lengkap jika ada laporan build individu yang hilang atau memiliki status tidak lengkap.

Ringkasan tes

Laporan pengujian gabungan menggabungkan bidang berikut dari semua laporan build individual:

- durasi-dalam-nano-detik: Waktu durasi pengujian maksimum dalam nanodetik di antara semua laporan pembuatan individu.
- total: Jumlah gabungan dari semua kasus uji, menjumlahkan jumlah total pengujian dari setiap build.
- jumlah status: Menyediakan tampilan konsolidasi status pengujian seperti lulus, gagal, atau dilewati, dihitung dengan menggabungkan jumlah setiap jenis status di semua build individual.

Ringkasan cakupan kode

Laporan cakupan kode gabungan menggabungkan bidang dari semua build individual menggunakan perhitungan berikut:

- cabang yang dicakup: Jumlah semua cabang yang dicakup dari laporan individu.
- cabang-terlewatkan: Jumlah semua cabang yang terlewat dari laporan individu.
- persentase cakupan cabang: `(Total covered branches / Total branches) * 100`
- garis-tercakup: Jumlah semua baris yang dicakup dari laporan individu.
- baris terlewatkan: Jumlah semua baris yang terlewat dari laporan individu.
- persentase pertanggungan garis: `(Total covered lines / Total lines) * 100`

ID eksekusi

Batch membangun ARN.

Kasus uji

Laporan gabungan berisi daftar gabungan dari semua kasus uji dari build individual, dapat diakses melalui DescribeTestCases API dan laporan build batch di konsol.

Pertanggungan kode

Laporan cakupan kode gabungan menyediakan informasi cakupan baris dan cabang gabungan untuk setiap file di semua build individual, dapat diakses melalui DescribeCodeCoverages API

dan laporan build batch di konsol. Catatan: Untuk file yang dicakup oleh beberapa file pengujian yang didistribusikan di pecahan yang berbeda, laporan gabungan menggunakan kriteria pemilihan berikut:

1. Seleksi utama didasarkan pada cakupan garis tertinggi di antara pecahan.
2. Jika cakupan garis sama di beberapa pecahan, pecahan dengan cakupan cabang tertinggi dipilih.

Eksekusi pengujian paralel untuk berbagai contoh kerangka kerja pengujian

Anda dapat menggunakan perintah `codebuild-tests-run` CLI untuk membagi dan menjalankan pengujian Anda di lingkungan eksekusi paralel. Bagian berikut memberikan `buildspec.yml` contoh untuk berbagai kerangka kerja, menggambarkan penggunaan `codebuild-tests-run` perintah.

- Setiap contoh di bawah ini mencakup `parallelism` level lima, yang berarti bahwa lima lingkungan eksekusi yang identik akan dibuat untuk membagi pengujian Anda. Anda dapat memilih `parallelism` level yang sesuai dengan proyek Anda dengan memodifikasi `parallelism` nilai di `build-fanout` bagian tersebut.
- Setiap contoh di bawah ini menunjukkan konfigurasi pengujian Anda untuk dibagi dengan nama file pengujian, yang merupakan default. Ini mendistribusikan tes secara merata di seluruh lingkungan eksekusi paralel.

Sebelum Anda memulai, lihat Jalankan tes paralel dalam build batch untuk informasi lebih lanjut.

Untuk daftar lengkap opsi saat menggunakan perintah `codebuild-tests-run` CLI, lihatGunakan perintah `codebuild-` tests-run CLI.

Topik

- Konfigurasikan tes paralel dengan Django
- Konfigurasikan tes paralel dengan Elixir
- Konfigurasikan tes paralel dengan Go
- Konfigurasikan tes paralel dengan Java (Maven)
- Konfigurasikan tes paralel dengan Javascript (Jest)
- Konfigurasikan tes paralel dengan Kotlin
- Konfigurasikan tes paralel dengan PHPUnit

- Konfigurasikan tes paralel dengan Pytest
- Konfigurasikan tes paralel dengan Ruby (Cucumber)
- Konfigurasikan tes paralel dengan Ruby (RSpec)

Konfigurasikan tes paralel dengan Django

Berikut ini adalah contoh `buildspec.yml` yang menunjukkan eksekusi uji paralel dengan Django pada platform Ubuntu:

```
version: 0.2

batch:
  fast-fail: false
  build-fanout:
    parallelism: 5

phases:
  install:
    commands:
      - echo 'Installing Python dependencies'
      - sudo yum install -y python3 python3-pip
      - python3 -m ensurepip --upgrade
      - python3 -m pip install django
  pre_build:
    commands:
      - echo 'Prebuild'
  build:
    commands:
      - echo 'Running Django Tests'
      - |
        codebuild-tests-run \
         --test-command 'python3 manage.py test $(echo "$CODEBUILD_CURRENT_SHARD_FILES"
 | sed -E "s/\//__/g; s/\.py$//; s/__/./g")' \
         --files-search "codebuild-glob-search '**/tests/*test_*.py'" \
         --sharding-strategy 'equal-distribution'
  post_build:
    commands:
      - echo 'Test execution completed'
```

Contoh di atas menunjukkan penggunaan variabel lingkunganCODEBUILD_CURRENT_SHARD_FILES. Di sini CODEBUILD_CURRENT_SHARD_FILES

digunakan untuk mengambil jalur file notasi titik yang didukung oleh Django. Gunakan tanda `CODEBUILD_CURRENT_SHARD_FILES` kutip ganda di dalam seperti yang ditunjukkan di atas.

Konfigurasikan tes paralel dengan Elixir

Berikut ini adalah contoh `buildspec.yml` yang menunjukkan eksekusi uji paralel dengan Elixir pada platform Ubuntu:

```
version: 0.2

batch:
  fast-fail: false
  build-fanout:
    parallelism: 5

phases:
  install:
    commands:
      - echo 'Installing Elixir dependencies'
      - sudo apt update
      - sudo DEBIAN_FRONTEND=noninteractive apt install -y elixir
      - elixir --version
      - mix --version
  pre_build:
    commands:
      - echo 'Prebuild'
  build:
    commands:
      - echo 'Running Elixir Tests'
      - |
        codebuild-tests-run \
         --test-command 'mix test' \
         --files-search "codebuild-glob-search '**/test/**/*_test.exs'" \
         --sharding-strategy 'equal-distribution'
  post_build:
    commands:
      - echo "Test execution completed"
```

Konfigurasikan tes paralel dengan Go

Berikut ini adalah contoh `buildspec.yml` yang menunjukkan eksekusi uji paralel dengan Go pada platform Linux:

```
version: 0.2

batch:
  fast-fail: false
  build-fanout:
    parallelism: 5
    ignore-failure: false

phases:
  install:
    commands:
      - echo 'Fetching Go version'
      - go version
  pre_build:
    commands:
      - echo 'prebuild'
  build:
    commands:
      - echo 'Running go Tests'
      - go mod init calculator
      - cd calc
      - |
        codebuild-tests-run \
         --test-command "go test -v calculator.go" \
         --files-search "codebuild-glob-search '**/*test.go'"
  post_build:
    commands:
      - echo "Test execution completed"
```

Dalam contoh di atas, `calculator.go` fungsi berisi fungsi matematika sederhana untuk menguji dan semua file uji dan `calculator.go` file ada di dalam `calc` folder.

Konfigurasikan tes paralel dengan Java (Maven)

Berikut ini adalah contoh `buildspec.yml` yang menunjukkan eksekusi uji paralel dengan Java pada platform Linux:

```
version: 0.2

batch:
  fast-fail: false
  build-fanout:
```

```
    parallelism: 5
    ignore-failure: false

phases:
  pre_build:
    commands:
      - echo 'prebuild'
  build:
    commands:
      - echo "Running mvn test"
      - |
        codebuild-tests-run \
          --test-command 'mvn test -Dtest=$(echo "$CODEBUILD_CURRENT_SHARD_FILES" | sed
 "s|src/test/java/||g; s/\.java//g; s|/|.|g; s/ /,/g" | tr "\n" "," | sed "s/,$//")' \
          --files-search "codebuild-glob-search '**/test/**/*.java'"

  post_build:
    commands:
      - echo "Running post-build steps..."
      - echo "Test execution completed"
```

Dalam contoh yang diberikan, variabel lingkungan `CODEBUILD_CURRENT_SHARD_FILES` berisi file uji di pecahan saat ini, dipisahkan oleh baris baru. File-file ini diubah menjadi daftar nama kelas yang dipisahkan koma dalam format yang diterima oleh `-Dtest` parameter untuk Maven.

Konfigurasikan tes paralel dengan Javascript (Jest)

Berikut ini adalah contoh `buildspec.yml` yang menunjukkan eksekusi uji paralel dengan Javascript pada platform Ubuntu:

```
version: 0.2

batch:
  fast-fail: true
  build-fanout:
    parallelism: 5
    ignore-failure: false

phases:
  install:
    commands:
      - echo 'Installing Node.js dependencies'
      - apt-get update
```

```
      - apt-get install -y nodejs
      - npm install
      - npm install --save-dev jest-junit
  pre_build:
    commands:
      - echo 'prebuild'
  build:
    commands:
      - echo 'Running JavaScript Tests'
      - |
         codebuild-tests-run \
          --test-command "npx jest" \
          --files-search "codebuild-glob-search '**/test/**/*.test.js'" \
          --sharding-strategy 'stability'
    post_build:
      commands:
        - echo 'Test execution completed'
```

Konfigurasikan tes paralel dengan Kotlin

Berikut ini adalah contoh `buildspec.yml` yang menunjukkan eksekusi uji paralel dengan Kotlin pada platform Linux:

```
version: 0.2

batch:
  fast-fail: false
  build-fanout:
    parallelism: 2
    ignore-failure: false

phases:
  install:
    runtime-versions:
      java: corretto11
    commands:
      - echo 'Installing dependencies'
      - KOTLIN_VERSION="1.8.20" # Replace with your desired version
      - curl -o kotlin-compiler.zip -L "https://github.com/JetBrains/kotlin/releases/
download/v${KOTLIN_VERSION}/kotlin-compiler-${KOTLIN_VERSION}.zip"
      - unzip kotlin-compiler.zip -d /usr/local
      - export PATH=$PATH:/usr/local/kotlinc/bin
      - kotlin -version
```

```
      - curl -O https://repo1.maven.org/maven2/org/junit/platform/junit-platform-
console-standalone/1.8.2/junit-platform-console-standalone-1.8.2.jar
  pre_build:
    commands:
      - echo 'prebuild'
  build:
    commands:
      - echo 'Running Kotlin Tests'
      - |
        codebuild-tests-run \
          --test-command 'kotlinc src/main/kotlin/*.kt $(echo
 "$CODEBUILD_CURRENT_SHARD_FILES" | tr "\n" " ") -d classes -cp junit-platform-console-
standalone-1.8.2.jar' \
          --files-search "codebuild-glob-search 'src/test/kotlin/*.kt'"
      - |
        codebuild-tests-run \
          --test-command '
            java -jar junit-platform-console-standalone-1.8.2.jar --class-path classes
 \
              $(for file in $CODEBUILD_CURRENT_SHARD_FILES; do
                 class_name=$(basename "$file" .kt)
                 echo "--select-class $class_name"
               done)
          ' \
          --files-search "codebuild-glob-search 'src/test/kotlin/*.kt'"
  post_build:
    commands:
      - echo "Test execution completed"
```

Dalam contoh di atas, `codebuild-tests-run` CLI digunakan dua kali. Selama proses pertama, kotlinc mengkompilasi file. `CODEBUILD_CURRENT_SHARD_FILES`Variabel mengambil file uji yang ditetapkan ke shard saat ini, yang kemudian diubah menjadi daftar yang dipisahkan spasi. Pada putaran kedua, JUnit mengeksekusi tes. Sekali `CODEBUILD_CURRENT_SHARD_FILES` lagi, mengambil file pengujian yang ditetapkan ke shard saat ini, tetapi kali ini mereka diubah menjadi nama kelas.

Konfigurasikan tes paralel dengan PHPUnit

Berikut ini adalah contoh `buildspec.yml` yang menunjukkan eksekusi uji paralel dengan PHPUnit pada platform Linux:

```
version: 0.2
```

```
batch:
   fast-fail: false
   build-fanout:
     parallelism: 5
     ignore-failure: false

phases:
   install:
     commands:
       - echo 'Install dependencies'
       - composer require --dev phpunit/phpunit
   pre_build:
     commands:
       - echo 'prebuild'
   build:
     commands:
       - echo 'Running phpunit Tests'
       - composer dump-autoload
       - |
         codebuild-tests-run \
          --test-command "./vendor/bin/phpunit --debug" \
          --files-search "codebuild-glob-search '**/tests/*Test.php'"
   post_build:
       commands:
         - echo 'Test execution completed'
```

Konfigurasikan tes paralel dengan Pytest

Berikut ini adalah contoh `buildspec.yml` yang menunjukkan eksekusi uji paralel dengan Pytest pada platform Ubuntu:

```
version: 0.2

batch:
  fast-fail: false
  build-fanout:
    parallelism: 5
    ignore-failure: false

phases:
  install:
    commands:
```

```
      - echo 'Installing Python dependencies'
      - apt-get update
      - apt-get install -y python3 python3-pip
      - pip3 install --upgrade pip
      - pip3 install pytest
  build:
    commands:
      - echo 'Running Python Tests'
      - |
         codebuild-tests-run \
          --test-command 'python -m pytest' \
          --files-search "codebuild-glob-search 'tests/test_*.py'" \
          --sharding-strategy 'equal-distribution'
  post_build:
    commands:
      - echo "Test execution completed"
```

Berikut ini adalah contoh `buildspec.yml` yang menunjukkan eksekusi uji paralel dengan Pytest pada platform Windows:

```
version: 0.2

batch:
  fast-fail: false
  build-fanout:
    parallelism: 5
    ignore-failure: false

phases:
  install:
    commands:
      - echo 'Installing Python dependencies'
      - pip install pytest
  pre_build:
    commands:
      - echo 'prebuild'
  build:
    commands:
      - echo 'Running pytest'
      - |
        & codebuild-tests-run `
         --test-command 'pytest @("$env:CODEBUILD_CURRENT_SHARD_FILES" -split \"`r?`n
\")'  `
```

```
        --files-search "codebuild-glob-search '**/test_*.py' '**/*_test.py'" `
        --sharding-strategy 'equal-distribution'
  post_build:
    commands:
      - echo "Test execution completed"
```

Dalam contoh di atas, variabel `CODEBUILD_CURRENT_SHARD_FILES` lingkungan digunakan untuk mengambil file uji yang ditetapkan ke shard saat ini dan diteruskan sebagai array ke perintah pytest.

Konfigurasikan tes paralel dengan Ruby (Cucumber)

Berikut ini adalah contoh `buildspec.yml` yang menunjukkan eksekusi uji paralel dengan Cucumber pada platform Linux:

```
version: 0.2

batch:
  fast-fail: false
  build-fanout:
    parallelism: 5
    ignore-failure: false

phases:
  install:
    commands:
      - echo 'Installing Ruby dependencies'
      - gem install bundler
      - bundle install
  pre_build:
    commands:
      - echo 'prebuild'
  build:
    commands:
      - echo 'Running Cucumber Tests'
      - cucumber --init
      - |
        codebuild-tests-run \
         --test-command "cucumber" \
         --files-search "codebuild-glob-search '**/*.feature'"
  post_build:
    commands:
      - echo "Test execution completed"
```

Konfigurasikan tes paralel dengan Ruby (RSpec)

Berikut ini adalah contoh `buildspec.yml` yang menunjukkan eksekusi uji paralel dengan RSpec pada platform Ubuntu:

```
version: 0.2

batch:
  fast-fail: false
  build-fanout:
    parallelism: 5
    ignore-failure: false

phases:
  install:
    commands:
      - echo 'Installing Ruby dependencies'
      - apt-get update
      - apt-get install -y ruby ruby-dev build-essential
      - gem install bundler
      - bundle install
  build:
    commands:
      - echo 'Running Ruby Tests'
      - |
         codebuild-tests-run \
          --test-command 'bundle exec rspec' \
          --files-search "codebuild-glob-search 'spec/**/*_spec.rb'" \
          --sharding-strategy 'equal-distribution'
  post_build:
    commands:
      - echo "Test execution completed"
```

Cache dibangun untuk meningkatkan kinerja

Anda dapat menghemat waktu saat proyek Anda dibangun dengan menggunakan cache. Cache dapat menyimpan bagian yang dapat digunakan kembali dari lingkungan build Anda dan menggunakannya di beberapa build. Proyek build Anda dapat menggunakan salah satu dari dua jenis caching: Amazon S3 atau lokal. Jika Anda menggunakan cache lokal, Anda harus memilih satu atau lebih dari tiga mode cache: cache sumber, cache lapisan Docker, dan cache khusus.

> **Note**
>
> Mode cache lapisan Docker hanya tersedia untuk lingkungan Linux. Jika Anda memilih mode ini, Anda harus menjalankan build Anda dalam mode istimewa. CodeBuild proyek yang diberikan mode istimewa memberikan akses kontainernya ke semua perangkat. Untuk informasi selengkapnya, lihat Hak istimewa Runtime dan kemampuan Linux di situs Docker Docker.

Topik

- Penyimpanan cache Amazon S3
- Caching lokal
- Tentukan cache lokal

Penyimpanan cache Amazon S3

Caching Amazon S3 menyimpan cache dalam bucket Amazon S3 yang tersedia di beberapa host build. Ini adalah pilihan yang baik untuk artefak bangunan berukuran kecil hingga menengah yang lebih mahal untuk dibangun daripada untuk diunduh.

Untuk menggunakan Amazon S3 dalam build, Anda dapat menentukan jalur untuk file yang ingin Anda simpan dalam cache`buildspec.yml`. CodeBuild akan secara otomatis menyimpan dan memperbarui cache ke lokasi Amazon S3 yang dikonfigurasi pada proyek. Jika Anda tidak menentukan jalur file, upaya terbaik CodeBuild akan menyimpan dependensi bahasa umum untuk membantu Anda mempercepat build. Anda dapat melihat detail cache di log build.

Selain itu, jika Anda ingin memiliki beberapa versi cache, Anda dapat menentukan kunci cache di`buildspec.yml`. CodeBuild menyimpan cache di bawah konteks kunci cache ini, dan membuat salinan cache unik yang tidak akan diperbarui setelah dibuat. Kunci cache dapat dibagikan di seluruh proyek juga. Fitur seperti kunci dinamis, versi cache, dan berbagi cache di seluruh build hanya tersedia ketika kunci ditentukan.

Untuk mempelajari lebih lanjut tentang sintaks cache dalam file buildspec, lihat cache di referensi buildspec.

Topik

- Hasilkan kunci dinamis

- kodebuild-hash-file
- Versi cache
- Berbagi cache antar proyek
- Contoh Buildspec

Hasilkan kunci dinamis

Kunci cache dapat menyertakan perintah shell dan variabel lingkungan untuk membuatnya unik, memungkinkan pembaruan cache otomatis ketika kunci berubah. Misalnya, Anda dapat menentukan kunci menggunakan hash `package-lock.json` file. Ketika dependensi dalam file itu berubah, hash — dan karenanya kunci cache — berubah, memicu pembuatan cache baru secara otomatis.

```
cache:
    key: npm-key-$(codebuild-hash-files package-lock.json)
```

CodeBuild akan mengevaluasi ekspresi `$(codebuild-hash-files package-lock.json)` untuk mendapatkan kunci akhir:

```
npm-key-abc123
```

Anda juga dapat menentukan kunci cache menggunakan variabel lingkungan, seperti`CODEBUILD_RESOLVED_SOURCE_VERSION`. Ini memastikan bahwa setiap kali sumber Anda berubah, kunci baru dihasilkan, menghasilkan cache baru yang disimpan secara otomatis:

```
cache:
   key: npm-key-$CODEBUILD_RESOLVED_SOURCE_VERSION
```

CodeBuild akan mengevaluasi ekspresi dan mendapatkan kunci akhir:

```
npm-key-046e8b67481d53bdc86c3f6affdd5d1afae6d369
```

kodebuild-hash-file

`codebuild-hash-files`adalah alat CLI yang menghitung SHA-256 hash untuk satu set file di direktori CodeBuild sumber menggunakan pola glob:

```
codebuild-hash-files <glob-pattern-1> <glob-pattern-2> ...
```

Berikut adalah beberapa contoh menggunakan`codebuild-hash-files`:

```
codebuild-hash-files package-lock.json
codebuild-hash-files '**/*.md'
```

Versi cache

Versi cache adalah hash yang dihasilkan dari jalur direktori yang di-cache. Jika dua cache memiliki versi yang berbeda, mereka diperlakukan sebagai cache yang berbeda selama proses pencocokan. Misalnya, dua cache berikut dianggap berbeda karena mereferensikan jalur yang berbeda:

```
version: 0.2

phases:
  build:
    commands:
      - pip install pandas==2.2.3 --target pip-dependencies
cache:
  key: pip-dependencies
  paths:
    - "pip-dependencies/**/*"
```

```
version: 0.2

phases:
  build:
    commands:
      - pip install pandas==2.2.3 --target tmp/pip-dependencies
cache:
  key: pip-dependencies
  paths:
    - "tmp/pip-dependencies/**/*"
```

Berbagi cache antar proyek

Anda dapat menggunakan bidang `cacheNamespace` API di bawah `cache` bagian untuk berbagi cache di beberapa proyek. Bidang ini mendefinisikan ruang lingkup cache. Untuk berbagi cache, Anda harus melakukan hal berikut:

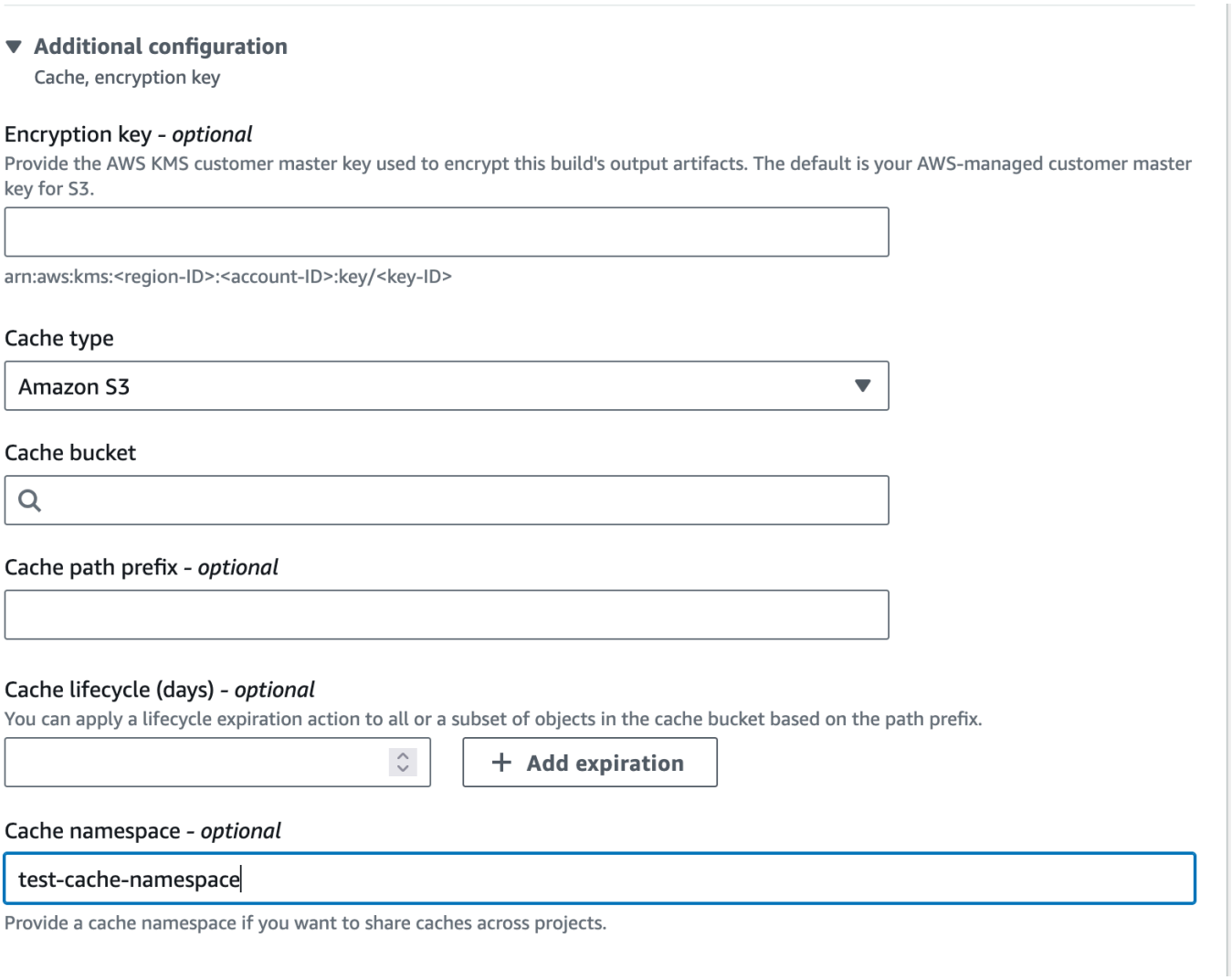
- Gunakan yang sama`cacheNamespace`.
- Tentukan cache yang sama`key`.

- Tentukan jalur cache yang identik.
- Gunakan bucket Amazon S3 yang sama dan `pathPrefix` jika disetel.

Ini memastikan konsistensi dan memungkinkan berbagi cache di seluruh proyek.

Tentukan namespace cache (konsol)

1. Buka AWS CodeBuild konsol di https://console.aws.amazon.com/codesuite/codebuild/home.
2. Pilih Buat proyek. Untuk informasi selengkapnya, lihat Buat proyek build (konsol) dan Jalankan build (konsol).
3. Di Artefak, pilih Konfigurasi tambahan.
4. Untuk jenis Cache, pilih Amazon S3.
5. Untuk Cache namespace - opsional, masukkan nilai namespace.



6. Lanjutkan dengan nilai default dan kemudian pilih Buat proyek build.

Tentukan namespace cache (AWS CLI)

Anda dapat menggunakan `--cache` parameter di AWS CLI untuk menentukan namespace cache.

```
--cache '{"type": "S3", "location": "your-s3-bucket", "cacheNamespace": "test-cache-namespace"}'
```

Contoh Buildspec

Berikut adalah beberapa contoh buildspec untuk bahasa umum:

Topik

- Keter Node.js gantungan cache
- Dependensi Cache Python
- Dependensi Cache Ruby
- Dependensi Cache Go

Keter Node.js gantungan cache

Jika proyek Anda menyertakan `package-lock.json` file dan digunakan `npm` untuk mengelola Node.js dependensi, contoh berikut menunjukkan cara mengatur caching. Secara default, `npm` menginstal dependensi ke `node_modules` direktori.

```
version: 0.2

phases:
  build:
    commands:
      - npm install
cache:
  key: npm-$(codebuild-hash-files package-lock.json)
  paths:
    - "node_modules/**/*"
```

Dependensi Cache Python

Jika proyek Anda menyertakan `requirements.txt` file dan menggunakan pip untuk mengelola dependensi Python, contoh berikut menunjukkan cara mengkonfigurasi caching. Secara default, pip menginstal paket ke `site-packages` direktori sistem.

```
version: 0.2

phases:
  build:
    commands:
      - pip install -r requirements.txt
cache:
  key: python-$(codebuild-hash-files requirements.txt)
  paths:
    - "/root/.pyenv/versions/${python_version}/lib/python${python_major_version}/site-
packages/**/*"
```

Selain itu, Anda dapat menginstal dependensi ke direktori tertentu dan mengkonfigurasi caching untuk direktori itu.

```
version: 0.2

phases:
  build:
    commands:
      - pip install -r requirements.txt --target python-dependencies
cache:
  key: python-$(codebuild-hash-files requirements.txt)
  paths:
    - "python-dependencies/**/*"
```

Dependensi Cache Ruby

Jika proyek Anda menyertakan `Gemfile.lock` file dan digunakan `Bundler` untuk mengelola dependensi permata, contoh berikut menunjukkan cara mengkonfigurasi caching secara efektif.

```
version: 0.2

phases:
  build:
    commands:
```

```
      - bundle install --path vendor/bundle
cache:
  key: ruby-$(codebuild-hash-files Gemfile.lock)
  paths:
    - "vendor/bundle/**/*"
```

Dependensi Cache Go

Jika proyek Anda menyertakan `go.sum` file dan menggunakan modul Go untuk mengelola dependensi, contoh berikut menunjukkan cara mengkonfigurasi caching. Secara default, modul Go diunduh dan disimpan di `${GOPATH}/pkg/mod` direktori.

```
version: 0.2

phases:
  build:
    commands:
      - go mod download
cache:
  key: go-$(codebuild-hash-files go.sum)
  paths:
    - "/go/pkg/mod/**/*"
```

Caching lokal

Caching lokal menyimpan cache secara lokal pada host build yang tersedia untuk host build itu saja. Ini adalah pilihan yang baik untuk artefak build menengah hingga besar karena cache segera tersedia di host build. Ini bukan pilihan terbaik jika build Anda jarang. Ini berarti bahwa kinerja build tidak terpengaruh oleh waktu transfer jaringan.

Jika Anda memilih cache lokal, Anda harus memilih satu atau lebih mode cache berikut:

- Mode cache sumber menyimpan metadata Git untuk sumber primer dan sekunder. Setelah cache dibuat, pembangunan berikutnya menarik perubahan antar pelaksanaan saja. Mode ini merupakan pilihan yang baik untuk proyek dengan direktori yang bersih dan berfungsi serta sumber yang merupakan repositori Git besar. Jika Anda memilih opsi ini dan proyek Anda tidak menggunakan repositori Git (AWS CodeCommit, GitHub, GitHub Enterprise Server, atau Bitbucket), opsi tersebut diabaikan.
- Mode cache lapisan Docker menyimpan lapisan Docker yang ada. Mode ini merupakan pilihan yang baik untuk proyek yang membangun atau menarik gambar Docker yang besar. Ia dapat

mencegah masalah performa yang disebabkan oleh penarikan keluar gambar Docker yang besar dari jaringan.

> **Note**
>
> - Anda dapat menggunakan cache lapisan Docker di lingkungan Linux saja.
> - Bendera `privileged` harus diatur agar proyek Anda memiliki izin Docker yang diperlukan.
>
> Secara default, daemon Docker diaktifkan untuk build non-VPC. Jika Anda ingin menggunakan wadah Docker untuk build VPC, lihat Runtime Privilege dan Kapabilitas Linux di situs Docker Docker dan aktifkan mode istimewa. Selain itu, Windows tidak mendukung mode istimewa.
> - Anda harus mempertimbangkan implikasi keamanan sebelum Anda menggunakan cache lapisan Docker.

- Mode cache khusus menyimpan direktori yang Anda tentukan dalam file buildspec. Mode ini adalah pilihan yang baik jika skenario build Anda tidak cocok untuk salah satu dari dua mode cache lokal lainnya. Jika Anda menggunakan cache kustom:
  - Hanya direktori yang dapat ditentukan untuk caching. Anda tidak dapat menentukan file individu.
  - Symlink digunakan untuk mereferensikan direktori cache.
  - Direktori yang di-cache ditautkan ke pembangunan Anda sebelum mengunduh sumber proyeknya. Item yang di-cache menggantikan item sumber jika memiliki nama yang sama. Direktori ditentukan menggunakan jalur cache di file buildspec. Untuk informasi selengkapnya, lihat Sintaks Buildspec.
  - Hindari nama direktori yang sama di sumber dan di cache. Locally-cached direktori dapat mengganti, atau menghapus konten, direktori di repositori sumber yang memiliki nama yang sama.

> **Note**
>
> Caching lokal tidak didukung dengan tipe `LINUX_GPU_CONTAINER` lingkungan dan tipe `BUILD_GENERAL1_2XLARGE` komputasi. Untuk informasi selengkapnya, lihat Membangun mode dan tipe komputasi lingkungan.

> **Note**
>
> Caching lokal tidak didukung saat Anda mengonfigurasi CodeBuild untuk bekerja dengan VPC. Untuk informasi lebih lanjut tentang menggunakan VPC dengan CodeBuild, lihatGunakan AWS CodeBuild dengan Amazon Virtual Private Cloud.

Tentukan cache lokal

Anda dapat menggunakan AWS CLI, konsol, SDK, atau CloudFormation untuk menentukan cache lokal. Untuk informasi selengkapnya tentang caching lokal, lihatCaching lokal.

Topik

- Tentukan caching lokal (CLI)
- Tentukan caching lokal (konsol)
- Tentukan caching lokal (CloudFormation)

Tentukan caching lokal (CLI)

Anda dapat menggunakan `--cache` parameter di AWS CLI untuk menentukan masing-masing dari tiga jenis cache lokal.

- Untuk menentukan cache sumber:

```
--cache type=LOCAL,mode=[LOCAL_SOURCE_CACHE]
```

- Untuk menentukan cache lapisan Docker:

```
--cache type=LOCAL,mode=[LOCAL_DOCKER_LAYER_CACHE]
```

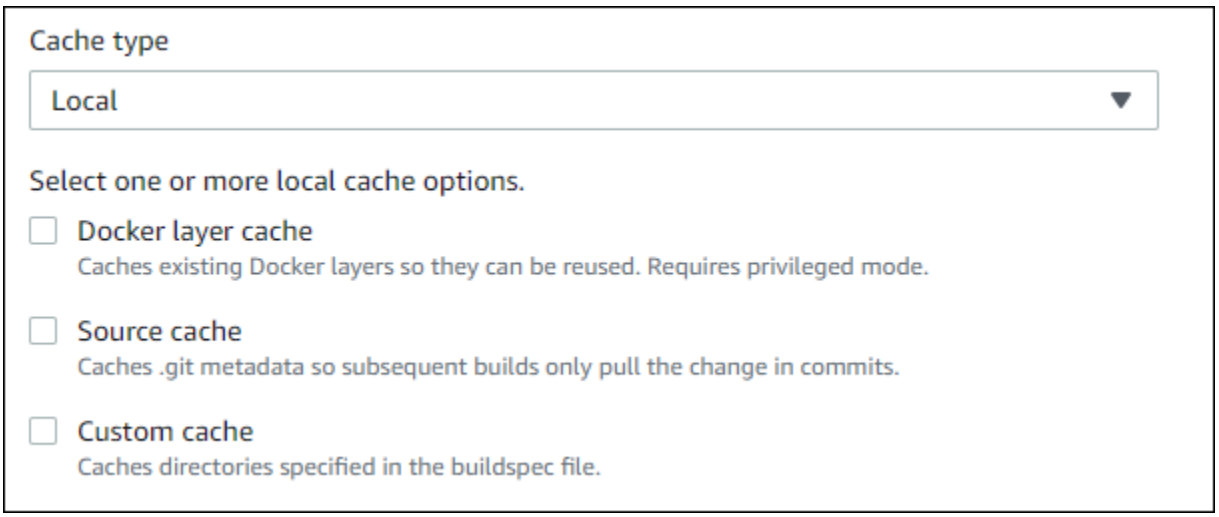
- Untuk menentukan cache khusus:

```
--cache type=LOCAL,mode=[LOCAL_CUSTOM_CACHE]
```

Untuk informasi selengkapnya, lihat Buat proyek build (AWS CLI).

Tentukan caching lokal (konsol)

Anda menentukan cache di bagian Artifacts dari konsol. Untuk jenis Cache, pilih Amazon S3 atau Lokal. Jika Anda memilih Lokal, pilih satu atau lebih dari tiga opsi cache lokal.



Untuk informasi selengkapnya, lihat Buat proyek build (konsol).

Tentukan caching lokal (CloudFormation)

Jika Anda menggunakan CloudFormation untuk menentukan cache lokal, pada `Cache` properti, untuk`Type`, tentukan`LOCAL`. YAML-formatted CloudFormation Kode sampel berikut menentukan ketiga jenis cache lokal. Anda dapat menentukan kombinasi jenis apa pun. Jika Anda menggunakan cache lapisan Docker, di bawah`Environment`, Anda harus mengatur `PrivilegedMode` ke `true` dan `Type` ke`LINUX_CONTAINER`.

```
CodeBuildProject:
    Type: AWS::CodeBuild::Project
    Properties:
      Name: MyProject
      ServiceRole: <service-role>
      Artifacts:
        Type: S3
        Location: <bucket-name>
        Name: myArtifact
        EncryptionDisabled: true
        OverrideArtifactName: true
      Environment:
        Type: LINUX_CONTAINER
        ComputeType: BUILD_GENERAL1_SMALL
```

```
      Image: aws/codebuild/standard:5.0
      Certificate: <bucket/cert.zip>
      # PrivilegedMode must be true if you specify LOCAL_DOCKER_LAYER_CACHE
      PrivilegedMode: true
    Source:
      Type: GITHUB
      Location: <github-location>
      InsecureSsl: true
      GitCloneDepth: 1
      ReportBuildStatus: false
    TimeoutInMinutes: 10
    Cache:
      Type: LOCAL
      Modes: # You can specify one or more cache mode,
        - LOCAL_CUSTOM_CACHE
        - LOCAL_DOCKER_LAYER_CACHE
        - LOCAL_SOURCE_CACHE
```

> **Note**
>
> Secara default, daemon Docker diaktifkan untuk build non-VPC. Jika Anda ingin menggunakan wadah Docker untuk build VPC, lihat Runtime Privilege dan Kapabilitas Linux di situs Docker Docker dan aktifkan mode istimewa. Selain itu, Windows tidak mendukung mode istimewa.

Lihat informasi yang lebih lengkap di Buat proyek build (CloudFormation).

Debug dibangun di AWS CodeBuild

AWS CodeBuild menyediakan dua metode untuk men-debug build selama pengembangan dan pemecahan masalah. Anda dapat menggunakan lingkungan CodeBuild Sandbox untuk menyelidiki masalah dan memvalidasi perbaikan secara real-time, atau Anda dapat menggunakan AWS Systems Manager Session Manager untuk menyambung ke container build dan melihat status container.

Debug dibangun dengan kotak pasir CodeBuild

Lingkungan CodeBuild kotak pasir menyediakan sesi debug interaktif di lingkungan yang aman dan terisolasi. Anda dapat berinteraksi dengan lingkungan secara langsung melalui perintah Konsol Manajemen AWS atau AWS CLI, mengeksekusi, dan memvalidasi proses build Anda langkah

demi langkah. Ini menggunakan model penagihan per detik yang hemat biaya dan mendukung integrasi asli yang sama dengan penyedia sumber dan AWS layanan seperti lingkungan build Anda. Anda juga dapat terhubung ke lingkungan sandbox menggunakan klien SSH atau dari lingkungan pengembangan terintegrasi (IDE) Anda.

Untuk mempelajari lebih lanjut tentang harga CodeBuild kotak pasir, kunjungi dokumentasi CodeBuild harga. Untuk instruksi terperinci, kunjungi Debug build dengan kotak pasir CodeBuild dokumentasi.

Debug dibangun dengan Session Manager

AWS Systems Manager Session Manager memungkinkan akses langsung ke build yang sedang berjalan di lingkungan eksekusi aktual mereka. Pendekatan ini memungkinkan Anda untuk terhubung ke container build aktif dan memeriksa proses build secara real-time. Anda dapat memeriksa sistem file, memantau proses yang sedang berjalan, dan memecahkan masalah saat terjadi.

Untuk instruksi terperinci, kunjungi Debug dibangun dengan Session Manager dokumentasi.

Debug build dengan kotak pasir CodeBuild

Di AWS CodeBuild, Anda dapat men-debug build dengan menggunakan CodeBuild kotak pasir untuk menjalankan perintah khusus dan memecahkan masalah build Anda.

Topik

- Prasyarat
- Debug build dengan CodeBuild kotak pasir (konsol)
- Debug build dengan CodeBuild kotak pasir (AWS CLI)
- Tutorial: Menghubungkan ke kotak pasir menggunakan SSH
- Pemecahan masalah AWS CodeBuild masalah koneksi SSH sandbox

Prasyarat

Sebelum menggunakan kotak CodeBuild pasir, pastikan peran CodeBuild layanan Anda memiliki kebijakan SSM berikut:

JSON

```
{
```

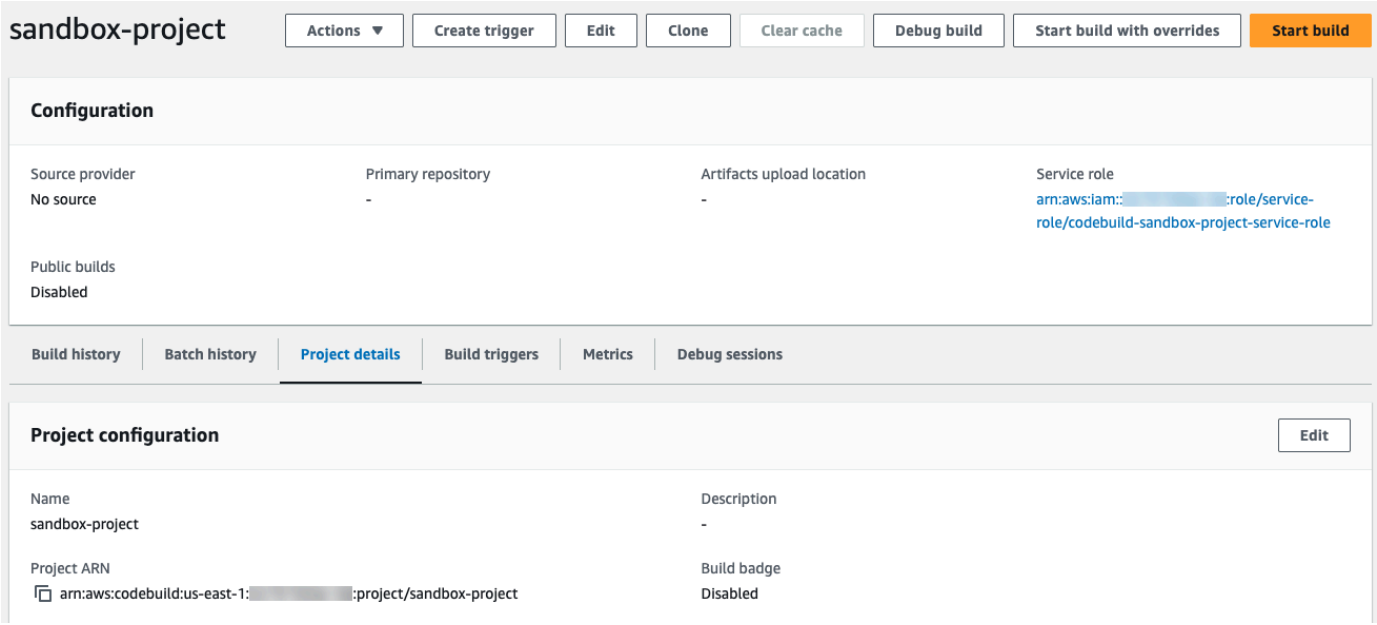
```
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssmmessages:CreateControlChannel",
                "ssmmessages:CreateDataChannel",
                "ssmmessages:OpenControlChannel",
                "ssmmessages:OpenDataChannel"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ssm:StartSession"
            ],
            "Resource": [
                "arn:aws:codebuild:us-east-1:111122223333:build/*",
                "arn:aws:ssm:us-east-1::document/AWS-StartSSHSession"
            ]
        }
    ]
}
```

Debug build dengan CodeBuild kotak pasir (konsol)

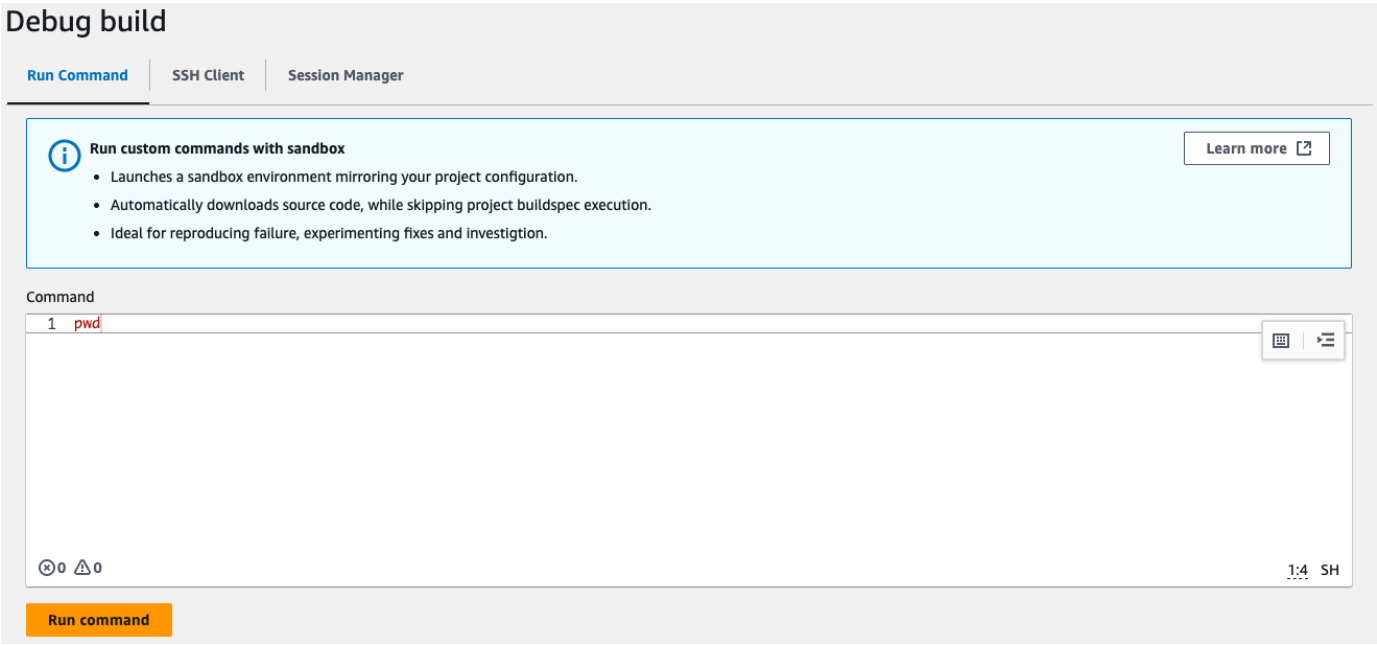
Gunakan petunjuk berikut untuk menjalankan perintah dan menghubungkan klien SSH Anda dengan CodeBuild kotak pasir di konsol.

Jalankan perintah dengan CodeBuild kotak pasir (konsol)

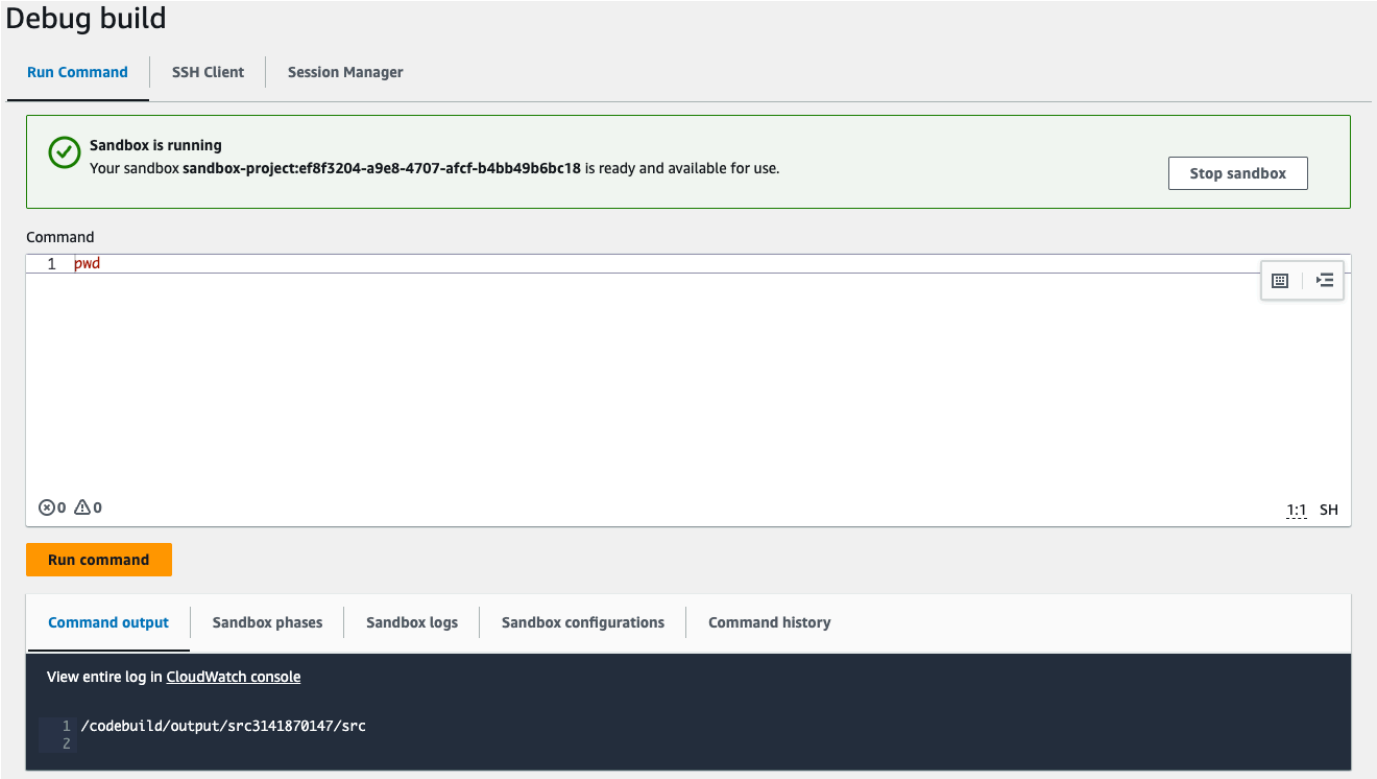
1. Buka AWS CodeBuild konsol di https://console.aws.amazon.com/codesuite/codebuild/home.
2. Di panel navigasi, pilih Bangun proyek. Pilih proyek build, lalu pilih Debug build.



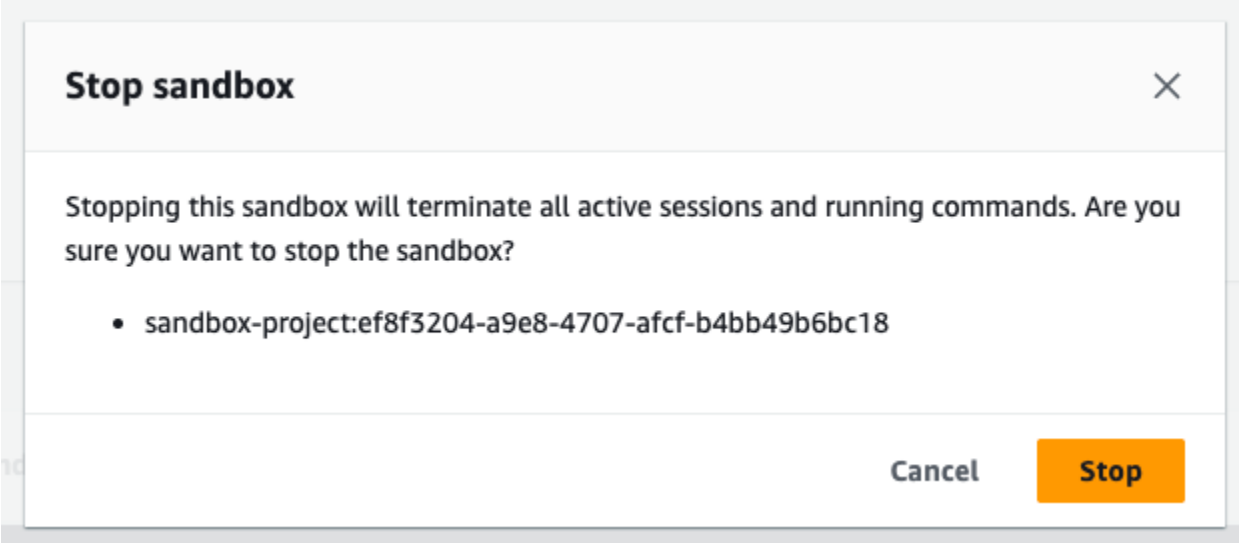
3. Di tab perintah Jalankan, masukkan perintah kustom Anda, lalu pilih J alankan perintah.

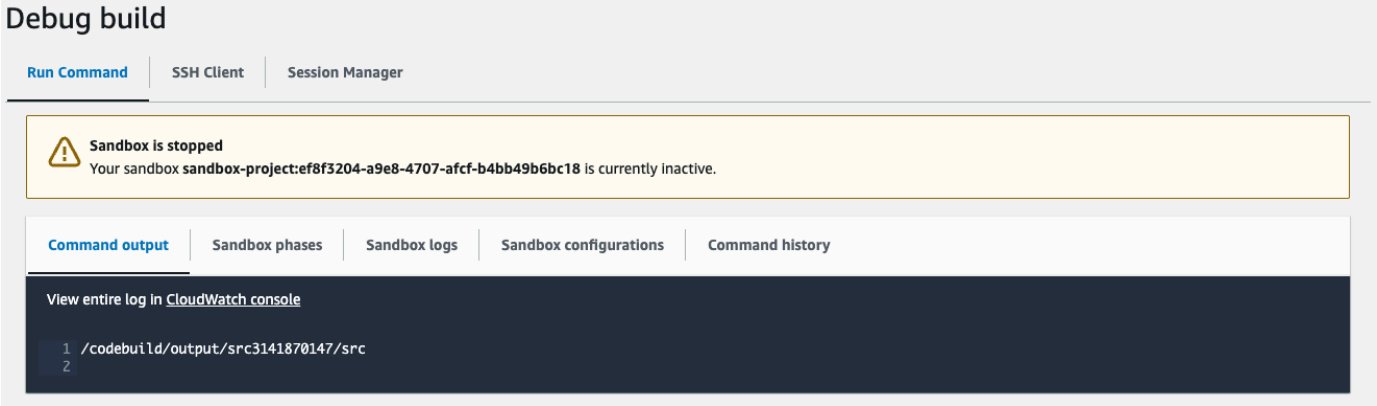


4. CodeBuild Sandbox Anda kemudian akan diinisialisasi dan mulai menjalankan perintah khusus Anda. Output akan ditampilkan di tab Output ketika selesai.



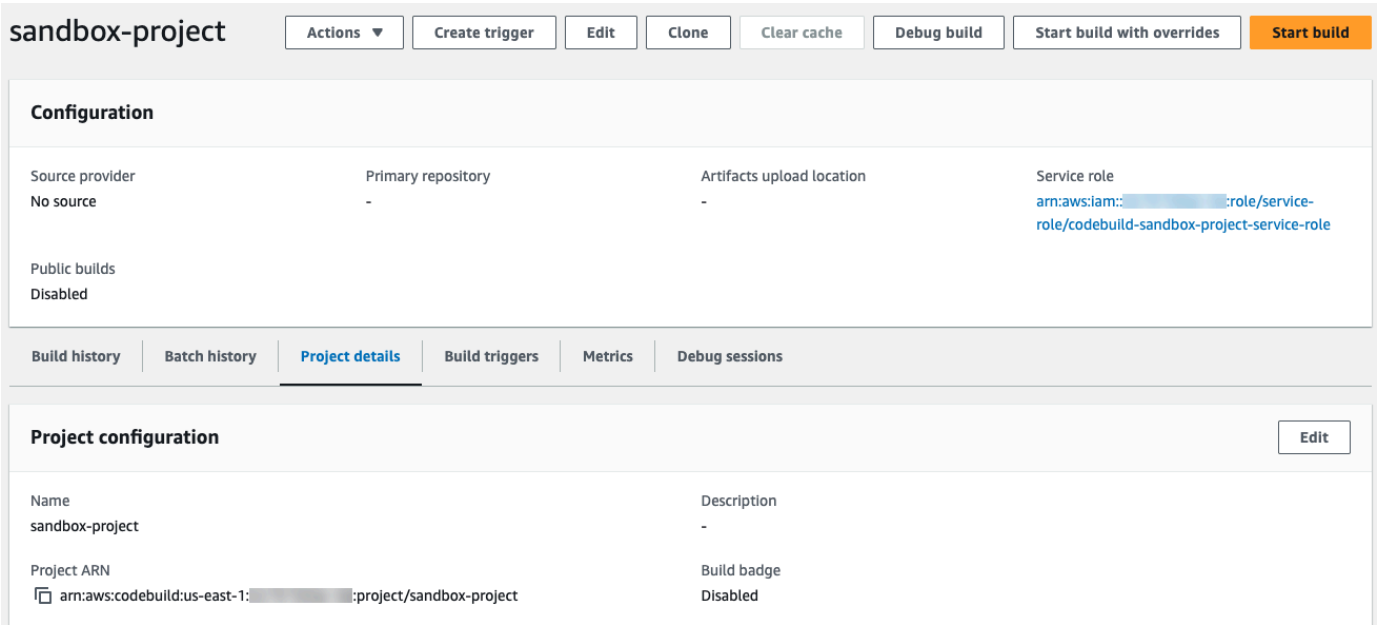
5. Saat pemecahan masalah selesai, Anda dapat menghentikan kotak pasir dengan memilih H entikan kotak pasir. Kemudian pilih Stop untuk mengonfirmasi bahwa kotak pasir Anda akan dihentikan.



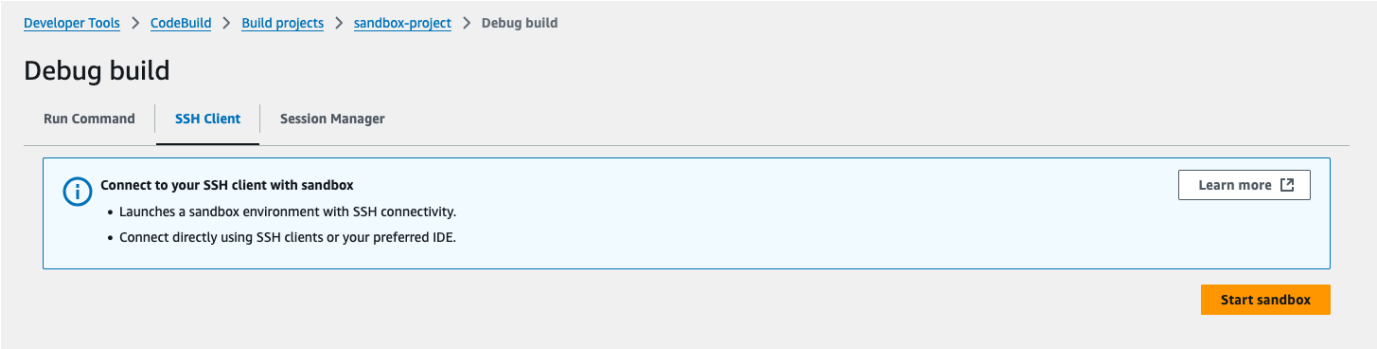


Hubungkan ke klien SSH Anda dengan CodeBuild sandbox (konsol)

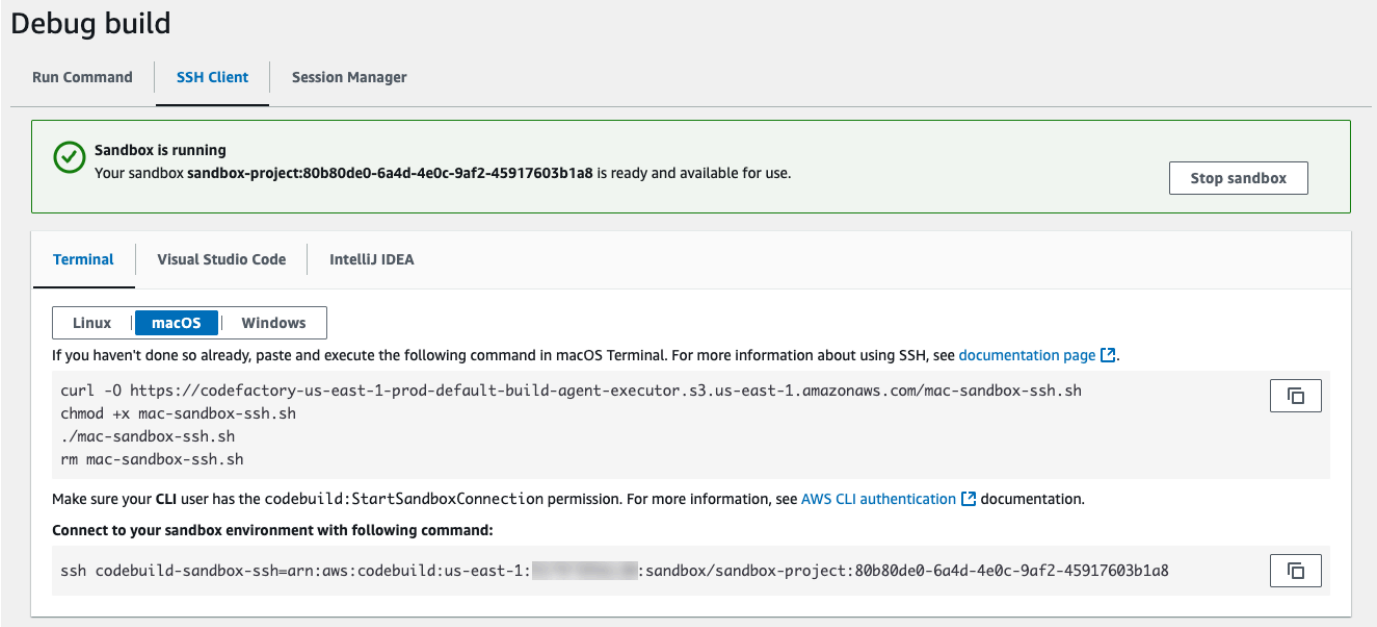
1. Buka AWS CodeBuild konsol di https://console.aws.amazon.com/codesuite/codebuild/home.
2. Di panel navigasi, pilih Bangun proyek. Pilih proyek build, lalu pilih Debug build.



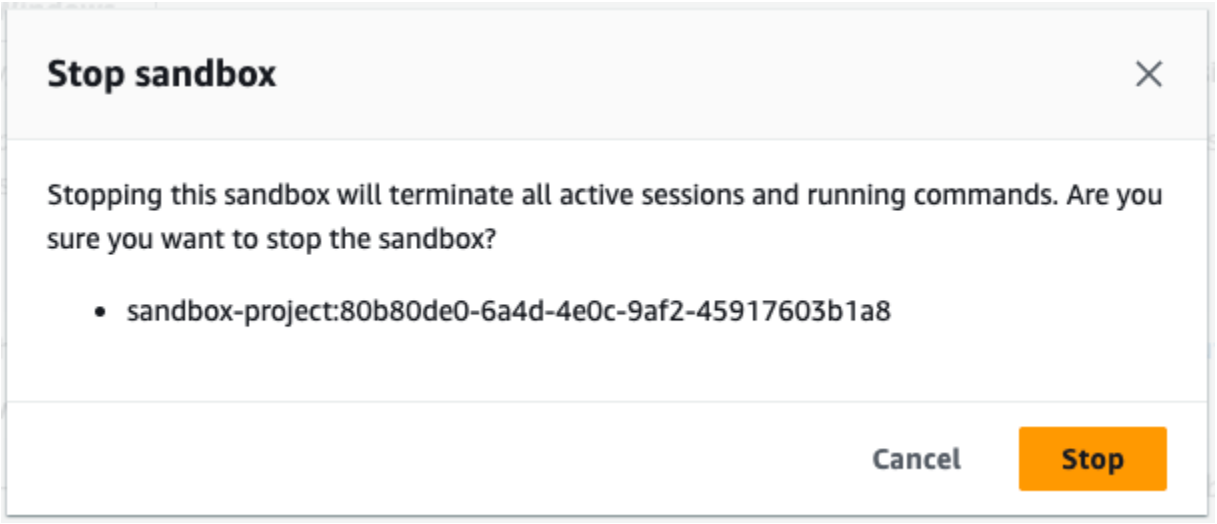
3. Di tab SSH Client dan pilih Start sandbox.

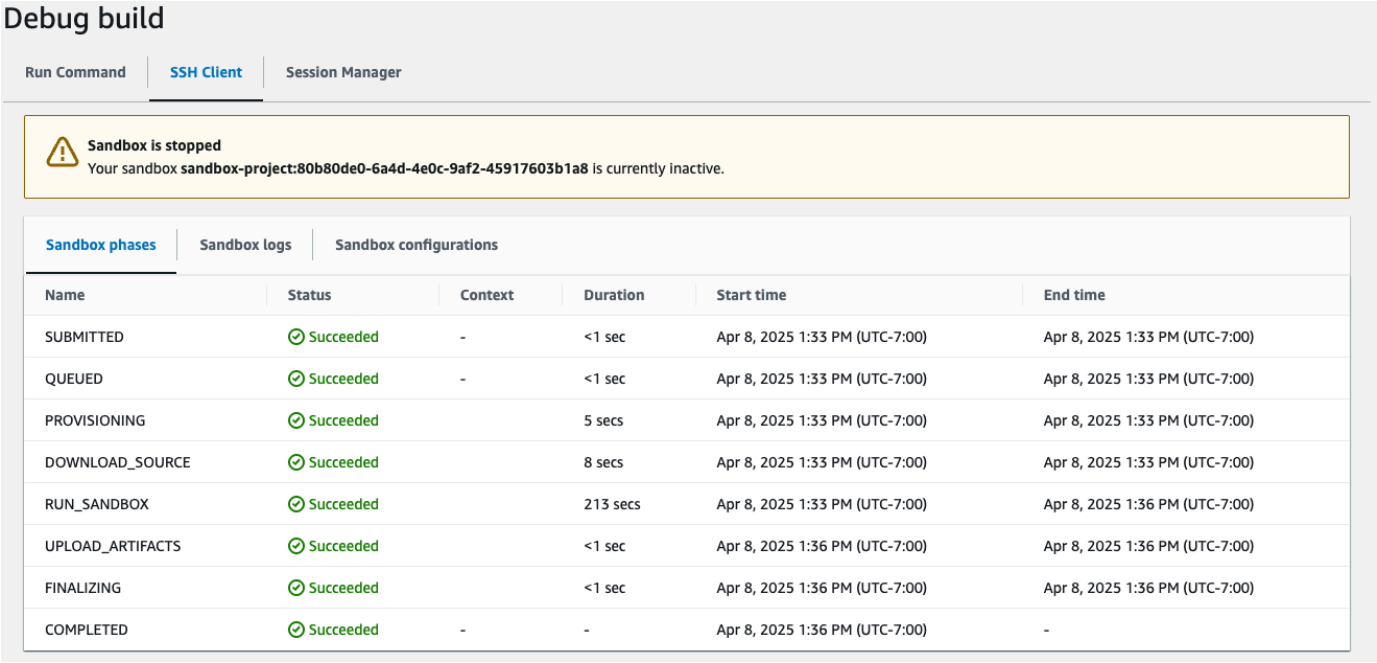


4. Setelah kotak CodeBuild pasir mulai berjalan, ikuti instruksi konsol untuk menghubungkan klien SSH Anda dengan kotak pasir.



5. Saat pemecahan masalah selesai, Anda dapat menghentikan kotak pasir dengan memilih H entikan kotak pasir. Kemudian pilih Stop untuk mengonfirmasi bahwa kotak pasir Anda akan dihentikan.





Debug build dengan CodeBuild kotak pasir (AWS CLI)

Gunakan petunjuk berikut untuk menjalankan perintah dan menghubungkan klien SSH Anda dengan CodeBuild sandbox.

Mulai kotak CodeBuild pasir (AWS CLI)

CLI command

```
aws codebuild start-sandbox --project-name $PROJECT_NAME
```

- `--project-name`: nama CodeBuild proyek

Sample request

```
aws codebuild start-sandbox --project-name "project-name"
```

Sample response

```
{
    "id": "project-name",
    "arn": "arn:aws:codebuild:us-west-2:962803963624:sandbox/project-name",
    "projectName": "project-name",
```

```
    "requestTime": "2025-02-06T11:24:15.560000-08:00",
    "status": "QUEUED",
    "source": {
        "type": "S3",
        "location": "arn:aws:s3:::cofa-e2e-test-1-us-west-2-beta-default-build-
sources/eb-sample-jetty-v4.zip",
        "insecureSsl": false
    },
    "environment": {
        "type": "LINUX_CONTAINER",
        "image": "aws/codebuild/standard:6.0",
        "computeType": "BUILD_GENERAL1_SMALL",
        "environmentVariables": [{
                "name": "foo",
                "value": "bar",
                "type": "PLAINTEXT"
            },
            {
                "name": "bar",
                "value": "baz",
                "type": "PLAINTEXT"
            }
        ],
        "privilegedMode": false,
        "imagePullCredentialsType": "CODEBUILD"
    },
    "timeoutInMinutes": 10,
    "queuedTimeoutInMinutes": 480,
    "logConfig": {
        "cloudWatchLogs": {
            "status": "ENABLED",
            "groupName": "group",
            "streamName": "stream"
        },
        "s3Logs": {
            "status": "ENABLED",
            "location": "codefactory-test-pool-1-us-west-2-beta-default-build-logs",
            "encryptionDisabled": false
        }
    },
    "encryptionKey": "arn:aws:kms:us-west-2:962803963624:alias/SampleEncryptionKey",
    "serviceRole": "arn:aws:iam::962803963624:role/BuildExecutionServiceRole",
    "currentSession": {
        "id": "0103e0e7-52aa-4a3d-81dd-bfc27226fa54",
```

```
        "currentPhase": "QUEUED",
        "status": "QUEUED",
        "startTime": "2025-02-06T11:24:15.626000-08:00",
        "logs": {
            "groupName": "group",
            "streamName": "stream/0103e0e7-52aa-4a3d-81dd-bfc27226fa54",
            "deepLink": "https://console.aws.amazon.com/cloudwatch/home?
region=us-west-2#logsV2:log-groups/log-group/group/log-events/stream
$252F0103e0e7-52aa-4a3d-81dd-bfc27226fa54",
            "s3DeepLink": "https://s3.console.aws.amazon.com/s3/object/
codefactory-test-pool-1-us-west-2-beta-default-build-logs/0103e0e7-52aa-4a3d-81dd-
bfc27226fa54.gz?region=us-west-2",
            "cloudWatchLogsArn": "arn:aws:logs:us-west-2:962803963624:log-
group:group:log-stream:stream/0103e0e7-52aa-4a3d-81dd-bfc27226fa54",
            "s3LogsArn": "arn:aws:s3:::codefactory-test-pool-1-us-west-2-beta-
default-build-logs/0103e0e7-52aa-4a3d-81dd-bfc27226fa54.gz",
            "cloudWatchLogs": {
                "status": "ENABLED",
                "groupName": "group",
                "streamName": "stream"
            },
            "s3Logs": {
                "status": "ENABLED",
                "location": "codefactory-test-pool-1-us-west-2-beta-default-build-
logs",
                "encryptionDisabled": false
            }
        }
    }
}
```

Dapatkan informasi tentang status kotak pasir (AWS CLI)

CLI command

```
aws codebuild batch-get-sandboxes --ids $SANDBOX_IDs
```

Sample request

```
aws codebuild stop-sandbox --id "arn:aws:codebuild:us-west-2:962803963624:sandbox/
project-name"
```

- `--ids`: Daftar `sandboxIds` atau `sandboxArns` yang dipisahkan koma.

Anda dapat memberikan ID kotak pasir atau kotak pasir ARN:

- ID Kotak Pasir: *<codebuild-project-name>*:*<UUID>*

  Misalnya, `project-name:d25be134-05cb-404a-85da-ac5f85d2d72c`.

- Kotak pasir ARN: arn:aws:codebuild: ::sandbox/: *<region>* *<account-id>* *<codebuild-project-name>* *<UUID>*

  Misalnya, `arn:aws:codebuild:us-west-2:962803963624:sandbox/project-name:d25be134-05cb-404a-85da-ac5f85d2d72c`.

Sample response


```
{
    "sandboxes": [{
        "id": "project-name",
        "arn": "arn:aws:codebuild:us-west-2:962803963624:sandbox/project-name",
        "projectName": "project-name",
        "requestTime": "2025-02-06T11:24:15.560000-08:00",
        "endTime": "2025-02-06T11:39:21.587000-08:00",
        "status": "STOPPED",
        "source": {
            "type": "S3",
            "location": "arn:aws:s3:::cofa-e2e-test-1-us-west-2-beta-default-build-
sources/eb-sample-jetty-v4.zip",
            "insecureSsl": false
        },
        "environment": {
            "type": "LINUX_CONTAINER",
            "image": "aws/codebuild/standard:6.0",
            "computeType": "BUILD_GENERAL1_SMALL",
            "environmentVariables": [{
                    "name": "foo",
                    "value": "bar",
                    "type": "PLAINTEXT"
                },
                {
                    "name": "bar",
                    "value": "baz",
```

```
                    "type": "PLAINTEXT"
                }
            ],
            "privilegedMode": false,
            "imagePullCredentialsType": "CODEBUILD"
        },
        "timeoutInMinutes": 10,
        "queuedTimeoutInMinutes": 480,
        "logConfig": {
            "cloudWatchLogs": {
                "status": "ENABLED",
                "groupName": "group",
                "streamName": "stream"
            },
            "s3Logs": {
                "status": "ENABLED",
                "location": "codefactory-test-pool-1-us-west-2-beta-default-build-
logs",
                "encryptionDisabled": false
            }
        },
        "encryptionKey": "arn:aws:kms:us-west-2:962803963624:alias/
SampleEncryptionKey",
        "serviceRole": "arn:aws:iam::962803963624:role/BuildExecutionServiceRole",
        "currentSession": {
            "id": "0103e0e7-52aa-4a3d-81dd-bfc27226fa54",
            "currentPhase": "COMPLETED",
            "status": "STOPPED",
            "startTime": "2025-02-06T11:24:15.626000-08:00",
            "endTime": "2025-02-06T11:39:21.600000-08:00",
            "phases": [{
                    "phaseType": "SUBMITTED",
                    "phaseStatus": "SUCCEEDED",
                    "startTime": "2025-02-06T11:24:15.577000-08:00",
                    "endTime": "2025-02-06T11:24:15.606000-08:00",
                    "durationInSeconds": 0
                },
                {
                    "phaseType": "QUEUED",
                    "phaseStatus": "SUCCEEDED",
                    "startTime": "2025-02-06T11:24:15.606000-08:00",
                    "endTime": "2025-02-06T11:24:16.067000-08:00",
                    "durationInSeconds": 0
                },
```

```
                {
                    "phaseType": "PROVISIONING",
                    "phaseStatus": "SUCCEEDED",
                    "startTime": "2025-02-06T11:24:16.067000-08:00",
                    "endTime": "2025-02-06T11:24:20.519000-08:00",
                    "durationInSeconds": 4,
                    "contexts": [{
                        "statusCode": "",
                        "message": ""
                    }]
                },
                {
                    "phaseType": "DOWNLOAD_SOURCE",
                    "phaseStatus": "SUCCEEDED",
                    "startTime": "2025-02-06T11:24:20.519000-08:00",
                    "endTime": "2025-02-06T11:24:22.238000-08:00",
                    "durationInSeconds": 1,
                    "contexts": [{
                        "statusCode": "",
                        "message": ""
                    }]
                },
                {
                    "phaseType": "RUNNING_SANDBOX",
                    "phaseStatus": "TIMED_OUT",
                    "startTime": "2025-02-06T11:24:22.238000-08:00",
                    "endTime": "2025-02-06T11:39:21.560000-08:00",
                    "durationInSeconds": 899,
                    "contexts": [{
                        "statusCode": "BUILD_TIMED_OUT",
                        "message": "Build has timed out. "
                    }]
                },
                {
                    "phaseType": "COMPLETED",
                    "startTime": "2025-02-06T11:39:21.560000-08:00"
                }
            ],
            "logs": {
                "groupName": "group",
                "streamName": "stream/0103e0e7-52aa-4a3d-81dd-bfc27226fa54",
                "deepLink": "https://console.aws.amazon.com/cloudwatch/
home?region=us-west-2#logsV2:log-groups/log-group/group/log-events/stream
$252F0103e0e7-52aa-4a3d-81dd-bfc27226fa54",
```

```
                "s3DeepLink": "https://s3.console.aws.amazon.com/s3/object/
codefactory-test-pool-1-us-west-2-beta-default-build-logs/0103e0e7-52aa-4a3d-81dd-
bfc27226fa54.gz?region=us-west-2",
                "cloudWatchLogsArn": "arn:aws:logs:us-west-2:962803963624:log-
group:group:log-stream:stream/0103e0e7-52aa-4a3d-81dd-bfc27226fa54",
                "s3LogsArn": "arn:aws:s3:::codefactory-test-pool-1-us-west-2-beta-
default-build-logs/0103e0e7-52aa-4a3d-81dd-bfc27226fa54.gz",
                "cloudWatchLogs": {
                    "status": "ENABLED",
                    "groupName": "group",
                    "streamName": "stream"
                },
                "s3Logs": {
                    "status": "ENABLED",
                    "location": "codefactory-test-pool-1-us-west-2-beta-default-
build-logs",
                    "encryptionDisabled": false
                }
            }
        }
    }],
    "sandboxesNotFound": []
}
```

Hentikan kotak pasir (AWS CLI)

CLI command

```
aws codebuild stop-sandbox --id $SANDBOX-ID
```

- `--id`: A `sandboxId` atau`sandboxArn`.

Sample request

```
aws codebuild stop-sandbox --id "arn:aws:codebuild:us-west-2:962803963624:sandbox/
project-name"
```

Sample response

```
{
    "id": "project-name",
```

```
    "arn": "arn:aws:codebuild:us-west-2:962803963624:sandbox/project-name",
    "projectName": "project-name",
    "requestTime": "2025-02-06T11:24:15.560000-08:00",
    "status": "STOPPING",
    "source": {
        "type": "S3",
        "location": "arn:aws:s3:::cofa-e2e-test-1-us-west-2-beta-default-build-
sources/eb-sample-jetty-v4.zip",
        "insecureSsl": false
    },
    "environment": {
        "type": "LINUX_CONTAINER",
        "image": "aws/codebuild/standard:6.0",
        "computeType": "BUILD_GENERAL1_SMALL",
        "environmentVariables": [{
                "name": "foo",
                "value": "bar",
                "type": "PLAINTEXT"
            },
            {
                "name": "bar",
                "value": "baz",
                "type": "PLAINTEXT"
            }
        ],
        "privilegedMode": false,
        "imagePullCredentialsType": "CODEBUILD"
    },
    "timeoutInMinutes": 10,
    "queuedTimeoutInMinutes": 480,
    "logConfig": {
        "cloudWatchLogs": {
            "status": "ENABLED",
            "groupName": "group",
            "streamName": "stream"
        },
        "s3Logs": {
            "status": "ENABLED",
            "location": "codefactory-test-pool-1-us-west-2-beta-default-build-logs",
            "encryptionDisabled": false
        }
    },
    "encryptionKey": "arn:aws:kms:us-west-2:962803963624:alias/SampleEncryptionKey",
    "serviceRole": "arn:aws:iam::962803963624:role/BuildExecutionServiceRole",
```

```
    "currentSession": {
        "id": "0103e0e7-52aa-4a3d-81dd-bfc27226fa54",
        "currentPhase": "RUN_SANDBOX",
        "status": "STOPPING",
        "startTime": "2025-02-06T11:24:15.626000-08:00",
        "phases": [{
                "phaseType": "SUBMITTED",
                "phaseStatus": "SUCCEEDED",
                "startTime": "2025-02-08T14:33:26.144000-08:00",
                "endTime": "2025-02-08T14:33:26.173000-08:00",
                "durationInSeconds": 0
            },
            {
                "phaseType": "QUEUED",
                "phaseStatus": "SUCCEEDED",
                "startTime": "2025-02-08T14:33:26.173000-08:00",
                "endTime": "2025-02-08T14:33:26.702000-08:00",
                "durationInSeconds": 0
            },
            {
                "phaseType": "PROVISIONING",
                "phaseStatus": "SUCCEEDED",
                "startTime": "2025-02-08T14:33:26.702000-08:00",
                "endTime": "2025-02-08T14:33:30.530000-08:00",
                "durationInSeconds": 3,
                "contexts": [{
                    "statusCode": "",
                    "message": ""
                }]
            },
            {
                "phaseType": "DOWNLOAD_SOURCE",
                "phaseStatus": "SUCCEEDED",
                "startTime": "2025-02-08T14:33:30.530000-08:00",
                "endTime": "2025-02-08T14:33:33.478000-08:00",
                "durationInSeconds": 2,
                "contexts": [{
                    "statusCode": "",
                    "message": ""
                }]
            },
            {
                "phaseType": "RUN_SANDBOX",
                "startTime": "2025-02-08T14:33:33.478000-08:00"
```

```
            }
        ],
        "logs": {
            "groupName": "group",
            "streamName": "stream/0103e0e7-52aa-4a3d-81dd-bfc27226fa54",
            "deepLink": "https://console.aws.amazon.com/cloudwatch/home?
region=us-west-2#logsV2:log-groups/log-group/group/log-events/stream
$252F0103e0e7-52aa-4a3d-81dd-bfc27226fa54",
            "s3DeepLink": "https://s3.console.aws.amazon.com/s3/object/
codefactory-test-pool-1-us-west-2-beta-default-build-logs/0103e0e7-52aa-4a3d-81dd-
bfc27226fa54.gz?region=us-west-2",
            "cloudWatchLogsArn": "arn:aws:logs:us-west-2:962803963624:log-
group:group:log-stream:stream/0103e0e7-52aa-4a3d-81dd-bfc27226fa54",
            "s3LogsArn": "arn:aws:s3:::codefactory-test-pool-1-us-west-2-beta-
default-build-logs/0103e0e7-52aa-4a3d-81dd-bfc27226fa54.gz",
            "cloudWatchLogs": {
                "status": "ENABLED",
                "groupName": "group",
                "streamName": "stream"
            },
            "s3Logs": {
                "status": "ENABLED",
                "location": "codefactory-test-pool-1-us-west-2-beta-default-build-
logs",
                "encryptionDisabled": false
            }
        }
    }
}
```

Mulai eksekusi perintah (AWS CLI)

CLI command

```
aws codebuild start-command-execution --command $COMMAND --type $TYPE --sandbox-id
 $SANDBOX-ID
```

- `--command`: Perintah yang perlu dieksekusi.
- `--sandbox-id`: A `sandboxId` atau`sandboxArn`.
- `--type`: Jenis perintah,`SHELL`.

Sample request

```
aws codebuild start-command-execution --command "echo "Hello World"" --type SHELL --
sandbox-id "arn:aws:codebuild:us-west-2:962803963624:sandbox/project-name
```

Sample response

```
{
    "id": "e1c658c2-02bb-42a8-9abb-94835241fcd6",
    "sandboxId": "f7126a4a-b0d5-452f-814c-fea73718f805",
    "submitTime": "2025-02-06T20:12:02.683000-08:00",
    "status": "SUBMITTED",
    "command": "echo \"Hello World\"",
    "type": "SHELL",
    "logs": {
        "groupName": "group",
        "streamName": "stream",
        "deepLink": "https://console.aws.amazon.com/cloudwatch/home?region=us-
west-2#logsV2:log-groups/log-group/group/log-events/stream",
        "s3DeepLink": "https://s3.console.aws.amazon.com/s3/object/codefactory-test-
pool-1-us-west-2-beta-default-build-logs/f7126a4a-b0d5-452f-814c-fea73718f805.gz?
region=us-west-2",
        "cloudWatchLogsArn": "arn:aws:logs:us-west-2:962803963624:log-
group:group:log-stream:stream",
        "s3LogsArn": "arn:aws:s3:::codefactory-test-pool-1-us-west-2-beta-default-
build-logs/f7126a4a-b0d5-452f-814c-fea73718f805.gz",
        "cloudWatchLogs": {
            "status": "ENABLED",
            "groupName": "group",
            "streamName": "stream"
        },
        "s3Logs": {
            "status": "ENABLED",
            "location": "codefactory-test-pool-1-us-west-2-beta-default-build-logs",
            "encryptionDisabled": false
        }
    }
}
```

Dapatkan informasi tentang eksekusi perintah (AWS CLI)

CLI command

```
aws codebuild batch-get-command-executions --command-execution-ids $COMMAND-IDs --
sandbox-id $SANDBOX-IDs
```

- `--command-execution-ids`: Daftar yang dipisahkan koma. `commandExecutionIds`
- `--sandbox-id`: A `sandboxId` atau`sandboxArn`.

Sample request

```
aws codebuild batch-get-command-executions --command-execution-
ids"c3c085ed-5a8f-4531-8e95-87d547f27ffd" --sandbox-id "arn:aws:codebuild:us-
west-2:962803963624:sandbox/project-name"
```

Sample response

```
{
    "commandExecutions": [{
        "id": "c3c085ed-5a8f-4531-8e95-87d547f27ffd",
        "sandboxId": "cd71e456-2a4c-4db4-ada5-da892b0bba05",
        "submitTime": "2025-02-10T20:18:17.118000-08:00",
        "startTime": "2025-02-10T20:18:17.939000-08:00",
        "endTime": "2025-02-10T20:18:17.976000-08:00",
        "status": "SUCCEEDED",
        "command": "echo \"Hello World\"",
        "type": "SHELL",
        "exitCode": "0",
        "standardOutputContent": "Hello World\n",
        "logs": {
            "groupName": "group",
            "streamName": "stream",
            "deepLink": "https://console.aws.amazon.com/cloudwatch/home?region=us-
west-2#logsV2:log-groups/log-group/group/log-events/stream",
            "s3DeepLink": "https://s3.console.aws.amazon.com/s3/object/
codefactory-test-pool-1-us-west-2-beta-default-build-logs/cd71e456-2a4c-4db4-ada5-
da892b0bba05.gz?region=us-west-2",
            "cloudWatchLogsArn": "arn:aws:logs:us-west-2:962803963624:log-
group:group:log-stream:stream",
```

```
            "s3LogsArn": "arn:aws:s3:::codefactory-test-pool-1-us-west-2-beta-
default-build-logs/cd71e456-2a4c-4db4-ada5-da892b0bba05.gz",
            "cloudWatchLogs": {
                "status": "ENABLED",
                "groupName": "group",
                "streamName": "stream"
            },
            "s3Logs": {
                "status": "ENABLED",
                "location": "codefactory-test-pool-1-us-west-2-beta-default-build-
logs",
                "encryptionDisabled": false
            }
        }
    }],
    "commandExecutionsNotFound": []
}
```

Daftar eksekusi perintah untuk kotak pasir (AWS CLI)

CLI command

```
aws codebuild list-command-executions-for-sandbox --sandbox-id $SANDBOX-ID --next-
token $NEXT_TOKEN --max-results $MAX_RESULTS --sort-order $SORT_ORDER
```

- `--next-token`: Token berikutnya, jika ada, untuk mendapatkan hasil yang dipaginasi. Anda akan mendapatkan nilai ini dari eksekusi kotak pasir daftar sebelumnya.
- `--max-results`: (Opsional) Jumlah maksimum catatan sandbox yang akan diambil.
- `--sort-order`: Urutan di mana catatan kotak pasir harus diambil.

Sample request

```
aws codebuild list-command-executions-for-sandbox --sandbox-id
 "arn:aws:codebuild:us-west-2:962803963624:sandbox/project-name"
```

Sample response

```
{
    "commandExecutions": [{
            "id": "aad6687e-07bc-45ab-a1fd-f5440229b528",
```

```
            "sandboxId": "cd71e456-2a4c-4db4-ada5-da892b0bba05",
            "submitTime": "2025-02-10T20:18:35.304000-08:00",
            "startTime": "2025-02-10T20:18:35.615000-08:00",
            "endTime": "2025-02-10T20:18:35.651000-08:00",
            "status": "FAILED",
            "command": "fail command",
            "type": "SHELL",
            "exitCode": "127",
            "standardErrContent": "/codebuild/output/tmp/script.sh: 4: fail: not
 found\n",
            "logs": {
                "groupName": "group",
                "streamName": "stream",
                "deepLink": "https://console.aws.amazon.com/cloudwatch/home?
region=us-west-2#logsV2:log-groups/log-group/group/log-events/stream",
                "s3DeepLink": "https://s3.console.aws.amazon.com/s3/object/
codefactory-test-pool-1-us-west-2-beta-default-build-logs/cd71e456-2a4c-4db4-ada5-
da892b0bba05.gz?region=us-west-2",
                "cloudWatchLogsArn": "arn:aws:logs:us-west-2:962803963624:log-
group:group:log-stream:stream",
                "s3LogsArn": "arn:aws:s3:::codefactory-test-pool-1-us-west-2-beta-
default-build-logs/cd71e456-2a4c-4db4-ada5-da892b0bba05.gz",
                "cloudWatchLogs": {
                    "status": "ENABLED",
                    "groupName": "group",
                    "streamName": "stream"
                },
                "s3Logs": {
                    "status": "ENABLED",
                    "location": "codefactory-test-pool-1-us-west-2-beta-default-
build-logs",
                    "encryptionDisabled": false
                }
            }
        },
        {
            "id": "c3c085ed-5a8f-4531-8e95-87d547f27ffd",
            "sandboxId": "cd71e456-2a4c-4db4-ada5-da892b0bba05",
            "submitTime": "2025-02-10T20:18:17.118000-08:00",
            "startTime": "2025-02-10T20:18:17.939000-08:00",
            "endTime": "2025-02-10T20:18:17.976000-08:00",
            "status": "SUCCEEDED",
            "command": "echo \"Hello World\"",
            "type": "SHELL",
```

```
            "exitCode": "0",
            "standardOutputContent": "Hello World\n",
            "logs": {
                "groupName": "group",
                "streamName": "stream",
                "deepLink": "https://console.aws.amazon.com/cloudwatch/home?
region=us-west-2#logsV2:log-groups/log-group/group/log-events/stream",
                "s3DeepLink": "https://s3.console.aws.amazon.com/s3/object/
codefactory-test-pool-1-us-west-2-beta-default-build-logs/cd71e456-2a4c-4db4-ada5-
da892b0bba05.gz?region=us-west-2",
                "cloudWatchLogsArn": "arn:aws:logs:us-west-2:962803963624:log-
group:group:log-stream:stream",
                "s3LogsArn": "arn:aws:s3:::codefactory-test-pool-1-us-west-2-beta-
default-build-logs/cd71e456-2a4c-4db4-ada5-da892b0bba05.gz",
                "cloudWatchLogs": {
                    "status": "ENABLED",
                    "groupName": "group",
                    "streamName": "stream"
                },
                "s3Logs": {
                    "status": "ENABLED",
                    "location": "codefactory-test-pool-1-us-west-2-beta-default-
build-logs",
                    "encryptionDisabled": false
                }
            }
        }
    ]
}
```

Daftar kotak pasir (AWS CLI)

CLI command

```
aws codebuild list-sandboxes --next-token $NEXT_TOKEN --max-results $MAX_RESULTS --
sort-order $SORT_ORDER
```

Sample request

```
aws codebuild list-sandboxes
```

Sample response


```
{
    "ids": [
        "s3-log-project-integ-test-temp1739250628149 85d64e0f-7880-41df-9a3c-
fb6597a266d2:827a5243-0841-4b69-a720-4438796f6967",
        "s3-log-project-integ-test-temp1739249999716bbd438dd-8bb8-47bd-
ba6b-0133ac65b3d3:e2fa4eab-73af-42e3-8903-92fddaf9f378",
        "s3-log-project-integ-test-
temp17392474779450fbdacc2-2d6e-4190-9ad5-28f891bb7415:cd71e456-2a4c-4db4-ada5-
da892b0bba05",
        "s3-log-project-integ-test-temp1739224628416430 1421c-5030-4fa1-b4d3-
ca15e44771c5:9e26ab3f-65e4-4896-a19c-56b1a95e630a",
        "s3-log-project-integ-test-temp17392136731949705 6d8d-6d8e-4f5a-a37c-
a62f5686731f:22d91b06-df1e-4e9c-a664-c0abb8d5920b",
        "s3-log-project-integ-test-temp1739213439503f6283f19-390c-4dc8-95a9-
c8480113384a:82cc413e-fc46-47ab-898f-ae23c83a613f",
        "s3-log-project-integ-test-temp1739054385570b1f1ddc2-0a23-4062-
bd0c-24e9e4a99b99:c02562f3-2396-42ec-98da-38e3fe5da13a",
        "s3-log-project-integ-test-temp173905400540237dab1ac-1fde-4dfb-a8f5-
c0114333dc89:d2f30493-f65e-4fa0-a7b6-08a5e77497b9",
        "s3-log-project-integ-test-
temp17390534055719c534090-7bc4-48f1-92c5-34acaec5bf1e:df5f1c8a-f017-43b7-91ba-
ad2619e2c059",
        "s3-log-project-integ-test-temp1739052719086a61813cc-
ebb9-4db4-9391-7f43cc984ee4:d61917ec-8037-4647-8d52-060349272c4a",
        "s3-log-project-integ-test-temp173898670094078b67edb-
c42f-42ed-9db2-4b5c1a5fc66a:ce33dfbc-beeb-4466-8c99-a3734a0392c7",
        "s3-log-project-integ-test-
temp17389863425584d21b7cd-32e2-4f11-9175-72c89ecaffef:046dadf0-1f3a-4d51-a2c0-
e88361924acf",
        "s3-log-project-integ-test-
temp1738985884273977ccd23-394b-46cc-90d3-7ab94cf764dc:0370dc41-9339-4b0a-91ed-51929761b244",
        "s3-log-project-integ-test-temp1738985365972241b614f-8e41-4387-
bd25-2b8351fbc9e0:076c392a-9630-47d8-85a9-116aa34edfff",
        "s3-log-project-integ-test-
temp1738985043988a51a9e2b-09d6-4d24-9c3c-1e6e21ac9fa8:6ea3949c-435b-4177-
aa4d-614d5956244c",
        "s3-log-project-integ-test-temp1738984123354c68b31ad-49d1-4f4b-981d-
b66c00565ff6:6c3fff6c-815b-48b5-ada3-737400a6dee8",
        "s3-log-project-integ-test-
temp1738977263715d4d5bf6c-370a-48bf-8ea6-905358a6cf92:968a0f54-724a-42d1-9207-6ed854b2fae8",
```

```
        "s3-log-project-integ-test-
temp173897358796816ce8d7d-2a5e-41ef-855b-4a94a8d2795d:80f9a7ce-930a-402e-934e-
d8b511d68b04",
        "s3-log-project-integ-test-temp17389730633301af5e452-0966-467c-
b684-4e36d47f568c:cabbe989-2e8a-473c-af25-32edc8c28646",
        "s3-log-project-integ-test-temp1738901503813173fd468-
b723-4d7b-9f9f-82e88d17f264:f7126a4a-b0d5-452f-814c-fea73718f805",
        "s3-log-project-integ-test-temp1738890502472c13616fb-
bd0f-4253-86cc-28b74c97a0ba:c6f197e5-3a53-45b6-863e-0e6353375437",
        "s3-log-project-integ-test-
temp1738890304468361 0daf3-8da7-43c6-8580-9978432432ce:d20aa317-8838-4966-
bbfc-85b908213df1",
        "s3-log-project-integ-test-temp173888857196780b5ab8b-e54b-44fd-a222-
c5a374fffe96:ab4b9970-ffae-47a0-b3a8-7b6790008cad",
        "s3-log-project-integ-test-temp1738888336931c11d378d-e74d-49a4-
a723-3b92e6f7daac:4922f0e8-9b7d-4119-9c9f-115cd85e703e",
        "s3-log-project-integ-test-temp17388881717651612a397-c23f-4d88-
ba87-2773cd3fc0c9:be91c3fc-418e-4feb-8a3a-ba58ff8f4e8a",
        "s3-log-project-integ-test-
temp17388879727174c3c62ed-6195-4afb-8a03-59674d0e1187:a48826a8-3c0d-43c5-
a1b5-1c98a0f978e9",
        "s3-log-project-integ-test-temp1738885948597cef305e4-b8b4-46b0-a65b-
e2d0a7b83294:c050e77d-e3f8-4829-9a60-46149628fe96",
        "s3-log-project-integ-test-temp173888561463001a7d2a8-
e4e4-4434-94db-09d3da9a9e17:8c3ac3f5-7111-4297-aec9-2470d3ead873",
        "s3-log-project-integ-test-
temp1738869855076eb19cafd-04fe-41bd-8aa0-40826d0c0d27:d25be134-05cb-404a-85da-
ac5f85d2d72c",
        "s3-project-integ-test-temp1738868157467148eacfc-d39b-49fc-a137-
e55381cd2978:4909557b-c221-4814-b4b6-7d9e93d37c35",
        "s3-project-integ-test-temp1738820926895abec0af2-
e33d-473c-9cf4-2122dd9d6876:8f5cf218-71d6-40a4-a4be-6cacebd7765f",
        "s3-project-integ-test-temp173881998877574f969a6-1c2e-4441-b463-
ab175b45ce32:04396851-c901-4986-9117-585528e3877f",
        "s3-project-integ-test-temp17388189812309abd2604-29ba-4cf6-
b6bf-073207b7db9c:540075c7-f5ec-41e8-9341-2233c09247eb",
        "s3-project-integ-test-temp1738818843474d3ea9ac1-b609-461b-
bbdb-2da245c9bc96:865d4c3c-fbfe-4ece-9c92-d0c928341404",
        "s3-project-integ-test-temp1738818542236006e9169-e6d9-4344-9b59-
f557e7aec619:1f9ffa87-da15-4290-83e2-eebdd877497b",
        "s3-project-integ-test-
temp173881809557486ad11fd-7931-48d7-81d5-499cea52a6bc:c4c2efc4-685f-4e13-8b0f-1ef85ec300b1",
        "s3-project-integ-test-temp1738817941033229410 20-3f0b-49c3-b836-
fcd818ec9484:0344cfba-de48-456d-b2a8-6566bd4a5d6e",
```

```
        "s3-project-integ-test-temp1738817680747b93d0d0b-ea16-497f-9559-
af25ee6dcfdf:654a3a55-d92a-4dc6-8da8-56fd4d40d7e1",
        "s3-project-integ-test-temp17388174027191255c3da-086c-4270-b047-
acac0b7bee0d:b7e82740-2c69-42fc-ab5a-dbf15bc016a1",
        "s3-project-integ-test-temp1738817099799016e7fa3-b9b5-46a2-
bcd5-0888c646743f:8705a6a4-79ff-427a-a1c3-85c4e8fe462e",
        "s3-project-integ-test-temp1738816479281bb0c3606-5ebf-4623-
bed5-12b60e9d3512:f23fc74b-a981-4835-8e28-375fcd4c99e4",
        "s3-project-integ-test-
temp1738816263585c939a133-4d37-482c-9238-1dbff34b7674:ca28e234-0045-4ae6-8732-938b17597f50",
        "s3-project-integ-test-
temp173881580873072d18733-8fe4-43b1-83f7-95f25bb27ccf:c6f0f55b-5736-47c7-
a3aa-1b8461a6d5ed"
    ]
}
```

Tutorial: Menghubungkan ke kotak pasir menggunakan SSH

Tutorial ini menunjukkan cara menghubungkan ke kotak CodeBuild pasir menggunakan klien SSH.

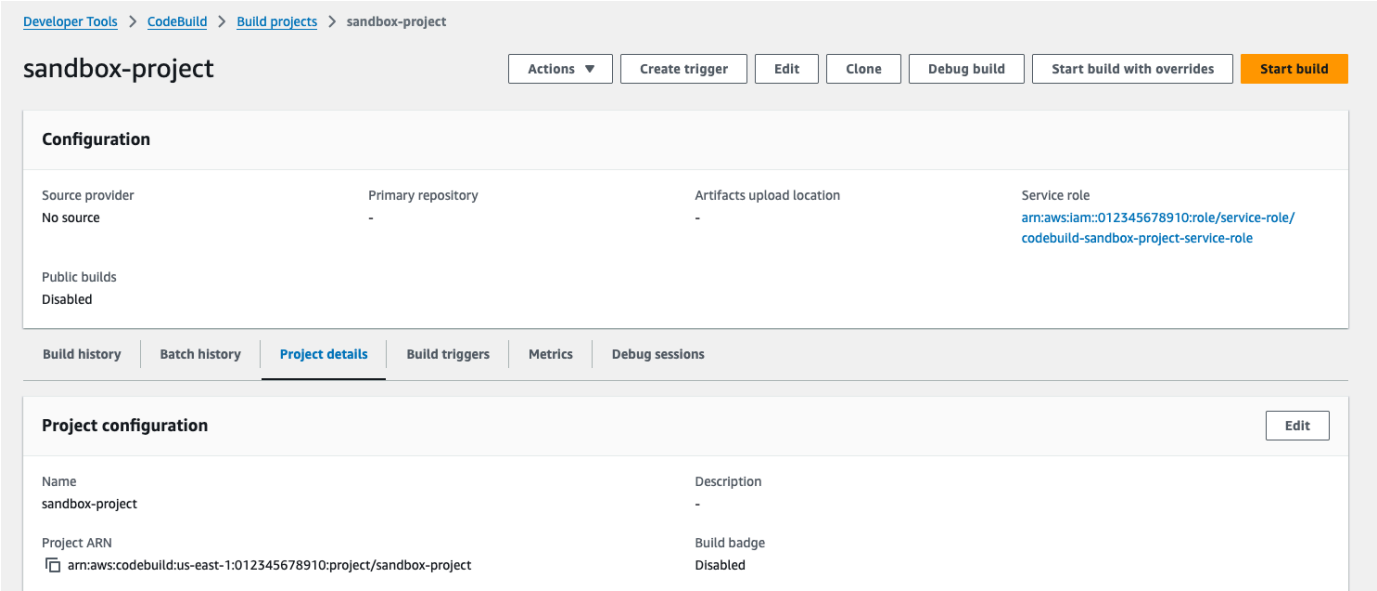
Untuk menyelesaikan tutorial ini, Anda harus terlebih dahulu:

- Pastikan Anda memiliki AWS CodeBuild proyek yang sudah ada.
- Siapkan izin IAM yang sesuai yang dikonfigurasi untuk peran CodeBuild proyek Anda.
- Instal dan konfigur AWS CLI asikan pada mesin lokal Anda.

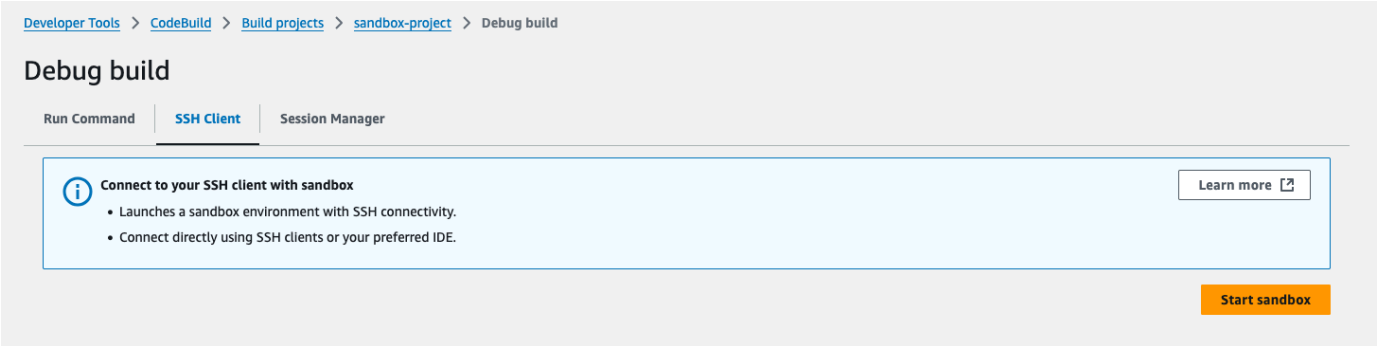
Langkah 1: Mulai kotak pasir

Untuk memulai kotak CodeBuild pasir di konsol

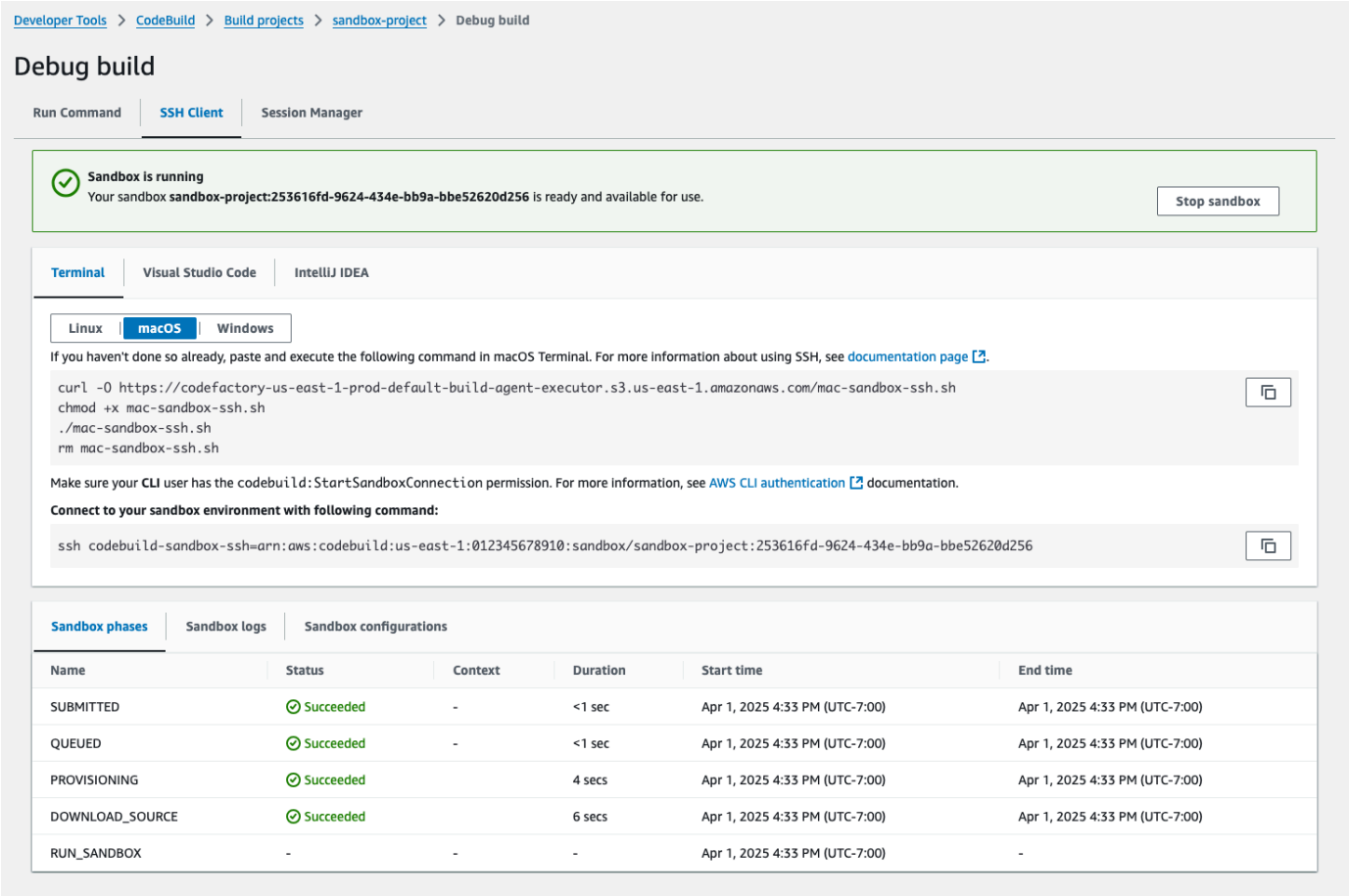
1. Buka AWS CodeBuild konsol di https://console.aws.amazon.com/codesuite/codebuild/home.
2. Di panel navigasi, pilih Bangun proyek. Pilih proyek build, lalu pilih Debug build.



3. Di tab SSH Client dan pilih Start sandbox.



4. Proses inisialisasi kotak pasir mungkin memakan waktu. Anda dapat terhubung ke kotak pasir saat statusnya berubah menjadi`RUN_SANDDB0X`.

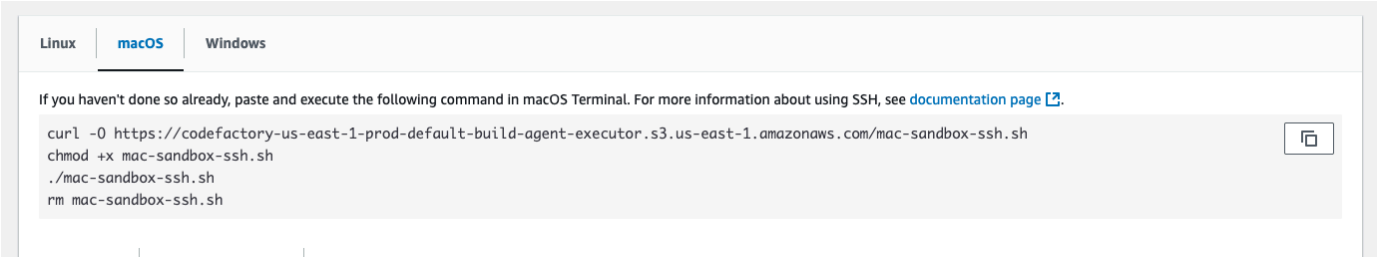


Langkah 2: Ubah konfigurasi SSH lokal

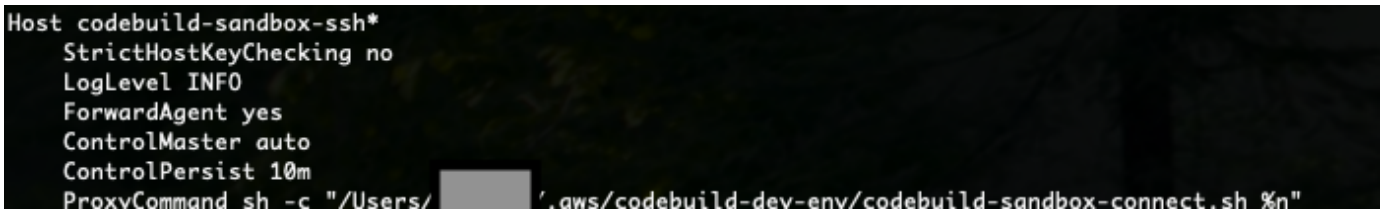
Jika Anda menyambung ke kotak pasir untuk pertama kalinya, Anda perlu melakukan proses penyiapan satu kali menggunakan langkah-langkah berikut:

Untuk memodifikasi konfigurasi SSH lokal di konsol

1. Temukan perintah pengaturan untuk sistem operasi Anda.
2. Buka terminal lokal Anda, lalu salin dan jalankan perintah yang disediakan untuk mengunduh dan menjalankan skrip untuk mengatur konfigurasi SSH lokal Anda. Misalnya, jika sistem operasi Anda adalah macOS, gunakan perintah berikut:



3. Skrip konfigurasi akan menambahkan konfigurasi yang diperlukan untuk menghubungkan ke kotak pasir Anda. Anda akan diminta untuk menerima perubahan ini.
4. Setelah konfigurasi berhasil, entri konfigurasi SSH baru untuk CodeBuild sandbox akan dibuat.



Langkah 3: Hubungkan ke kotak pasir

Untuk memodifikasi konfigurasi SSH lokal di konsol

1. Konfigur AWS CLI asikan Otentikasi dan pastikan AWS CLI pengguna Anda memiliki `codebuild:StartSandboxConnection` izin. Untuk informasi selengkapnya, lihat Mengautentikasi menggunakan kredentif pengguna IAM untuk AWS CLI dalam Panduan Pengguna Antar AWS muka Baris Perintah untuk Versi 1.
2. Hubungkan ke kotak pasir Anda dengan perintah berikut:

```
ssh codebuild-sandbox-ssh=arn:aws:codebuild:us-east-1:<account-id>:sandbox/<sandbox-id>
```

> **Note**
>
> Untuk memecahkan masalah kegagalan koneksi, gunakan `-v` flag untuk mengaktifkan output verbose. Misalnya, `ssh -v codebuild-sandbox-ssh=arn:aws:codebuild:us-east-1:<account-id>:sandbox/<sandbox-id>`. Untuk panduan pemecahan masalah tambahan, lihatPemecahan masalah AWS CodeBuild masalah koneksi SSH sandbox.

Langkah 4: Tinjau hasil Anda

Setelah terhubung, Anda dapat men-debug kegagalan build, menguji perintah build, bereksperimen dengan perubahan konfigurasi, dan memverifikasi variabel lingkungan dan dependensi dengan kotak pasir Anda.

Pemecahan masalah AWS CodeBuild masalah koneksi SSH sandbox

Gunakan informasi dalam topik ini untuk membantu Anda mengidentifikasi, mendiagnosis, dan mengatasi masalah koneksi SSH CodeBuild sandbox.

Topik

- StartSandboxConnectionInvalidInputExceptionkesalahan saat SSH masuk ke lingkungan CodeBuild kotak pasir
- Kesalahan: “Tidak dapat menemukan kredenSIAL” saat SSH masuk ke lingkungan CodeBuild kotak pasir
- StartSandboxConnectionAccessDeniedExceptionkesalahan saat SSH masuk ke lingkungan CodeBuild kotak pasir
- Kesalahan: “ssh: Tidak dapat menyelesaikan nama host” saat SSH masuk ke lingkungan CodeBuild kotak pasir

`StartSandboxConnectionInvalidInputException`kesalahan saat SSH masuk ke lingkungan CodeBuild kotak pasir

Masalah: Saat mencoba menyambung ke lingkungan CodeBuild sandbox menggunakan perintah`ssh codebuild-sandbox-ssh=<sandbox-arn>`, Anda mungkin mengalami `InvalidInputException` kesalahan seperti:

```
An error occurred (InvalidInputException) when calling the StartSandboxConnection
operation: Failed to start SSM session for {sandbox-arn}
User: arn:aws:sts::<account-ID>:assumed-role/<service-role-name>/AWSCodeBuild-<UUID>
is not authorized to perform: ssm:StartSession on resource.
```

```
An error occurred (InvalidInputException) when calling the StartSandboxConnection
operation: Failed to start SSM session for
sandbox <sandbox-arn>: codebuild:<UUID> is not connected.
```

Kemungkinan penyebab :

- Agen Manajer Sistem Amazon EC2 Hilang: Image build tidak memiliki agen SSM yang diinstal atau dikonfigurasi dengan benar.
- Izin Tidak Mencukupi: Peran layanan CodeBuild proyek tidak memiliki izin SSM yang diperlukan.

Solusi yang disarankan: Jika Anda menggunakan gambar khusus untuk build Anda, lakukan hal berikut.

1. Instal SSM Agent. Untuk informasi selengkapnya, lihat Menginstal dan menghapus Agen SSM secara manual di instans Amazon EC2 untuk Linux di. Versi Agen SSM harus `3.0.1295.0` atau lebih baru.
2. Salin file, https://github.com/aws/aws-codebuild-docker-images/blob/master/ubuntu/standard/7.0/amazon-ssm-agent.json ke `/etc/amazon/ssm/` direktori di gambar Anda. Ini mengaktifkan Mode Kontainer di agen SSM.
3. Pastikan peran layanan CodeBuild proyek Anda memiliki izin berikut, lalu mulai ulang lingkungan sandbox:

```
{
   "Effect": "Allow",
      "Action": [
         "ssmmessages:CreateControlChannel",
         "ssmmessages:CreateDataChannel",
         "ssmmessages:OpenControlChannel",
         "ssmmessages:OpenDataChannel"
      ],
      "Resource": "*"
 },
 {
    "Effect": "Allow",
    "Action": [
       "ssm:StartSession"
     ],
     "Resource": [
        "arn:aws:codebuild:region:account-id:build/*",
        "arn:aws:ssm:region::document/AWS-StartSSHSession"
     ]
 }
```

Kesalahan: “Tidak dapat menemukan kredenSIAL” saat SSH masuk ke lingkungan CodeBuild kotak pasir

Masalah: Saat mencoba menyambung ke lingkungan CodeBuild sandbox menggunakan perintah`ssh codebuild-sandbox-ssh=<sandbox-arn>`, Anda mungkin mengalami kesalahan kredenSIAL berikut:

```
Unable to locate credentials. You can configure credentials by running
"aws configure".
```

Kemungkinan penyebab: AWS kredenSIAL belum dikonfigurasi dengan benar di lingkungan lokal Anda.

Solusi yang disarankan: Konfigurasikan AWS CLI kredenSIAL Anda dengan mengikuti dokumentasi resmi: Meng konfigurasi pengaturan untuk Panduan Pengguna AWS Antarmuka Baris Perintah untuk Versi 2. AWS CLI

`StartSandboxConnectionAccessDeniedException`kesalahan saat SSH masuk ke lingkungan CodeBuild kotak pasir

Masalah: Saat mencoba menyambung ke lingkungan CodeBuild sandbox menggunakan perintah`ssh codebuild-sandbox-ssh=<sandbox-arn>`, Anda mungkin mengalami kesalahan izin berikut:

```
An error occurred (AccessDeniedException) when calling the StartSandboxConnection
operation:
User: arn:aws:sts::account-id:assumed-role/role-name
is not authorized to perform: codebuild:StartSandboxConnection on resource:
sandbox-arn
because no identity-based policy allows the codebuild:StartSandboxConnection action
```

Kemungkinan penyebab: Kre AWS denSIAL Anda tidak memiliki CodeBuild izin yang diperlukan untuk melakukan operasi ini.

Solusi yang disarankan: Pastikan pengguna atau peran IAM yang terkait dengan AWS CLI kredensional Anda memiliki izin berikut:

```
{
    "Effect": "Allow",
    "Action": [
       "codebuild:StartSandboxConnection"
     ],
     "Resource": [
        "arn:aws:codebuild:region:account-id:sandbox/*"
     ]
}
```

Kesalahan: “ssh: Tidak dapat menyelesaikan nama host” saat SSH masuk ke lingkungan CodeBuild kotak pasir

Masalah: Saat mencoba menyambung ke lingkungan CodeBuild sandbox menggunakan perintah`ssh codebuild-sandbox-ssh=<sandbox-arn>`, Anda mengalami kesalahan resolusi nama host berikut:

```
ssh: Could not resolve hostname
```

Kemungkinan penyebab: Kesalahan ini biasanya terjadi ketika skrip koneksi CodeBuild sandbox yang diperlukan belum dijalankan dengan benar di lingkungan lokal Anda.

Solusi yang disarankan:

1. Unduh skrip koneksi CodeBuild kotak pasir.
2. Jalankan skrip di terminal Anda untuk membuat konfigurasi SSH yang diperlukan.
3. Coba kembali koneksi SSH Anda ke lingkungan sandbox.

Debug dibangun dengan Session Manager

Di AWS CodeBuild, Anda dapat menjeda build yang sedang berjalan dan kemudian menggunakan AWS Systems Manager Session Manager untuk menyambung ke container build dan melihat status container.

> **Note**
>
> Fitur ini tidak tersedia di lingkungan Windows.

Topik

- Prasyarat
- Jeda build
- Mulai membangun
- Connect ke container build
- Lanjutkan build

Prasyarat

Untuk mengizinkan Session Manager digunakan dengan sesi build, Anda harus mengaktifkan koneksi sesi untuk build. Ada dua prasyarat:

- CodeBuild Gambar kurasi standar Linux sudah menginstal agen SSM dan agen SSM diaktifkan. ContainerMode

  Jika Anda menggunakan gambar kustom untuk build Anda, lakukan hal berikut:

  1. Instal SSM Agent. Untuk informasi selengkapnya, lihat Instal Agen SSM secara manual pada instans EC2 untuk Linux di Panduan Pengguna. AWS Systems Manager Versi Agen SSM harus 3.0.1295.0 atau yang lebih baru.
  2. Salin file https://github.com/aws/aws-codebuild-docker-images/blob/master/ubuntu/standard/5.0/amazon-ssm-agent.jsonke `/etc/amazon/ssm/` direktori di gambar Anda. Ini memungkinkan Mode Kontainer di agen SSM.

  > **Note**
  >
  > Gambar khusus akan membutuhkan agen SSM terbaru agar fitur ini berfungsi seperti yang diharapkan.

- Peran CodeBuild layanan harus memiliki kebijakan SSM berikut:

  JSON

```
{
   "Version":"2012-10-17",
   "Statement": [
     {
       "Effect": "Allow",
       "Action": [
          "ssmmessages:CreateControlChannel",
          "ssmmessages:CreateDataChannel",
          "ssmmessages:OpenControlChannel",
          "ssmmessages:OpenDataChannel"
       ],
       "Resource": "*"
     }
   ]
```

```
}
```

CodeBuild Konsol dapat melampirkan kebijakan ini secara otomatis ke peran layanan saat memulai pembuatan. Atau, Anda dapat melampirkan kebijakan ini ke peran layanan Anda secara manual.

- Jika aktivitas sesi Audit dan logging diaktifkan di preferensi Systems Manager, peran CodeBuild layanan juga harus memiliki izin tambahan. Izin berbeda, tergantung di mana log disimpan.

  CloudWatch Log

  Jika menggunakan CloudWatch Log untuk menyimpan log Anda, tambahkan izin berikut ke peran CodeBuild layanan:

  JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "logs:DescribeLogGroups",
            "Resource": "arn:aws:logs:us-east-1:111122223333:log-group:*:*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogStream",
                "logs:PutLogEvents"
            ],
            "Resource": "arn:aws:logs:us-east-1:111122223333:log-
group:MyLogGroup:*"
        }
    ]
}
```

  Amazon S3

  Jika menggunakan Amazon S3 untuk menyimpan log Anda, tambahkan izin berikut ke peran CodeBuild layanan:

  JSON

```
{
```

```
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetEncryptionConfiguration",
        "s3:PutObject"
      ],
      "Resource": [
        "arn:aws:s3:::<bucket-name>",
        "arn:aws:s3:::<bucket-name>/*"
      ]
    }
  ]
}
```

Untuk informasi selengkapnya, lihat Aktivitas sesi audit dan pencatatan di Panduan AWS Systems Manager Pengguna.

Jeda build

Untuk menjeda build, masukkan codebuild-breakpoint perintah di salah satu fase build di file buildspec Anda. Build akan dijeda pada titik ini, yang memungkinkan Anda terhubung ke container build dan melihat container dalam statusnya saat ini.

Misalnya, tambahkan yang berikut ini ke fase build di file buildspec Anda.

```
phases:
  pre_build:
    commands:
      - echo Entered the pre_build phase...
      - echo "Hello World" > /tmp/hello-world
      - codebuild-breakpoint
```

Kode ini membuat `/tmp/hello-world` file dan kemudian menjeda build pada saat ini.

Mulai membangun

Agar Session Manager dapat digunakan dengan sesi build, Anda harus mengaktifkan koneksi sesi untuk build. Untuk melakukan ini, saat memulai build, ikuti langkah-langkah berikut:

1. Buka AWS CodeBuild konsol di https://console.aws.amazon.com/codesuite/codebuild/home.
2. Di panel navigasi, pilih Membangun proyek. Pilih proyek build, lalu pilih Start build with overrides.
3. Pilih Penggantian build lanjutan.
4. Di bagian Lingkungan, pilih opsi Aktifkan koneksi sesi. Jika opsi ini tidak dipilih, semua codebuild-resume perintah codebuild-breakpoint dan diabaikan.
5. Buat perubahan lain yang diinginkan, dan pilih Mulai membangun.
6. Pantau status build di konsol. Saat sesi tersedia, tautan Pengelola AWS Sesi akan muncul di bagian Status build.

Connect ke container build

Anda dapat terhubung ke kontainer build dengan salah satu dari dua cara:

CodeBuild konsol

Di browser web, buka tautan AWS Session Manager untuk terhubung ke container build. Sesi terminal terbuka yang memungkinkan Anda menelusuri dan mengontrol container build.

AWS CLI

> **Note**
>
> Mesin lokal Anda harus menginstal plugin Session Manager untuk prosedur ini. Untuk informasi selengkapnya, lihat Menginstal Plugin Session Manager untuk AWS CLI di AWS Systems Manager Panduan Pengguna.

1. Panggil batch-get-builds api dengan ID build untuk mendapatkan informasi tentang build, termasuk pengenal target sesi. Nama properti pengidentifikasi target sesi bervariasi tergantung pada jenis output `aws` perintah. Inilah `--output json` sebabnya mengapa ditambahkan ke perintah.

   ```
   aws codebuild batch-get-builds --ids <buildID> --region <region> --output json
   ```

2. Salin nilai `sessionTarget` properti. Nama `sessionTarget` properti dapat bervariasi tergantung pada jenis output `aws` perintah. Inilah `--output json` sebabnya mengapa ditambahkan ke perintah di langkah sebelumnya.
3. Gunakan perintah berikut untuk menyambung ke container build.

```
aws ssm start-session --target <sessionTarget> --region <region>
```

Untuk contoh ini, verifikasi bahwa `/tmp/hello-world` file itu ada dan berisi teks`Hello World`.

Lanjutkan build

Setelah Anda selesai memeriksa container build, keluarkan codebuild-resume perintah dari shell container.

```
$ codebuild-resume
```

Hapus build di AWS CodeBuild

Anda dapat menggunakan AWS CLI atau AWS SDK untuk menghapus build in. AWS CodeBuild

Topik

- Hapus build ()AWS CLI
- Hapus build (AWS SDK)

Hapus build ()AWS CLI

Jalankan perintah `batch-delete-builds`:

```
aws codebuild batch-delete-builds --ids ids
```

Pada perintah sebelumnya, ganti placeholder berikut:

- *`ids`*: String yang dibutuhkan. ID build yang akan dihapus. Untuk menentukan beberapa build, pisahkan setiap ID build dengan spasi. Untuk mendapatkan daftar ID build, lihat topik berikut:
  - Melihat daftar ID build (AWS CLI)
  - Melihat daftar ID build untuk proyek build (AWS CLI)

Jika berhasil, `buildsDeleted` array akan muncul di output, yang berisi Amazon Resource Name (ARN) dari setiap build yang berhasil dihapus. Informasi tentang build yang tidak berhasil dihapus muncul dalam output dalam `buildsNotDeleted` array.

Misalnya, jika Anda menjalankan perintah ini:

```
aws codebuild batch-delete-builds --ids my-demo-build-project:f8b888d2-5e1e-4032-8645-
b115195648EX my-other-demo-build-project:a18bc6ee-e499-4887-b36a-8c90349c7eEX
```

Informasi yang mirip dengan berikut ini muncul di output:

```
{
  "buildsNotDeleted": [
    {
      "id": "arn:aws:codebuild:us-west-2:123456789012:build/my-demo-build-
project:f8b888d2-5e1e-4032-8645-b115195648EX",
      "statusCode": "BUILD_IN_PROGRESS"
    }
  ],
  "buildsDeleted": [
    "arn:aws:codebuild:us-west-2:123456789012:build/my-other-demo-build-
project:a18bc6ee-e499-4887-b36a-8c90349c7eEX"
  ]
}
```

Hapus build (AWS SDK)

Untuk informasi tentang penggunaan AWS CodeBuild dengan AWS SDK, lihat. AWS Referensi SDK dan alat

Coba ulang membangun secara manual AWS CodeBuild

Anda dapat menggunakan AWS CodeBuild konsol, AWS CLI, atau AWS SDK untuk mencoba lagi secara manual baik build tunggal atau build batch. AWS CodeBuild

Topik

- Coba ulang build secara manual (konsol)
- Coba ulang build secara manual (AWS CLI)
- Coba ulang build secara manual (AWS SDK)

Coba ulang build secara manual (konsol)

1. Buka AWS CodeBuild konsol di https://console.aws.amazon.com/codesuite/codebuild/home.

2. Lakukan salah satu tindakan berikut:

   - Jika *build-ID* halaman *build-project-name*: ditampilkan, pilih Retry build.
   - Di panel navigasi, pilih Build history. Dalam daftar build, pilih kotak untuk build, lalu pilih Retry build.
   - Di panel navigasi, pilih Bangun proyek. Dalam daftar proyek build, di kolom Nama, pilih tautan untuk nama proyek build. Dalam daftar build, pilih kotak untuk build, lalu pilih Retry build.

> **Note**
>
> Secara default, hanya 100 build atau proyek build terbaru yang ditampilkan. Untuk melihat lebih banyak build atau proyek build, pilih ikon roda gigi, lalu pilih nilai yang berbeda untuk Build per halaman atau Proyek per halaman atau gunakan panah mundur dan maju.

Coba ulang build secara manual (AWS CLI)

- Jalankan perintah retry-build:

```
aws codebuild retry-build --id <build-id> --idempotency-token <idempotencyToken>
```

Pada perintah sebelumnya, ganti placeholder berikut:

- *<build-id>*: String yang diperlukan. ID build atau batch build untuk dicoba lagi. Untuk mendapatkan daftar ID build, lihat topik berikut:
  - Melihat daftar ID build (AWS CLI)
  - Melihat daftar ID build batch (AWS CLI)
  - Melihat daftar ID build untuk proyek build (AWS CLI)
  - Lihat daftar ID build batch untuk proyek build (AWS CLI)
- `--idempotency-token`: Opsional. Jika Anda menjalankan retry-build perintah dengan opsi tersebut, pengenal unik yang peka huruf besar-kecil, atau token, disertakan dengan `retry-build` permintaan. Token berlaku selama 5 menit setelah permintaan. Jika Anda mengulangi `retry-build` permintaan dengan token yang sama, tetapi mengubah parameter, CodeBuild mengembalikan kesalahan ketidakcocokan parameter.

Coba ulang build secara manual (AWS SDK)

Untuk informasi selengkapnya tentang penggunaan AWS CodeBuild dengan AWS SDK, lihat. AWS Referensi SDK dan alat

Coba lagi membangun secara otomatis di AWS CodeBuild

Anda dapat menggunakan AWS CodeBuild konsol, AWS CLI, atau AWS SDK untuk mencoba ulang build Anda secara otomatis. AWS CodeBuild Dengan auto-retry diaktifkan, secara otomatis CodeBuild akan memanggil `RetryBuild` menggunakan peran layanan proyek setelah gagal membangun hingga batas yang ditentukan. Misalnya, jika batas coba ulang otomatis disetel ke dua, CodeBuild akan memanggil `RetryBuild` API untuk mencoba ulang build Anda secara otomatis hingga dua kali tambahan.

> **Note**
>
> CodeBuild tidak mendukung percobaan ulang otomatis untuk. CodePipeline

Topik

- Coba lagi build secara otomatis (konsol)
- Coba lagi build secara otomatis ()AWS CLI
- Coba ulang build (AWS SDK) secara otomatis

Coba lagi build secara otomatis (konsol)

1. Buka AWS CodeBuild konsol di https://console.aws.amazon.com/codesuite/codebuild/home.
2. Pilih Buat proyek. Untuk informasi selengkapnya, lihat Buat proyek build (konsol) dan Jalankan build (konsol).
   - Di Lingkungan:
     - Untuk Auto-retry limit, masukkan jumlah maksimum percobaan ulang otomatis yang diinginkan setelah build gagal.
3. Di Lingkungan, pilih Konfigurasi tambahan.
4. Lanjutkan dengan nilai default dan kemudian pilih Create build project.

Coba lagi build secara otomatis ()AWS CLI

- Jalankan perintah create-project:

```
aws codebuild create-project \
    --name "<project-name>" \
    --auto-retry-limit <auto-retry-limit> \
    --source "<source>" \
    --artifacts {<artifacts>} \
    --environment "{\"type\": \"environment-type>\",\"image\": \"image-type>\",
\"computeType\": \"compute-type>\"}" \
    --service-role "service-role>"
```

  Pada perintah sebelumnya, ganti placeholder berikut:

  - *<auto-retry-limit>*: Setel batas coba ulang otomatis ke jumlah maksimum percobaan ulang otomatis yang diinginkan setelah build gagal.
  - *<project-name>*,*<source>*,*<artifacts>*,*environment-type>*,, *image-type>compute-type>*, dan*service-role>*: Tetapkan pengaturan konfigurasi proyek yang Anda inginkan.

Coba ulang build (AWS SDK) secara otomatis

Untuk informasi selengkapnya tentang penggunaan AWS CodeBuild dengan AWS SDK, lihat. AWS Referensi SDK dan alat

Berhenti membangun AWS CodeBuild

Anda dapat menggunakan AWS CodeBuild konsol, AWS CLI, atau AWS SDK untuk menghentikan build in AWS CodeBuild.

Topik

- Menghentikan build (konsol)
- Hentikan membangun (AWS CLI)
- Hentikan membangun (AWS SDK)

Menghentikan build (konsol)

1. Buka AWS CodeBuild konsol di https://console.aws.amazon.com/codesuite/codebuild/home.
2. Lakukan salah satu tindakan berikut:
   - Jika *build-ID* halaman *build-project-name*: ditampilkan, pilih Stop build.
   - Di panel navigasi, pilih Build history. Dalam daftar build, pilih kotak untuk build, lalu pilih Stop build.
   - Di panel navigasi, pilih Bangun proyek. Dalam daftar proyek build, di kolom Nama, pilih tautan untuk nama proyek build. Dalam daftar build, pilih kotak untuk build, lalu pilih Stop build.

> **Note**
>
> Secara default, hanya 100 build atau proyek build terbaru yang ditampilkan. Untuk melihat lebih banyak build atau proyek build, pilih ikon roda gigi, lalu pilih nilai yang berbeda untuk Build per halaman atau Proyek per halaman atau gunakan panah mundur dan maju. Jika AWS CodeBuild tidak berhasil menghentikan build (misalnya, jika proses pembuatan sudah selesai), tombol Stop dinonaktifkan atau mungkin tidak muncul.

Hentikan membangun (AWS CLI)

- Jalankan perintah stop-build:

```
aws codebuild stop-build --id id
```

Pada perintah sebelumnya, ganti placeholder berikut:

- *id*: String yang diperlukan. ID build yang akan dihentikan. Untuk mendapatkan daftar ID build, lihat topik berikut:
  - Melihat daftar ID build (AWS CLI)
  - Melihat daftar ID build untuk proyek build (AWS CLI)

Jika AWS CodeBuild berhasil menghentikan build, `buildStatus` nilai dalam `build` objek dalam output adalah`STOPPED`.

Jika CodeBuild tidak berhasil menghentikan build (misalnya, jika build sudah selesai), `buildStatus` nilai dalam `build` objek dalam output adalah status build akhir (misalnya,`SUCCEEDED`).

Hentikan membangun (AWS SDK)

Untuk informasi selengkapnya tentang penggunaan AWS CodeBuild dengan AWS SDK, lihat. AWS Referensi SDK dan alat

Hentikan pembuatan batch AWS CodeBuild

Anda dapat menggunakan AWS CodeBuild konsol, AWS CLI, atau AWS SDK untuk menghentikan pembuatan batch. AWS CodeBuild

> **Note**
>
> Jika Anda menggunakan komputasi Lambda dalam build batch Anda, pembuatan Lambda yang sedang berlangsung tidak dapat dihentikan.

Topik

- Menghentikan pembuatan batch (konsol)
- Hentikan pembuatan batch (AWS CLI)
- Hentikan pembuatan batch (AWS SDK)

Menghentikan pembuatan batch (konsol)

1. Buka AWS CodeBuild konsol di https://console.aws.amazon.com/codesuite/codebuild/home.
2. Lakukan salah satu tindakan berikut:
   - Jika *`build-ID`* halaman *`build-project-name`*: ditampilkan, pilih Stop build.
   - Di panel navigasi, pilih Build history. Dalam daftar build, pilih kotak untuk build, lalu pilih Stop build.

- Di panel navigasi, pilih Bangun proyek. Dalam daftar proyek build, di kolom Nama, pilih tautan untuk nama proyek build. Dalam daftar build, pilih kotak untuk build, lalu pilih Stop build.

> **Note**
>
> Secara default, hanya 100 build atau proyek build terbaru yang ditampilkan. Untuk melihat lebih banyak build atau proyek build, pilih ikon roda gigi, lalu pilih nilai yang berbeda untuk Build per halaman atau Proyek per halaman atau gunakan panah mundur dan maju.

Hentikan pembuatan batch (AWS CLI)

- Jalankan perintah `stop-build-batch`:

```
aws codebuild stop-build-batch --id <batch-build-id>
```

Pada perintah sebelumnya, ganti placeholder berikut:

- *<batch-build-id>*: String yang diperlukan. Pengidentifikasi batch build untuk berhenti. Untuk mendapatkan daftar pengenal build batch, lihat topik berikut:
  - Melihat daftar ID build batch (AWS CLI)
  - Lihat daftar ID build batch untuk proyek build (AWS CLI)

Hentikan pembuatan batch (AWS SDK)

Untuk informasi selengkapnya tentang penggunaan AWS CodeBuild dengan AWS SDK, lihat. AWS Referensi SDK dan alat

Pemicu AWS CodeBuild dibangun secara otomatis

Anda dapat membuat pemicu pada proyek untuk menjadwalkan build sekali setiap jam, hari, atau minggu. Anda juga dapat mengedit pemicu untuk menggunakan aturan khusus dengan ekspresi CloudWatch cron Amazon. Misalnya, menggunakan ekspresi cron, Anda dapat menjadwalkan build pada waktu tertentu setiap hari kerja. Untuk informasi tentang membuat dan mengedit pemicu, lihat Buat AWS CodeBuild pemicu danEdit AWS CodeBuild pemicu.

Topik

- Buat AWS CodeBuild pemicu
- Edit AWS CodeBuild pemicu

Buat AWS CodeBuild pemicu

Anda dapat membuat pemicu pada proyek untuk menjadwalkan build sekali setiap jam, hari, atau minggu. Anda juga dapat membuat pemicu menggunakan aturan khusus dengan ekspresi CloudWatch cron Amazon. Misalnya, menggunakan ekspresi cron, Anda dapat menjadwalkan build pada waktu tertentu setiap hari kerja.

> **Note**
>
> Tidak mungkin memulai build batch dari pemicu build, EventBridge peristiwa Amazon, atau AWS Step Functions tugas.

Topik

- Buat AWS CodeBuild pemicu (konsol)
- Buat AWS CodeBuild pemicu secara terprogram

Buat AWS CodeBuild pemicu (konsol)

Gunakan prosedur berikut untuk membuat pemicu menggunakan. Konsol Manajemen AWS

Untuk membuat pemicu

1. Buka AWS CodeBuild konsol di https://console.aws.amazon.com/codesuite/codebuild/home.
2. Di panel navigasi, pilih Membangun proyek.
3. Pilih tautan untuk proyek build yang ingin Anda tambahkan pemicunya, lalu pilih tab Pemicu build.

> **Note**
>
> Secara default, 100 proyek build terbaru ditampilkan. Untuk melihat proyek build lainnya, pilih ikon roda gigi, lalu pilih nilai berbeda untuk Proyek per halaman atau gunakan panah mundur dan maju.

4. Pilih Buat pemicu.
5. Masukkan nama di nama Pemicu.
6. Dari daftar drop-down Frekuensi, pilih frekuensi untuk pemicu Anda. Jika Anda ingin membuat frekuensi menggunakan ekspresi cron, pilih Kustom.
7. Tentukan parameter untuk frekuensi pemicu Anda. Anda dapat memasukkan beberapa karakter pertama pilihan Anda di kotak teks untuk memfilter item menu tarik-turun.

> **Note**
>
> Jam dan menit mulai berbasis nol. Menit awal adalah angka antara nol dan 59. Jam awal adalah angka antara nol dan 23. Misalnya, pemicu harian yang dimulai setiap hari pukul 12:15 P.M. memiliki jam mulai 12 dan menit mulai 15. Pemicu harian yang dimulai setiap hari pada tengah malam memiliki jam awal nol dan menit awal nol. Pemicu harian yang dimulai setiap hari pada pukul 11:59 P.M. memiliki jam mulai 23 dan menit mulai 59.

Frekuensi	Parameter yang Diperlukan	Detail
Per Jam	Mulai menit	Gunakan menu tarik-turun menit Mulai.
Harian	Mulai menit Jam mulai	Gunakan menu tarik-turun menit Mulai. Gunakan menu drop-down Start hour.
Mingguan	Mulai menit Jam mulai	Gunakan menu tarik-turun menit Mulai.

Frekuensi	Parameter yang Diperlukan	Detail
	Mulai hari	Gunakan menu drop-down Start hour. Gunakan menu drop-down Mulai hari.
Khusus	Ekspresi cron	Masukkan ekspresi cron dalam ekspresi Cron. Ekspresi cron memiliki enam bidang wajib yang dipisahkan oleh spasi putih. Bidang menentukan nilai awal untuk menit, jam, hari bulan, bulan, hari dalam seminggu, dan tahun. Anda dapat menggunakan wildcard untuk menentukan rentang, nilai tambahan, dan lainnya. Misalnya, ekspresi cron **`0 9 ? * MON-FRI *`** menjadwalkan build setiap hari kerja pada pukul 9:00. A.M Untuk informasi selengkapnya, lihat Ekspresi Cron di Panduan Pengguna CloudWatch Acara Amazon.

8. Pilih Aktifkan pemicu ini.
9. (Opsional) Perluas bagian Lanjutan. Dalam versi Sumber, ketik versi sumber Anda.

   - Untuk Amazon S3, masukkan ID versi yang sesuai dengan versi artefak input yang ingin Anda buat. Jika versi Sumber dibiarkan kosong, versi terbaru digunakan.
   - Untuk AWS CodeCommit, ketik ID komit. Jika versi Sumber dibiarkan kosong, ID komit HEAD cabang default digunakan.
   - Untuk GitHub atau GitHub Enterprise, ketikkan ID komit, ID permintaan tarik, nama cabang, atau nama tag yang sesuai dengan versi kode sumber yang ingin Anda buat. Jika Anda

menentukan ID permintaan tarik, itu harus menggunakan format `pr/pull-request-ID` (misalnya,`pr/25`). Jika Anda menentukan nama cabang, ID komit HEAD cabang digunakan. Jika versi Source kosong, ID komit HEAD cabang default digunakan.

- Untuk Bitbucket, ketik ID komit, nama cabang, atau nama tag yang sesuai dengan versi kode sumber yang ingin Anda buat. Jika Anda menentukan nama cabang, ID komit HEAD cabang digunakan. Jika versi Source kosong, ID komit HEAD cabang default digunakan.

10. (Opsional) Tentukan batas waktu antara 5 menit dan 2160 menit (36 jam). Nilai ini menentukan berapa lama AWS CodeBuild mencoba membangun sebelum berhenti. Jika Jam dan Menit dibiarkan kosong, nilai batas waktu default yang ditentukan dalam proyek digunakan.
11. Pilih Buat pemicu.

Buat AWS CodeBuild pemicu secara terprogram

CodeBuild menggunakan EventBridge aturan Amazon untuk pemicu build. Anda dapat menggunakan EventBridge API untuk membuat pemicu build secara terprogram untuk proyek Anda. CodeBuild Lihat Referensi Amazon EventBridge API untuk informasi selengkapnya.

Edit AWS CodeBuild pemicu

Anda dapat mengedit pemicu pada proyek untuk menjadwalkan build setiap jam, hari, atau minggu sekali. Anda juga dapat mengedit pemicu untuk menggunakan aturan khusus dengan ekspresi CloudWatch cron Amazon. Misalnya, menggunakan ekspresi cron, Anda dapat menjadwalkan build pada waktu tertentu setiap hari kerja. Untuk informasi tentang membuat pemicu, lihatBuat AWS CodeBuild pemicu.

Topik

- Edit AWS CodeBuild pemicu (konsol)
- Edit AWS CodeBuild pemicu secara terprogram

Edit AWS CodeBuild pemicu (konsol)

Gunakan prosedur berikut untuk mengedit pemicu menggunakan. Konsol Manajemen AWS

Untuk mengedit pemicu

1. Buka AWS CodeBuild konsol di https://console.aws.amazon.com/codesuite/codebuild/home.

2. Di panel navigasi, pilih Membangun proyek.
3. Pilih link untuk proyek build yang ingin diubah, lalu pilih tab Build triggers.

> **Note**
>
> Secara default, 100 proyek build terbaru ditampilkan. Untuk melihat proyek build lainnya, pilih ikon roda gigi, lalu pilih nilai berbeda untuk Proyek per halaman atau gunakan panah mundur dan maju.

4. Pilih tombol radio di sebelah pemicu yang ingin Anda ubah, lalu pilih Edit.
5. Dari daftar drop-down Frekuensi, pilih frekuensi untuk pemicu Anda. Jika Anda ingin membuat frekuensi menggunakan ekspresi cron, pilih Kustom.
6. Tentukan parameter untuk frekuensi pemicu Anda. Anda dapat memasukkan beberapa karakter pertama pilihan Anda di kotak teks untuk memfilter item menu drop-down.

> **Note**
>
> Jam dan menit mulai berbasis nol. Menit awal adalah angka antara nol dan 59. Jam awal adalah angka antara nol dan 23. Misalnya, pemicu harian yang dimulai setiap hari pukul 12:15 P.M. memiliki jam mulai 12 dan menit mulai 15. Pemicu harian yang dimulai setiap hari pada tengah malam memiliki jam awal nol dan menit awal nol. Pemicu harian yang dimulai setiap hari pada pukul 11:59 P.M. memiliki jam mulai 23 dan menit mulai 59.

Frekuensi	Parameter yang Diperlukan	Detail
Per Jam	Mulai menit	Gunakan menu tarik-turun menit Mulai.
Harian	Mulai menit Jam mulai	Gunakan menu tarik-turun menit Mulai. Gunakan menu drop-down Start hour.
Mingguan	Mulai menit Jam mulai	Gunakan menu tarik-turun menit Mulai.

Frekuensi	Parameter yang Diperlukan	Detail
	Mulai hari	Gunakan menu drop-down Start hour. Gunakan menu drop-down Mulai hari.
Khusus	Ekspresi cron	Masukkan ekspresi cron dalam ekspresi Cron. Ekspresi cron memiliki enam bidang wajib yang dipisahkan oleh spasi putih. Bidang menentukan nilai awal untuk menit, jam, hari bulan, bulan, hari dalam seminggu, dan tahun. Anda dapat menggunakan wildcard untuk menentukan rentang, nilai tambahan, dan lainnya. Misalnya, ekspresi cron **`0 9 ? * MON-FRI *`** menjadwalkan build setiap hari kerja pukul 9:00. A.M Untuk informasi selengkapnya, lihat Ekspresi Cron di Panduan Pengguna CloudWatch Acara Amazon.

7. Pilih Aktifkan pemicu ini.

> **Note**
>
> Anda dapat menggunakan CloudWatch konsol Amazon di https://console.aws.amazon.com/cloudwatch/untuk mengedit versi sumber, batas waktu, dan opsi lain yang tidak tersedia. AWS CodeBuild

Edit AWS CodeBuild pemicu secara terprogram

CodeBuild menggunakan EventBridge aturan Amazon untuk pemicu build. Anda dapat menggunakan EventBridge API untuk mengedit pemicu build secara terprogram untuk proyek Anda. CodeBuild Lihat Referensi Amazon EventBridge API untuk informasi selengkapnya.

Lihat detail build di AWS CodeBuild

Anda dapat menggunakan AWS CodeBuild konsol AWS CLI, atau AWS SDK untuk melihat detail tentang build yang dikelola oleh. CodeBuild

Topik

- Lihat detail build (konsol)
- Lihat detail build (AWS CLI)
- Lihat detail build (AWS SDK)
- Bangun transisi fase

Lihat detail build (konsol)

1. Buka AWS CodeBuild konsol di https://console.aws.amazon.com/codesuite/codebuild/home.
2. Lakukan salah satu tindakan berikut:
   - Di panel navigasi, pilih Build history. Dalam daftar build, di kolom Build run, pilih link untuk build.
   - Di panel navigasi, pilih Membangun proyek. Dalam daftar proyek build, di kolom Name, pilih link untuk nama proyek build. Kemudian, dalam daftar build, di kolom Build run, pilih link untuk build.

> **Note**
>
> Secara default, hanya 10 proyek build atau build terbaru yang ditampilkan. Untuk melihat lebih banyak proyek build atau build, pilih ikon roda gigi, lalu pilih nilai yang berbeda untuk Build per halaman atau Proyek per halaman atau gunakan panah mundur dan maju.

Lihat detail build (AWS CLI)

Untuk informasi lebih lanjut tentang menggunakan AWS CLI with AWS CodeBuild, lihatReferensi baris perintah.

Jalankan perintah batch-get-builds:

```
aws codebuild batch-get-builds --ids ids
```

Ganti placeholder berikut:

- *ids*: String yang dibutuhkan. Satu atau beberapa ID build untuk melihat detailnya. Untuk menentukan lebih dari satu ID build, pisahkan setiap ID build dengan spasi. Anda dapat menentukan hingga 100 ID build. Untuk mendapatkan daftar ID build, lihat topik berikut:
  - Melihat daftar ID build (AWS CLI)
  - Melihat daftar ID build untuk proyek build (AWS CLI)

Misalnya, jika Anda menjalankan perintah ini:

```
aws codebuild batch-get-builds --ids codebuild-demo-project:e9c4f4df-3f43-41d2-
ab3a-60fe2EXAMPLE codebuild-demo-project:815e755f-bade-4a7e-80f0-efe51EXAMPLE my-other-
project:813bb6c6-891b-426a-9dd7-6d8a3EXAMPLE
```

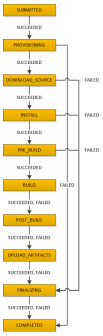
Jika perintah berhasil, data yang mirip dengan yang dijelaskan dalam Untuk melihat informasi build yang diringkas muncul di output.

Lihat detail build (AWS SDK)

Untuk informasi selengkapnya tentang penggunaan AWS CodeBuild dengan AWS SDK, lihat. AWS Referensi SDK dan alat

Bangun transisi fase

Dibangun dalam AWS CodeBuild proses secara bertahap:



> **Important**
>
> `UPLOAD_ARTIFACTS`Fase selalu dicoba, bahkan jika `BUILD` fase gagal.

Melihat daftar ID build di AWS CodeBuild

Anda dapat menggunakan AWS CodeBuild konsol AWS CLI, atau AWS SDK untuk melihat daftar ID build untuk build yang dikelola oleh. CodeBuild

Topik

- Melihat daftar ID build (konsol)
- Melihat daftar ID build (AWS CLI)
- Melihat daftar ID build batch (AWS CLI)
- Melihat daftar ID build (AWS SDK)

Melihat daftar ID build (konsol)

1. Buka AWS CodeBuild konsol di https://console.aws.amazon.com/codesuite/codebuild/home.
2. Di panel navigasi, pilih Build history.

> **Note**
>
> Secara default, hanya 10 build terbaru yang ditampilkan. Untuk melihat lebih banyak build, pilih ikon roda gigi, lalu pilih nilai yang berbeda untuk Build per halaman atau gunakan panah mundur dan maju.

Melihat daftar ID build (AWS CLI)

Untuk informasi lebih lanjut tentang menggunakan AWS CLI with CodeBuild, lihatReferensi baris perintah.

- Jalankan perintah list-builds:

```
aws codebuild list-builds --sort-order sort-order --next-token next-token
```

Pada perintah sebelumnya, ganti placeholder berikut:

- *sort-order*: String opsional yang digunakan untuk menunjukkan cara membuat daftar ID build. Nilai yang valid mencakup ASCENDING dan DESCENDING.
- *next-token*: String opsional. Selama proses sebelumnya, jika ada lebih dari 100 item dalam daftar, hanya 100 item pertama yang dikembalikan, bersama dengan string unik yang disebut token berikutnya. Untuk mendapatkan kumpulan item berikutnya dalam daftar, jalankan perintah ini lagi, tambahkan token berikutnya ke panggilan. Untuk mendapatkan semua item dalam daftar, terus jalankan perintah ini dengan setiap token berikutnya berikutnya, sampai tidak ada lagi token berikutnya yang dikembalikan.

Misalnya, jika Anda menjalankan perintah ini:

```
aws codebuild list-builds --sort-order ASCENDING
```

Hasil yang mirip dengan berikut ini mungkin muncul di output:

```
{
  "nextToken": "4AEA6u7J...The full token has been omitted for brevity...MzY2OA==",
  "ids": [
    "codebuild-demo-project:815e755f-bade-4a7e-80f0-efe51EXAMPLE"
    "codebuild-demo-project:84a7f3d1-d40e-4956-b4cf-7a9d4EXAMPLE"
    ... The full list of build IDs has been omitted for brevity ...
    "codebuild-demo-project:931d0b72-bf6f-4040-a472-5c707EXAMPLE"
  ]
}
```

Jika Anda menjalankan perintah ini lagi:

```
aws codebuild list-builds --sort-order ASCENDING --next-token 4AEA6u7J...The full
 token has been omitted for brevity...MzY2OA==
```

Hasil yang mirip dengan berikut ini mungkin muncul di output:

```
{
  "ids": [
    "codebuild-demo-project:49015049-21cf-4b50-9708-df115EXAMPLE",
    "codebuild-demo-project:543e7206-68a3-46d6-a4da-759abEXAMPLE",
    ... The full list of build IDs has been omitted for brevity ...
    "codebuild-demo-project:c282f198-4582-4b38-bdc0-26f96EXAMPLE"
  ]
}
```

Melihat daftar ID build batch (AWS CLI)

Untuk informasi lebih lanjut tentang menggunakan AWS CLI with CodeBuild, lihatReferensi baris perintah.

- Jalankan perintah list-build-batches:

```
aws codebuild list-build-batches --sort-order sort-order --next-token next-token
```

Pada perintah sebelumnya, ganti placeholder berikut:

- *sort-order*: String opsional yang digunakan untuk menunjukkan cara membuat daftar ID build batch. Nilai yang valid mencakup `ASCENDING` dan `DESCENDING`.
- *next-token*: String opsional. Selama proses sebelumnya, jika ada lebih dari 100 item dalam daftar, hanya 100 item pertama yang dikembalikan, bersama dengan string unik yang disebut token berikutnya. Untuk mendapatkan kumpulan item berikutnya dalam daftar, jalankan perintah ini lagi, tambahkan token berikutnya ke panggilan. Untuk mendapatkan semua item dalam daftar, terus jalankan perintah ini dengan setiap token berikutnya berikutnya, sampai tidak ada lagi token berikutnya yang dikembalikan.

Misalnya, jika Anda menjalankan perintah ini:

```
aws codebuild list-build-batches --sort-order ASCENDING
```

Hasil yang mirip dengan berikut ini mungkin muncul di output:

```
{
  "nextToken": "4AEA6u7J...The full token has been omitted for brevity...MzY2OA==",
  "ids": [
    "codebuild-demo-project:815e755f-bade-4a7e-80f0-efe51EXAMPLE"
    "codebuild-demo-project:84a7f3d1-d40e-4956-b4cf-7a9d4EXAMPLE"
    ... The full list of build IDs has been omitted for brevity ...
    "codebuild-demo-project:931d0b72-bf6f-4040-a472-5c707EXAMPLE"
  ]
}
```

Jika Anda menjalankan perintah ini lagi:

```
aws codebuild list-build-batches --sort-order ASCENDING --next-token 4AEA6u7J...The
 full token has been omitted for brevity...MzY2OA==
```

Hasil yang mirip dengan berikut ini mungkin muncul di output:

```
{
  "ids": [
    "codebuild-demo-project:49015049-21cf-4b50-9708-df115EXAMPLE",
    "codebuild-demo-project:543e7206-68a3-46d6-a4da-759abEXAMPLE",
    ... The full list of build IDs has been omitted for brevity ...
    "codebuild-demo-project:c282f198-4582-4b38-bdc0-26f96EXAMPLE"
  ]
}
```

Melihat daftar ID build (AWS SDK)

Untuk informasi selengkapnya tentang penggunaan CodeBuild dengan AWS SDK, lihat. AWS Referensi SDK dan alat

Melihat daftar ID build untuk proyek build di AWS CodeBuild

Anda dapat menggunakan AWS CodeBuild konsol, AWS CLI, atau AWS SDK untuk melihat daftar ID build untuk proyek build di CodeBuild.

Topik

- Melihat daftar ID build untuk proyek build (konsol)
- Melihat daftar ID build untuk proyek build (AWS CLI)
- Lihat daftar ID build batch untuk proyek build (AWS CLI)
- Melihat daftar ID build untuk proyek build (AWS SDK)

Melihat daftar ID build untuk proyek build (konsol)

1. Buka CodeBuild konsol di https://console.aws.amazon.com/codebuild/.
2. Di panel navigasi, pilih Bangun proyek. Dalam daftar proyek build, di kolom Nama, pilih proyek build.

> **Note**
>
> Secara default, hanya 100 build atau proyek build terbaru yang ditampilkan. Untuk melihat lebih banyak build atau proyek build, pilih ikon roda gigi, lalu pilih nilai yang berbeda untuk Build per halaman atau Proyek per halaman atau gunakan panah mundur dan maju.

Melihat daftar ID build untuk proyek build (AWS CLI)

Untuk informasi lebih lanjut tentang menggunakan AWS CLI with AWS CodeBuild, lihatReferensi baris perintah.

Jalan list-builds-for-project kan perintah, sebagai berikut:

```
aws codebuild list-builds-for-project --project-name project-name --sort-order sort-order --next-token next-token
```

Pada perintah sebelumnya, ganti placeholder berikut:

- *project-name*: String yang diperlukan digunakan untuk menunjukkan nama proyek build untuk mencantumkan ID build. Untuk mendapatkan daftar proyek pembangunan, lihatLihat daftar nama proyek build (AWS CLI).
- *sort-order*: String opsional digunakan untuk menunjukkan cara membuat daftar ID build. Nilai yang valid mencakup ASCENDING dan DESCENDING.
- *next-token*: String opsional. Selama proses sebelumnya, jika ada lebih dari 100 item dalam daftar, hanya 100 item pertama yang dikembalikan, bersama dengan string unik yang disebut token berikutnya. Untuk mendapatkan batch item berikutnya dalam daftar, jalankan perintah ini lagi, tambahkan token berikutnya ke panggilan. Untuk mendapatkan semua item dalam daftar, terus jalankan perintah ini dengan setiap token berikutnya yang dikembalikan, sampai tidak ada lagi token berikutnya yang dikembalikan.

Misalnya, jika Anda menjalankan perintah yang mirip dengan ini:

```
aws codebuild list-builds-for-project --project-name codebuild-demo-project --sort-
order ASCENDING
```

Hasil seperti berikut mungkin muncul di output:

```
{
  "nextToken": "4AEA6u7J...The full token has been omitted for brevity...MzY2OA==",
  "ids": [
    "codebuild-demo-project:9b175d16-66fd-4e71-93a0-50a08EXAMPLE"
    "codebuild-demo-project:a9d1bd09-18a2-456b-8a36-7d65aEXAMPLE"
    ... The full list of build IDs has been omitted for brevity ...
    "codebuild-demo-project:fe70d102-c04f-421a-9cfa-2dc15EXAMPLE"
  ]
}
```

Jika Anda menjalankan perintah ini lagi:

```
aws codebuild list-builds-for-project --project-name codebuild-demo-project --
sort-order ASCENDING --next-token 4AEA6u7J...The full token has been omitted for
 brevity...MzY2OA==
```

Anda mungkin melihat hasil seperti berikut di output:

```
{
```

```
  "ids": [
    "codebuild-demo-project:98253670-7a8a-4546-b908-dc890EXAMPLE"
    "codebuild-demo-project:ad5405b2-1ab3-44df-ae2d-fba84EXAMPLE"
    ... The full list of build IDs has been omitted for brevity ...
    "codebuild-demo-project:f721a282-380f-4b08-850a-e0ac1EXAMPLE"
  ]
}
```

Lihat daftar ID build batch untuk proyek build (AWS CLI)

Untuk informasi lebih lanjut tentang menggunakan AWS CLI with AWS CodeBuild, lihatReferensi baris perintah.

Jalan list-build-batches-for-project kan perintah, sebagai berikut:

```
aws codebuild list-build-batches-for-project --project-name project-name --sort-
order sort-order --next-token next-token
```

Pada perintah sebelumnya, ganti placeholder berikut:

- *project-name*: String yang diperlukan digunakan untuk menunjukkan nama proyek build untuk mencantumkan ID build. Untuk mendapatkan daftar proyek pembangunan, lihatLihat daftar nama proyek build (AWS CLI).
- *sort-order*: String opsional digunakan untuk menunjukkan cara membuat daftar ID build. Nilai yang valid mencakup `ASCENDING` dan `DESCENDING`.
- *next-token*: String opsional. Selama proses sebelumnya, jika ada lebih dari 100 item dalam daftar, hanya 100 item pertama yang dikembalikan, bersama dengan string unik yang disebut token berikutnya. Untuk mendapatkan batch item berikutnya dalam daftar, jalankan perintah ini lagi, tambahkan token berikutnya ke panggilan. Untuk mendapatkan semua item dalam daftar, terus jalankan perintah ini dengan setiap token berikutnya yang dikembalikan, sampai tidak ada lagi token berikutnya yang dikembalikan.

Misalnya, jika Anda menjalankan perintah yang mirip dengan ini:

```
aws codebuild list-build-batches-for-project --project-name codebuild-demo-project --
sort-order ASCENDING
```

Hasil seperti berikut mungkin muncul di output:

```
{
  "nextToken": "4AEA6u7J...The full token has been omitted for brevity...MzY2OA==",
  "ids": [
    "codebuild-demo-project:9b175d16-66fd-4e71-93a0-50a08EXAMPLE"
    "codebuild-demo-project:a9d1bd09-18a2-456b-8a36-7d65aEXAMPLE"
    ... The full list of build IDs has been omitted for brevity ...
    "codebuild-demo-project:fe70d102-c04f-421a-9cfa-2dc15EXAMPLE"
  ]
}
```

Jika Anda menjalankan perintah ini lagi:

```
aws codebuild list-build-batches-for-project --project-name codebuild-demo-project
 --sort-order ASCENDING --next-token 4AEA6u7J...The full token has been omitted for
 brevity...MzY2OA==
```

Anda mungkin melihat hasil seperti berikut di output:

```
{
  "ids": [
    "codebuild-demo-project:98253670-7a8a-4546-b908-dc890EXAMPLE"
    "codebuild-demo-project:ad5405b2-1ab3-44df-ae2d-fba84EXAMPLE"
    ... The full list of build IDs has been omitted for brevity ...
    "codebuild-demo-project:f721a282-380f-4b08-850a-e0ac1EXAMPLE"
  ]
}
```

Melihat daftar ID build untuk proyek build (AWS SDK)

Untuk informasi selengkapnya tentang penggunaan AWS CodeBuild dengan AWS SDK, lihat. AWS Referensi SDK dan alat

Laporan pengujian di AWS CodeBuild

Anda dapat membuat laporan CodeBuild yang berisi detail tentang pengujian yang dijalankan selama build. Anda dapat membuat pengujian seperti pengujian unit, pengujian konfigurasi, dan pengujian fungsional.

Format file laporan pengujian berikut didukung:

- Mentimun JSON (.json)
- JUnit XML (.xml)
- nUnit XML (.xml)
- XML NuNiT3 (.xml)
- XML TestNG (.xml)
- Visual Studio TRX (.trx)
- Visual Studio TRX XML (.xml)

> **Note**
>
> Versi terbaru yang didukung `cucumber-js` adalah 7.3.2.

Buat kasus uji Anda dengan kerangka pengujian apa pun yang dapat membuat file laporan dalam salah satu format ini (misalnya, plugin Surefire JUnit, TestNG, atau Cucumber).

Untuk membuat laporan pengujian, Anda menambahkan nama grup laporan ke file buildspec proyek build dengan informasi tentang kasus pengujian Anda. Saat Anda menjalankan proyek build, kasus uji dijalankan dan laporan pengujian dibuat. Laporan pengujian baru dibuat di grup laporan setiap kali kasus uji dijalankan. Anda tidak perlu membuat grup laporan sebelum menjalankan pengujian. Jika Anda menentukan nama grup laporan CodeBuild , membuat grup laporan untuk Anda saat menjalankan laporan. Jika Anda ingin menggunakan grup laporan yang sudah ada, Anda menentukan ARN-nya di file buildspec.

Anda dapat menggunakan laporan pengujian untuk membantu memecahkan masalah selama proses build. Jika Anda memiliki banyak laporan pengujian dari beberapa build proyek build, Anda dapat menggunakan laporan pengujian untuk melihat tren dan tingkat pengujian dan kegagalan untuk membantu Anda mengoptimalkan build.

Laporan kedaluwarsa dalam 30 hari setelah dibuat. Anda tidak dapat melihat laporan pengujian yang kedaluwarsa. Jika Anda ingin menyimpan laporan pengujian selama lebih dari 30 hari, Anda dapat mengekspor file data mentah hasil pengujian Anda ke bucket Amazon S3. File uji yang diekspor tidak kedaluwarsa. Informasi tentang bucket S3 ditentukan saat Anda membuat grup laporan.

> **Note**
>
> Per CodeBuild an layanan yang ditentukan dalam proyek digunakan untuk izin untuk mengunggah ke bucket S3.

Topik

- Buat laporan pengujian
- Buat laporan cakupan kode
- Auto-discover laporan di CodeBuild
- Grup laporan
- Kerangka uji
- Lihat laporan pengujian
- Izin laporan uji
- Status laporan pengujian

Buat laporan pengujian

Untuk membuat laporan pengujian, Anda menjalankan proyek build yang dikonfigurasi dengan satu hingga lima grup laporan dalam file buildspec-nya. Laporan pengujian dibuat selama menjalankan. Ini berisi hasil kasus uji yang ditentukan untuk kelompok laporan. Laporan pengujian baru dibuat untuk setiap build berikutnya yang menggunakan file buildspec yang sama.

Untuk membuat laporan pengujian

1. Buat proyek pembangunan. Untuk informasi, lihat Membuat proyek pembangunan di AWS CodeBuild.
2. Konfigurasikan file buildspec proyek Anda dengan informasi laporan pengujian:

    a. Tambahkan `reports:` bagian dan tentukan ARN dari grup laporan yang ada, atau nama grup laporan.

Jika Anda menentukan ARN, CodeBuild menggunakan grup laporan itu.

Jika Anda menentukan nama, CodeBuild buat grup laporan untuk Anda menggunakan nama proyek Anda, dan nama yang Anda tentukan, dalam format *<project-name>*-*<report-group-name>*. Jika grup laporan bernama sudah ada, CodeBuild gunakan grup laporan tersebut.

b. Di bawah grup laporan, tentukan lokasi file yang berisi hasil pengujian. Jika Anda menggunakan lebih dari satu grup laporan, tentukan lokasi file hasil pengujian untuk masing-masing grup. Laporan pengujian baru dibuat setiap kali proyek build Anda berjalan. Untuk informasi selengkapnya, lihat Tentukan file pengujian.

c. Di `commands` bagian `post_build` urutan `build` atau, tentukan perintah yang menjalankan kasus pengujian yang Anda tentukan untuk grup laporan Anda. Untuk informasi selengkapnya, lihat Tentukan perintah pengujian.

Berikut ini adalah contoh bagian buildspec: `reports`

```
reports:
  php-reports:
    files:
      - "reports/php/*.xml"
    file-format: "JUNITXML"
  nunit-reports:
    files:
      - "reports/nunit/*.xml"
    file-format: "NUNITXML"
```

3. Jalankan build proyek build. Untuk informasi selengkapnya, lihat Jalankan AWS CodeBuild build secara manual.

4. Ketika build selesai, pilih build baru yang dijalankan dari Ri wayat build di halaman proyek Anda. Pilih Laporan untuk melihat laporan pengujian. Untuk informasi selengkapnya, lihat Melihat laporan pengujian untuk build.

Buat laporan cakupan kode

CodeBuild memungkinkan Anda menghasilkan laporan cakupan kode untuk pengujian Anda. Laporan cakupan kode berikut disediakan:

Cakupan baris

Cakupan baris mengukur berapa banyak pernyataan yang tercakup dalam pengujian Anda. Pernyataan adalah instruksi tunggal, tidak termasuk komentar atau syarat.

```
line coverage = (total lines covered)/(total number of lines)
```

Cakupan cabang

Cakupan cabang mengukur berapa banyak cabang yang tercakup dalam pengujian Anda dari setiap cabang di struktur kontrol, seperti pernyataan `if` atau `case`.

```
branch coverage = (total branches covered)/(total number of branches)
```

Berikut format file laporan cakupan kode yang didukung:

- JaCoCo XML-XM
- SimpleCov JSON[1]
- Clover XML
- Cobertura XML
- INFO LCOV

[1] CodeBuild menerima laporan cakupan kode JSON yang dihasilkan oleh simplecov, bukan simplecov-json.

Buat laporan cakupan kode

Untuk membuat laporan cakupan kode, Anda menjalankan proyek build yang dikonfigurasi dengan setidaknya satu grup laporan cakupan kode dalam file buildspec. CodeBuild akan menafsirkan hasil cakupan kode dan memberikan laporan cakupan kode untuk dijalankan. Laporan pengujian baru dibuat untuk setiap build berikutnya yang menggunakan file buildspec yang sama.

Untuk membuat laporan pengujian

1. Buat proyek build. Untuk informasi, lihat Membuat proyek pembangunan di AWS CodeBuild.
2. Konfigurasikan file buildspec proyek Anda dengan informasi laporan pengujian:

   a. Tambahkan `reports:` bagian dan tentukan nama untuk grup laporan Anda. CodeBuild membuat grup laporan untuk Anda menggunakan nama proyek Anda dan nama yang Anda

tentukan dalam format `project-name -report-group-name-in-buildspec`. Jika Anda sudah memiliki grup laporan yang ingin Anda gunakan, tentukan ARN-nya. Jika Anda menggunakan nama alih-alih ARN, CodeBuild buat grup laporan baru. Untuk informasi selengkapnya, lihat Reports syntax in the buildspec file.

b. Di bawah grup laporan, tentukan lokasi file yang berisi hasil cakupan kode. Jika Anda menggunakan lebih dari satu grup laporan, tentukan lokasi file hasil untuk setiap grup laporan. Laporan cakupan kode baru dibuat setiap kali proyek build Anda berjalan. Untuk informasi selengkapnya, lihat Tentukan file pengujian.

   Ini adalah contoh yang menghasilkan laporan cakupan kode untuk file hasil JaCoCo XHTML yang terletak di test-`results/jacoco-coverage-report.xml`.

```
reports:
  jacoco-report:
    files:
      - 'test-results/jacoco-coverage-report.xml'
    file-format: 'JACOCOXML'
```

c. Di `commands` bagian `post_build` urutan `build` atau, tentukan perintah yang menjalankan analisis cakupan kode. Untuk informasi selengkapnya, lihat Tentukan perintah pengujian.

3. Jalankan build proyek build. Untuk informasi selengkapnya, lihat Jalankan AWS CodeBuild build secara manual.
4. Saat build selesai, pilih build baru yang dijalankan dari riwayat Build di halaman project Anda. Pilih Laporan untuk melihat laporan cakupan kode. Lihat informasi yang lebih lengkap di Melihat laporan pengujian untuk build.

Auto-discover laporan di CodeBuild

Dengan penemuan otomatis, CodeBuild menelusuri semua file build Anda setelah fase build selesai, mencari jenis file laporan yang didukung, dan secara otomatis membuat grup dan laporan laporan cakupan pengujian dan kode baru. Untuk setiap jenis laporan yang ditemukan CodeBuild , buat grup laporan baru dengan pola berikut:

```
<project-name>-<report-file-format>-AutoDiscovered
```

> **Note**
>
> Jika file laporan yang ditemukan memiliki jenis format yang sama, mereka akan ditempatkan ke dalam grup laporan atau laporan yang sama.

Laporan pencarian otomatis dikonfigurasi oleh variabel lingkungan proyek Anda:

`CODEBUILD_CONFIG_AUTO_DISCOVER`

Variabel ini menentukan apakah deteksi otomatis laporan dinonaktifkan selama pembuatan. Secara default, pencarian otomatis laporan diaktifkan untuk semua build. Untuk menonaktifkan fitur ini, atur `CODEBUILD_CONFIG_AUTO_DISCOVER` ke`false`.

`CODEBUILD_CONFIG_AUTO_DISCOVER_DIR`

(Opsional) Variabel ini menentukan CodeBuild di mana mencari file laporan potensial. Perhatikan bahwa CodeBuild pencarian `**/*` secara default.

Variabel lingkungan ini dapat dimodifikasi selama fase build. Misalnya, jika Anda hanya ingin mengaktifkan laporan auto-discover untuk build di cabang `main` git, Anda dapat memeriksa cabang git selama proses pembuatan dan menyetel `CODEBUILD_CONFIG_AUTO_DISCOVER` ke false jika build tidak ada di `main` cabang. Pelaporan pencarian otomatis dapat dinonaktifkan menggunakan konsol atau menggunakan variabel lingkungan proyek.

Topik

- Konfigurasikan pencarian otomatis laporan menggunakan konsol
- Konfigurasikan deteksi otomatis laporan menggunakan variabel lingkungan proyek

Konfigurasikan pencarian otomatis laporan menggunakan konsol

Gunakan prosedur berikut untuk mengonfigurasi penemuan otomatis laporan menggunakan konsol.

Untuk mengonfigurasi pencarian otomatis laporan menggunakan konsol

1. Buat proyek build atau pilih proyek build untuk diedit. Untuk informasi, lihat Membuat proyek pembangunan di AWS CodeBuild atauUbah pengaturan proyek build di AWS CodeBuild.
2. Di Lingkungan, pilih Konfigur asi tambahan.

3. Untuk menonaktifkan pencarian otomatis laporan, di Laporan pencarian otomatis, pilih Non aktifkan pencarian otomatis laporan.
4. (Opsional) Di Auto-discover direktori - opsional, masukkan pola direktori CodeBuild untuk mencari file format laporan yang didukung. Perhatikan bahwa CodeBuild pencarian `**/*` secara default.

Konfigurasikan deteksi otomatis laporan menggunakan variabel lingkungan proyek

Gunakan prosedur berikut untuk mengonfigurasi penemuan otomatis laporan menggunakan variabel lingkungan proyek.

Untuk mengonfigurasi pencarian otomatis laporan menggunakan variabel lingkungan proyek

1. Buat proyek build atau pilih proyek build untuk diedit. Untuk informasi, lihat Membuat proyek pembangunan di AWS CodeBuild atauUbah pengaturan proyek build di AWS CodeBuild.
2. Di variabel lingkungan, lakukan hal berikut:

   a. Untuk menonaktifkan pencarian otomatis laporan, untuk Nama, masukkan **`CODEBUILD_CONFIG_AUTO_DISCOVER`** dan untuk Nilai, masukkan**`false`**. Ini menonaktifkan pencarian otomatis laporan.

   b. (Opsional) Untuk Nama, masukkan **`CODEBUILD_CONFIG_AUTO_DISCOVER_DIR`** dan untuk Nilai, masukkan direktori CodeBuild tempat mencari file format laporan yang didukung. Misalnya, `output/*xml` mencari `.xml` file di `output` direktori

Grup laporan

Grup laporan berisi laporan pengujian dan menentukan pengaturan bersama. Anda menggunakan file buildspec untuk menentukan kasus uji yang akan dijalankan dan perintah untuk menjalankannya saat dibangun. Untuk setiap grup laporan yang dikonfigurasi dalam proyek build, menjalankan proyek build membuat laporan pengujian. Beberapa kali menjalankan proyek build yang dikonfigurasi dengan grup laporan membuat beberapa laporan pengujian dalam grup laporan tersebut, masing-masing dengan hasil kasus pengujian yang sama yang ditentukan untuk grup laporan tersebut.

Kasus uji ditentukan untuk grup laporan dalam file buildspec proyek build. Anda dapat menentukan hingga lima grup laporan dalam satu proyek build. Saat Anda menjalankan build, semua kasus uji

berjalan. Laporan pengujian baru dibuat dengan hasil dari setiap kasus uji yang ditentukan untuk grup laporan. Setiap kali Anda menjalankan build baru, kasus uji dijalankan dan laporan pengujian baru dibuat dengan hasil pengujian baru.

Grup laporan dapat digunakan di lebih dari satu proyek build. Semua laporan pengujian yang dibuat dengan satu grup laporan berbagi konfigurasi yang sama, seperti opsi ekspor dan izin, bahkan jika laporan pengujian dibuat menggunakan proyek build yang berbeda. Laporan pengujian yang dibuat dengan satu grup laporan di beberapa proyek build dapat berisi hasil dari menjalankan set kasus uji yang berbeda (satu set kasus uji untuk setiap proyek build). Ini karena Anda dapat menentukan file kasus uji yang berbeda untuk grup laporan di setiap file buildspec proyek. Anda juga dapat mengubah file kasus uji untuk grup laporan dalam proyek build dengan mengedit file buildspec-nya. Proses build berikutnya membuat laporan pengujian baru yang berisi hasil file kasus uji di buildspec yang diperbarui.

Topik

- Membuat grup laporan
- Penamaan grup laporan
- Bagikan grup laporan
- Tentukan file pengujian
- Tentukan perintah pengujian
- Menandai grup laporan di AWS CodeBuild
- Memperbarui grup laporan

Membuat grup laporan

Anda dapat menggunakan CodeBuild konsol, file AWS CLI, atau buildspec untuk membuat grup laporan. Peran IAM Anda harus memiliki izin yang diperlukan untuk membuat grup laporan. Untuk informasi selengkapnya, lihat Izin laporan uji.

Topik

- Buat grup laporan (buildspec)
- Membuat grup laporan (konsol)
- Buat grup laporan (CLI)
- Buat grup laporan (CloudFormation)

Buat grup laporan (buildspec)

Grup laporan yang dibuat menggunakan buildspec tidak mengekspor file hasil pengujian mentah. Anda dapat melihat grup laporan dan menentukan pengaturan ekspor. Untuk informasi selengkapnya, lihat Memperbarui grup laporan.

Untuk membuat grup laporan menggunakan file buildspec

1. Pilih nama grup laporan yang tidak terkait dengan grup laporan di AWS akun Anda.
2. Konfigurasikan `reports` bagian file buildspec dengan nama ini. Dalam contoh ini, nama grup laporan adalah `new-report-group` dan kasus uji penggunaan dibuat dengan kerangka kerja JUnit:

```
reports:
 new-report-group: #surefire junit reports
   files:
     - '**/*'
   base-directory: 'surefire/target/surefire-reports'
```

Nama grup laporan juga dapat ditentukan dengan menggunakan variabel lingkungan di buildspec:

```
version: 0.2
env:
  variables:
    REPORT_GROUP_NAME: "new-report-group"
phases:
  build:
    commands:
      - ...
...
reports:
 $REPORT_GROUP_NAME:
   files:
     - '**/*'
   base-directory: 'surefire/target/surefire-reports'
```

Untuk informasi selengkapnya, lihat Tentukan file pengujian dan Reports syntax in the buildspec file.

3. Di `commands` bagian tersebut, tentukan perintah untuk menjalankan tes Anda. Untuk informasi selengkapnya, lihat Tentukan perintah pengujian.
4. Jalankan build. Ketika build selesai, grup laporan baru dibuat dengan nama yang menggunakan format`project-name-report-group-name`. Untuk informasi selengkapnya, lihat Penamaan grup laporan.

Membuat grup laporan (konsol)

Gunakan prosedur berikut untuk membuat grup laporan menggunakan Konsol Manajemen AWS.

Untuk membuat grup laporan

1. Buka AWS CodeBuild konsol di https://console.aws.amazon.com/codesuite/codebuild/home.
2. Di panel navigasi, pilih Grup laporan.
3. Pilih Buat grup laporan.
4. Untuk Nama grup laporan, masukkan nama untuk grup laporan Anda.
5. (Opsional) Untuk Tag, masukkan nama dan nilai tag apa pun yang ingin AWS digunakan layanan pendukung. Gunakan Tambahkan baris untuk menambahkan tag. Anda dapat menambahkan hingga 50 tag.
6. Jika Anda ingin mengunggah data mentah hasil laporan pengujian Anda ke bucket Amazon S3:
    a. Pilih Ekspor ke Amazon S3.
    b. Untuk nama bucket S3, masukkan nama bucket S3.
    c. (Opsional) Untuk pemilik bucket S3, AWS masukkan pengenal akun akun yang memiliki bucket S3. Ini memungkinkan data laporan untuk diekspor ke bucket Amazon S3 yang dimiliki oleh akun selain akun yang menjalankan build.
    d. Untuk Path prefix, masukkan path di bucket S3 tempat Anda ingin mengunggah hasil pengujian.
    e. Pilih Kompres data hasil tes dalam file zip untuk mengompres file data hasil tes mentah Anda.
    f. Perluas Konfigurasi tambahan untuk menampilkan opsi enkripsi. Pilih salah satu cara berikut:

- K AWS unci terkelola default untuk menggunakan Kunci yang dikelola AWS untuk Amazon S3. Untuk informasi selengkapnya, lihat CMK yang dikelola pelanggan di Panduan AWS Key Management Service Pengguna. Ini adalah opsi enkripsi default.
- Pilih kunci khusus untuk menggunakan kunci yang dikelola pelanggan yang Anda buat dan konfigurasikan. Untuk kunci AWS KMS enkripsi, masukkan ARN kunci enkripsi Anda. Formatnya adalah `arn:aws:kms:<region-id>: <aws-account-id>:key/<key-id>` . Untuk informasi selengkapnya, lihat Membuat kunci KMS di Panduan AWS Key Management Service Pengguna.
- Nonaktifkan enkripsi artefak untuk menonaktifkan enkripsi. Anda dapat memilih ini jika Anda ingin membagikan hasil tes Anda, atau mempublikasikannya ke situs web statis. (Situs web dinamis dapat menjalankan kode untuk mendekripsi hasil pengujian.)

Untuk informasi selengkapnya tentang enkripsi data saat diam, lihatEnkripsi data.

> **Note**
>
> Per CodeBuild an layanan yang ditentukan dalam proyek digunakan untuk izin untuk mengunggah ke bucket S3.

7. Pilih Buat grup laporan.

Buat grup laporan (CLI)

Gunakan prosedur berikut untuk membuat grup laporan menggunakan AWS CLI.

Untuk membuat grup laporan

1. Buat file bernama `CreateReportGroup.json`.
2. Tergantung pada kebutuhan Anda, salin salah satu cuplikan kode JSON berikut ke: `CreateReportGroup.json`

   - Gunakan JSON berikut untuk menentukan bahwa grup laporan pengujian Anda mengekspor file hasil pengujian mentah ke bucket Amazon S3.

```
{
  "name": "<report-name>",
```

```
  "type": "TEST",
  "exportConfig": {
    "exportConfigType": "S3",
    "s3Destination": {
      "bucket": "<bucket-name>",
      "bucketOwner": "<bucket-owner>",
      "path": "<path>",
      "packaging": "NONE | ZIP",
      "encryptionDisabled": "false",
      "encryptionKey": "<your-key>"
    },
    "tags": [
      {
        "key": "tag-key",
        "value": "tag-value"
      }
    ]
  }
}
```

- Ganti *<bucket-name>* dengan nama bucket Amazon S3 Anda dan *<path>* dengan jalur di bucket Anda ke tempat Anda ingin mengekspor file.
- Jika Anda ingin mengompres file yang diekspor, untuk`packaging`, tentukan`ZIP`. Jika tidak, tentukan`NONE`.
- `bucketOwner`bersifat opsional dan hanya diperlukan jika bucket Amazon S3 dimiliki oleh akun selain akun yang menjalankan build.
- Gunakan `encryptionDisabled` untuk menentukan apakah akan mengenkripsi file yang diekspor. Jika Anda mengenkripsi file yang diekspor, masukkan kunci yang dikelola pelanggan Anda. Untuk informasi selengkapnya, lihat Memperbarui grup laporan.

- Gunakan JSON berikut untuk menentukan bahwa laporan pengujian Anda tidak mengekspor file pengujian mentah:

```
{
  "name": "<report-name>",
  "type": "TEST",
  "exportConfig": {
    "exportConfigType": "NO_EXPORT"
  }
}
```

> **Note**
>
> Per CodeBuild an layanan yang ditentukan dalam proyek digunakan untuk izin untuk mengunggah ke bucket S3.

3. Jalankan perintah berikut:

```
aws codebuild create-report-group --cli-input-json file://
CreateReportGroupInput.json
```

Buat grup laporan (CloudFormation)

Gunakan petunjuk berikut untuk membuat grup laporan menggunakan CloudFormation template

Untuk membuat grup laporan menggunakan CloudFormation template

Anda dapat menggunakan file CloudFormation template untuk membuat dan menyediakan grup laporan. Untuk informasi selengkapnya, lihat Panduan Pengguna CloudFormation.

Template CloudFormation YAML berikut membuat grup laporan yang tidak mengekspor file hasil pengujian mentah.

```
Resources:
  CodeBuildReportGroup:
    Type: AWS::CodeBuild::ReportGroup
    Properties:
      Name: my-report-group-name
      Type: TEST
      ExportConfig:
        ExportConfigType: NO_EXPORT
```

Template CloudFormation YAML berikut membuat grup laporan yang mengekspor file hasil pengujian mentah ke bucket Amazon S3.

```
Resources:
  CodeBuildReportGroup:
    Type: AWS::CodeBuild::ReportGroup
    Properties:
      Name: my-report-group-name
```

```
    Type: TEST
    ExportConfig:
      ExportConfigType: S3
      S3Destination:
        Bucket: amzn-s3-demo-bucket
        Path: path-to-folder-for-exported-files
        Packaging: ZIP
        EncryptionKey: my-KMS-encryption-key
        EncryptionDisabled: false
```

> **Note**
>
> Per CodeBuild an layanan yang ditentukan dalam proyek digunakan untuk izin untuk mengunggah ke bucket S3.

Penamaan grup laporan

Saat Anda menggunakan AWS CLI atau AWS CodeBuild konsol untuk membuat grup laporan, Anda menentukan nama untuk grup laporan. Jika Anda menggunakan buildspec untuk membuat grup laporan baru, itu diberi nama menggunakan format. *`project-name`*`-`*`report-group-name-specified-in-buildspec`* Semua laporan yang dibuat dengan menjalankan build proyek build tersebut milik grup laporan baru yang memiliki nama baru.

Jika Anda tidak CodeBuild ingin membuat grup laporan baru, tentukan ARN grup laporan dalam file buildspec proyek build. Anda dapat menentukan ARN grup laporan dalam beberapa proyek build. Setelah setiap proyek build berjalan, grup laporan berisi laporan pengujian yang dibuat oleh setiap proyek build.

Misalnya, jika Anda membuat satu grup laporan dengan nama tersebut`my-report-group`, dan kemudian menggunakan namanya dalam dua proyek build berbeda bernama `my-project-1` dan `my-project-2` dan membuat build dari kedua proyek, dua grup laporan baru akan dibuat. Hasilnya adalah tiga kelompok laporan dengan nama-nama berikut:

- `my-report-group`: Tidak memiliki laporan pengujian.
- `my-project-1-my-report-group`: Berisi laporan dengan hasil pengujian yang dijalankan oleh proyek build bernama`my-project-1`.
- `my-project-2-my-report-group`: Berisi laporan dengan hasil pengujian yang dijalankan oleh proyek build bernama`my-project-2`.

Jika Anda menggunakan ARN grup laporan yang diberi nama `my-report-group` di kedua proyek, dan kemudian menjalankan build dari setiap proyek, Anda masih memiliki satu grup laporan (`my-report-group`). Grup laporan itu berisi laporan pengujian dengan hasil pengujian yang dijalankan oleh kedua proyek build.

Jika Anda memilih nama grup laporan yang bukan milik grup laporan di AWS akun Anda, lalu menggunakan nama itu untuk grup laporan dalam file buildspec dan menjalankan build proyek buildnya, grup laporan baru akan dibuat. Format nama grup laporan baru adalah*`project-name`*`-`*`new-group-name`*. Misalnya, jika tidak ada grup laporan di AWS akun Anda dengan nama tersebut`new-report-group`, dan tentukan dalam proyek build yang disebut`test-project`, proses build akan membuat grup laporan baru dengan nama tersebut`test-project-new-report-group`.

Bagikan grup laporan

Berbagi grup laporan memungkinkan beberapa AWS akun atau pengguna untuk melihat grup laporan, laporannya yang belum kedaluwarsa, dan hasil pengujian laporannya. Dalam model ini, akun yang memiliki grup laporan (pemilik) berbagi grup laporan dengan akun lain (konsumen). Konsumen tidak dapat mengedit grup laporan. Laporan kedaluwarsa dalam 30 hari setelah dibuat.

Topik

- Bagikan grup laporan
- Layanan terkait
- Mengakses grup laporan yang dibagikan dengan Anda
- Batalkan pembagian grup laporan bersama
- Mengidentifikasi grup laporan bersama
- Izin grup laporan bersama

Bagikan grup laporan

Saat Anda berbagi grup laporan, konsumen diberikan akses baca-saja ke grup laporan dan laporannya. Konsumen dapat menggunakan AWS CLI untuk melihat grup laporan, laporannya, dan hasil kasus uji untuk setiap laporan. Konsumen tidak dapat:

- Melihat grup laporan bersama atau laporannya di CodeBuild konsol.
- Mengedit grup laporan bersama.

- Gunakan ARN grup laporan bersama dalam proyek untuk menjalankan laporan. Pembuatan proyek yang menentukan grup laporan bersama gagal.

Anda dapat menggunakan CodeBuild konsol untuk menambahkan grup laporan ke berbagi sumber daya yang ada. Jika Anda ingin menambahkan grup laporan ke berbagi sumber daya baru, Anda harus membuatnya terlebih dahulu di AWS RAM konsol.

Untuk berbagi grup laporan dengan unit organisasi atau seluruh organisasi, Anda harus mengaktifkan berbagi dengan AWS Organizations. Untuk informasi lebih lanjut, lihat Aktifkan pembagian dengan AWS Organizations dalam Panduan Pengguna AWS RAM .

Anda dapat menggunakan CodeBuild konsol, AWS RAM konsol, atau AWS CLI untuk berbagi grup laporan yang Anda miliki.

Prasyarat

Untuk membagikan grup laporan, AWS akun Anda harus memilikinya. Anda tidak dapat membagikan grup laporan yang telah dibagikan dengan Anda.

Untuk berbagi grup laporan yang Anda miliki (CodeBuild konsol)

1. Buka AWS CodeBuild konsol di https://console.aws.amazon.com/codesuite/codebuild/home.
2. Di panel navigasi, pilih Grup laporan.
3. Pilih proyek yang ingin Anda bagikan, lalu pilih Bagikan. Untuk informasi selengkapnya, lihat Membuat berbagi sumber daya di Panduan AWS RAM Pengguna.

Untuk membagikan grup laporan yang Anda miliki (AWS RAM konsol)

Lihat Membuat berbagi sumber daya di Panduan AWS RAM Pengguna.

Untuk membagikan grup laporan yang Anda miliki (AWS RAM perintah)

Gunakan perintah create-resource-share.

Untuk berbagi grup laporan yang Anda miliki (CodeBuild perintah)

Gunakan perintah put-resource-policy:

1. Buat file bernama `policy.json` dan salin yang berikut ke dalamnya.

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[{
     "Effect":"Allow",
     "Principal":{
       "AWS":"111122223333"
     },
     "Action":[
       "codebuild:BatchGetReportGroups",
       "codebuild:BatchGetReports",
       "codebuild:ListReportsForReportGroup",
       "codebuild:DescribeTestCases"],
     "Resource":"arn:aws:iam::*:role/Service*"
   }]
 }
```

2. Perbarui `policy.json` dengan grup laporan ARN dan pengidentifikasi untuk dibagikan. Contoh berikut memberikan akses read-only ke grup laporan dengan ARN ke Alice dan pengguna root `arn:aws:codebuild:us-west-2:123456789012:report-group/my-report-group` untuk AWS akun yang diidentifikasi oleh 123456789012.

   JSON

```
{
   "Version":"2012-10-17",
   "Statement":[{
     "Effect":"Allow",
     "Principal":{
       "AWS": [
          "arn:aws:iam::123456789012:user/Alice",
          "123456789012"
        ]
     },
     "Action":[
       "codebuild:BatchGetReportGroups",
       "codebuild:BatchGetReports",
       "codebuild:ListReportsForReportGroup",
       "codebuild:DescribeTestCases"],
```

```
    "Resource":"arn:aws:codebuild:us-west-2:123456789012:report-group/my-
report-group"
   }]
 }
```

3. Jalankan perintah berikut.

```
aws codebuild put-resource-policy --resource-arn report-group-arn --policy file://
policy.json
```

Layanan terkait

Berbagi grup laporan terintegrasi dengan AWS Resource Access Manager (AWS RAM), layanan yang memungkinkan Anda berbagi AWS sumber daya dengan AWS akun atau melalui akun apa pun AWS Organizations. Dengan AWS RAM, Anda berbagi sumber daya yang Anda miliki dengan membuat pembagian sumber daya yang menentukan sumber daya dan konsumen untuk dibagikan. Konsumen dapat berupa AWS akun individu, unit organisasi di AWS Organizations, atau seluruh organisasi di dalamnya AWS Organizations.

Untuk informasi selengkapnya, silakan lihat Panduan Pengguna AWS RAM.

Mengakses grup laporan yang dibagikan dengan Anda

Untuk mengakses grup laporan bersama, peran IAM konsumen memerlukan `BatchGetReportGroups` izin. Anda dapat melampirkan kebijakan berikut ke peran IAM mereka:

```
{
    "Effect": "Allow",
    "Resource": [
        "*"
    ],
    "Action": [
        "codebuild:BatchGetReportGroups"
    ]
}
```

Untuk informasi selengkapnya, lihat Menggunakan kebijakan berbasis identitas untuk AWS CodeBuild.

Batalkan pembagian grup laporan bersama

Grup laporan yang tidak dibagikan, termasuk laporannya dan hasil uji kasusnya, hanya dapat diakses oleh pemiliknya. Jika Anda membatalkan berbagi grup laporan, AWS akun atau pengguna mana pun yang sebelumnya Anda bagikan tidak dapat mengakses grup laporan, laporannya, atau hasil kasus uji dalam laporan.

Untuk membatalkan berbagi grup laporan bersama yang Anda miliki, Anda harus menghapusnya dari berbagi sumber daya. Anda dapat menggunakan AWS RAM konsol atau AWS CLI untuk melakukan ini.

Untuk membatalkan berbagi grup laporan bersama yang Anda miliki (AWS RAM konsol)

Lihat Memperbarui Pembagian Sumber Daya di Panduan Pengguna.AWS RAM

Untuk membatalkan berbagi grup laporan bersama yang Anda miliki (AWS RAM perintah)

Gunakan perintah disassociate-resource-share.

Untuk membatalkan berbagi grup laporan yang Anda miliki (CodeBuild perintah)

Jalankan perintah delete-resource-policy dan tentukan ARN grup laporan yang ingin Anda batalkan berbagi:

```
aws codebuild delete-resource-policy --resource-arn report-group-arn
```

Mengidentifikasi grup laporan bersama

Pemilik dan konsumen dapat menggunakan AWS CLI untuk mengidentifikasi grup laporan bersama.

Untuk mengidentifikasi dan mendapatkan informasi tentang grup laporan bersama dan laporannya, gunakan perintah berikut:

- Untuk melihat ARN grup laporan yang dibagikan dengan Anda, jalankan: `list-shared-report-groups`

  ```
  aws codebuild list-shared-report-groups
  ```

- Untuk melihat ARN laporan dalam grup laporan, jalankan `list-reports-for-report-group` menggunakan grup laporan ARN:

```
aws codebuild list-reports-for-report-group --report-group-arn report-group-arn
```

- Untuk melihat informasi tentang kasus uji dalam laporan, jalankan describe-test-cases menggunakan laporan ARN:

```
aws codebuild describe-test-cases --report-arn report-arn
```

Outputnya terlihat seperti berikut:

```
{
    "testCases": [
        {
            "status": "FAILED",
            "name": "Test case 1",
            "expired": 1575916770.0,
            "reportArn": "report-arn",
            "prefix": "Cucumber tests for agent",
            "message": "A test message",
            "durationInNanoSeconds": 1540540,
            "testRawDataPath": "path-to-output-report-files"
        },
        {
            "status": "SUCCEEDED",
            "name": "Test case 2",
            "expired": 1575916770.0,
            "reportArn": "report-arn",
            "prefix": "Cucumber tests for agent",
            "message": "A test message",
            "durationInNanoSeconds": 1540540,
            "testRawDataPath": "path-to-output-report-files"
        }
    ]
}
```

Izin grup laporan bersama

Izin untuk pemilik

Pemilik grup laporan dapat mengedit grup laporan dan menentukannya dalam proyek untuk menjalankan laporan.

Izin untuk konsumen

Konsumen grup laporan dapat melihat grup laporan, laporannya, dan hasil kasus uji untuk laporannya. Konsumen tidak dapat mengedit grup laporan atau laporannya, dan tidak dapat menggunakannya untuk membuat laporan.

Tentukan file pengujian

Anda menentukan file hasil pengujian dan lokasinya untuk setiap grup laporan di `reports` bagian file buildspec proyek build Anda. Untuk informasi selengkapnya, lihat Reports syntax in the buildspec file.

Berikut ini adalah `reports` bagian contoh yang menentukan dua grup laporan untuk proyek build. Satu ditentukan dengan ARN-nya, yang lain dengan nama. `files`Bagian ini menentukan file yang berisi hasil uji kasus. `base-directory`Bagian opsional menentukan direktori tempat file kasus uji berada. `discard-paths`Bagian opsional menentukan apakah jalur ke file hasil pengujian yang diunggah ke bucket Amazon S3 akan dibuang.

```
reports:
  arn:aws:codebuild:your-region:your-aws-account-id:report-group/report-group-name-1:
 #surefire junit reports
    files:
      - '**/*'
    base-directory: 'surefire/target/surefire-reports'
    discard-paths: false

  sampleReportGroup: #Cucumber reports from json plugin
    files:
      - 'cucumber-json/target/cucumber-json-report.json'
    file-format: CUCUMBERJSON #Type of the report, defaults to JUNITXML
```

Tentukan perintah pengujian

Anda menentukan perintah yang menjalankan kasus uji Anda di `commands` bagian file buildspec Anda. Perintah ini menjalankan kasus uji yang ditentukan untuk grup laporan Anda di `reports` bagian file buildspec Anda. Berikut ini adalah `commands` bagian contoh yang mencakup perintah untuk menjalankan pengujian dalam file uji:

```
commands:
    - echo Running tests for surefire junit
    - mvn test -f surefire/pom.xml -fn
```

```
    - echo
    - echo Running tests for cucumber with json plugin
    - mvn test -Dcucumber.options="--plugin json:target/cucumber-json-report.json" -f
cucumber-json/pom.xml -fn
```

Untuk informasi selengkapnya, lihat Sintaks Buildspec.

Menandai grup laporan di AWS CodeBuild

Tag adalah label atribut kustom yang Anda atau AWS tetapkan ke AWS sumber daya. Setiap AWS tag memiliki dua bagian:

- Kunci tag (misalnya, `CostCenter`, `Environment`, `Project`, atau `Secret`). Kunci tanda peka terhadap huruf besar dan kecil.
- Bidang opsional yang dikenal sebagai nilai tag (misalnya, `111122223333`, `Production`, atau nama tim). Mengabaikan nilai tag sama dengan menggunakan rangkaian kosong. Seperti kunci tanda, nilai tanda peka huruf besar dan kecil.

Bersama-sama ini dikenal sebagai pasangan nilai-kunci. Untuk batasan jumlah tag yang dapat Anda miliki di grup laporan dan batasan pada kunci dan nilai tag, lihatTanda.

Tag membantu Anda mengidentifikasi dan mengatur AWS sumber daya Anda. Banyak AWS layanan mendukung penandaan, sehingga Anda dapat menetapkan tag yang sama ke sumber daya dari layanan yang berbeda untuk menunjukkan bahwa sumber daya terkait. Misalnya, Anda dapat menetapkan tag yang sama ke grup CodeBuild laporan yang ditetapkan ke bucket Amazon S3. Untuk informasi lebih lanjut tentang penggunaan tanda, lihat laporan resmi Praktik terbaik.

Pada tahun CodeBuild, sumber daya utama adalah kelompok laporan dan proyek. Anda dapat menggunakan CodeBuild konsol, CodeBuild API AWS CLI, atau AWS SDK untuk menambahkan, mengelola, dan menghapus tag untuk grup laporan. Selain mengidentifikasi, mengatur, dan melacak grup laporan Anda dengan tag, Anda dapat menggunakan tag dalam kebijakan IAM untuk membantu mengontrol siapa yang dapat melihat dan berinteraksi dengan grup laporan Anda. Untuk contoh kebijakan akses berbasis tag, lihat Menggunakan tag untuk mengontrol akses AWS CodeBuild sumber daya.

Topik

- Menambahkan tag ke grup laporan
- Melihat tag untuk grup laporan

- Mengedit tag untuk grup laporan
- Menghapus tag dari grup laporan

Menambahkan tag ke grup laporan

Menambahkan tag ke grup laporan dapat membantu Anda mengidentifikasi dan mengatur AWS sumber daya serta mengelola akses ke sana. Pertama, Anda menambahkan satu atau beberapa tag (pasangan nilai kunci) ke grup laporan. Perlu diingat bahwa ada batasan jumlah tag yang dapat Anda miliki di grup laporan. Ada pembatasan pada karakter yang dapat Anda gunakan di bidang kunci dan nilai. Untuk informasi selengkapnya, lihat Tanda. Setelah Anda memiliki tag, Anda dapat membuat kebijakan IAM untuk mengelola akses ke grup laporan berdasarkan tag ini. Anda dapat menggunakan CodeBuild konsol atau AWS CLI untuk menambahkan tag ke grup laporan.

> **Important**
>
> Menambahkan tag ke grup laporan dapat memengaruhi akses ke grup laporan tersebut. Sebelum menambahkan tag ke grup laporan, pastikan untuk meninjau kebijakan IAM apa pun yang mungkin menggunakan tag untuk mengontrol akses ke sumber daya seperti grup laporan. Untuk contoh kebijakan akses berbasis tag, lihat Menggunakan tag untuk mengontrol akses AWS CodeBuild sumber daya.

Untuk informasi selengkapnya tentang menambahkan tag ke grup laporan saat Anda membuatnya, lihatMembuat grup laporan (konsol).

Topik

- Menambahkan tag ke grup laporan (konsol)
- Menambahkan tag ke grup laporan (AWS CLI)

Menambahkan tag ke grup laporan (konsol)

Anda dapat menggunakan CodeBuild konsol untuk menambahkan satu atau beberapa tag ke grup CodeBuild laporan.

1. Buka CodeBuild konsol di https://console.aws.amazon.com/codebuild/.
2. Di Grup laporan, pilih nama grup laporan tempat Anda ingin menambahkan tag.
3. Pada panel navigasi, silakan pilih Pengaturan.

4. Jika tidak ada tag yang ditambahkan ke grup laporan, pilih Tambah tag. Anda juga dapat memilih Edit, lalu pilih Tambah tag.
5. Di Kunci, masukkan sebuah nama untuk tag tersebut. Anda dapat menambahkan nilai opsional untuk tag di Nilai.
6. (Opsional) Untuk menambahkan tag lain, pilih Tambahkan tag lagi.
7. Setelah Anda selesai menambahkan tag, pilih Kirim.

Menambahkan tag ke grup laporan (AWS CLI)

Untuk menambahkan tag ke grup laporan saat Anda membuatnya, lihatBuat grup laporan (CLI). Di`CreateReportGroup.json`, tambahkan tag Anda.

Untuk menambahkan tag ke grup laporan yang ada, lihat Memperbarui grup laporan (CLI) dan tambahkan tag Anda`UpdateReportGroupInput.json`.

Dalam langkah-langkah ini, kami menganggap bahwa Anda telah menginstal versi terbaru dari AWS CLI atau diperbarui ke versi terkini. Untuk informasi lebih lanjut, lihat Menginstal AWS Command Line Interface.

Melihat tag untuk grup laporan

Tag dapat membantu Anda mengidentifikasi dan mengatur AWS sumber daya Anda dan mengelola akses ke sana. Untuk informasi lebih lanjut tentang penggunaan tanda, lihat laporan resmi Praktik terbaik. Untuk contoh kebijakan akses berbasis tag, lihat Deny or allow actions on report groups based on resource tags.

Melihat tag untuk grup laporan (konsol)

Anda dapat menggunakan CodeBuild konsol untuk melihat tag yang terkait dengan grup CodeBuild laporan.

1. Buka CodeBuild konsol di https://console.aws.amazon.com/codebuild/.
2. Di Grup laporan, pilih nama grup laporan tempat Anda ingin melihat tag.
3. Pada panel navigasi, silakan pilih Pengaturan.

Melihat tag untuk grup laporan (AWS CLI)

Ikuti langkah-langkah ini untuk menggunakan AWS CLI untuk melihat AWS tag untuk grup laporan. Jika tidak ada tag yang ditambahkan, daftar tag yang dikembalikan kosong.

1. Gunakan konsol atau AWS CLI untuk menemukan ARN grup laporan Anda. Catat itu.

   AWS CLI

   Jalankan perintah berikut.

   ```
   aws list-report-groups
   ```

   Perintah ini mengembalikan JSON-formatted informasi yang mirip dengan berikut ini:

   ```
   {
       "reportGroups": [
           "arn:aws:codebuild:region:123456789012:report-group/report-group-1",
           "arn:aws:codebuild:region:123456789012:report-group/report-group-2",
           "arn:aws:codebuild:region:123456789012:report-group/report-group-3"
       ]
   }
   ```

   Grup laporan ARN diakhiri dengan namanya, yang dapat Anda gunakan untuk mengidentifikasi ARN untuk grup laporan Anda.

   Console

   1. Buka CodeBuild konsol di https://console.aws.amazon.com/codebuild/.
   2. Di Grup laporan, pilih nama grup laporan Anda dengan tag yang ingin Anda lihat.
   3. Di Konfigurasi, temukan ARN grup laporan Anda.

2. Jalankan perintah berikut. Gunakan ARN yang Anda catat untuk parameter tersebut. `--report-group-arns`

   ```
   aws codebuild batch-get-report-groups --report-group-arns
    arn:aws:codebuild:region:123456789012:report-group/report-group-name
   ```

   Jika berhasil, perintah ini mengembalikan JSON-formatted informasi yang berisi `tags` bagian yang mirip dengan berikut ini:

   ```
   {
       ...
       "tags": {
           "Status": "Secret",
           "Project": "TestBuild"
   ```

```
    }
    ...
}
```

Mengedit tag untuk grup laporan

Anda dapat mengubah nilai untuk tag yang terkait dengan grup laporan. Anda juga dapat mengubah nama kunci, yang setara dengan menghapus tag saat ini dan menambahkan tag yang berbeda dengan nama baru dan nilai yang sama dengan kunci lainnya. Perlu diingat bahwa ada batasan pada karakter yang dapat Anda gunakan di bidang kunci dan nilai. Untuk informasi selengkapnya, lihat Tanda.

> **Important**
>
> Mengedit tag untuk grup laporan dapat memengaruhi akses ke grup laporan tersebut. Sebelum Anda mengedit nama (kunci) atau nilai tag untuk grup laporan, pastikan untuk meninjau kebijakan IAM apa pun yang mungkin menggunakan kunci atau nilai tag untuk mengontrol akses ke sumber daya seperti grup laporan. Untuk contoh kebijakan akses berbasis tag, lihat Deny or allow actions on report groups based on resource tags.

Mengedit tag untuk grup laporan (konsol)

Anda dapat menggunakan CodeBuild konsol untuk mengedit tag yang terkait dengan grup CodeBuild laporan.

1. Buka CodeBuild konsol di https://console.aws.amazon.com/codebuild/.
2. Di Grup laporan, pilih nama grup laporan tempat Anda ingin mengedit tag.
3. Pada panel navigasi, silakan pilih Pengaturan.
4. Pilih Edit.
5. Lakukan salah satu tindakan berikut:
   - Untuk mengubah tag, masukkan nama baru di Kunci. Mengubah nama tag adalah setara dengan menghapus tag dan menambahkan tag baru dengan nama kunci baru.
   - Untuk mengubah nilai sebuah tag, masukkan nilai baru. Jika Anda ingin mengubah nilai menjadi nol, hapus nilai saat ini dan biarkan bidang kosong.
6. Setelah Anda selesai mengedit tag, pilih Kirim.

Mengedit tag untuk grup laporan (AWS CLI)

Untuk menambahkan, mengubah, atau menghapus tag dari grup laporan, lihatMemperbarui grup laporan (CLI). Perbarui tag di`UpdateReportGroupInput.json`.

Menghapus tag dari grup laporan

Anda dapat menghapus satu atau beberapa tag yang terkait dengan grup laporan. Menghapus tag tidak menghapus tag dari AWS sumber daya lain yang terkait dengan tag tersebut.

> **Important**
>
> Menghapus tag untuk grup laporan dapat memengaruhi akses ke grup laporan tersebut. Sebelum menghapus tag dari grup laporan, pastikan untuk meninjau kebijakan IAM apa pun yang mungkin menggunakan kunci atau nilai tag untuk mengontrol akses ke sumber daya seperti grup laporan. Untuk contoh kebijakan akses berbasis tag, lihat Menggunakan tag untuk mengontrol akses AWS CodeBuild sumber daya.

Menghapus tag dari grup laporan (konsol)

Anda dapat menggunakan CodeBuild konsol untuk menghapus asosiasi antara tag dan grup CodeBuild laporan.

1. Buka CodeBuild konsol di https://console.aws.amazon.com/codebuild/.
2. Di Grup laporan, pilih nama grup laporan tempat Anda ingin menghapus tag.
3. Pada panel navigasi, silakan pilih Pengaturan.
4. Pilih Edit.
5. Temukan tag yang ingin Anda hapus, lalu pilih Hapus tag.
6. Setelah Anda selesai menghapus tag, pilih Kirim.

Menghapus tag dari grup laporan (AWS CLI)

Ikuti langkah-langkah berikut untuk menggunakan AWS CLI untuk menghapus tag dari grup CodeBuild laporan. Menghapus tag tidak menghapusnya, tetapi hanya menghapus asosiasi antara tag dan grup laporan.

> **Note**
>
> Jika Anda menghapus grup CodeBuild laporan, semua asosiasi tag akan dihapus dari grup laporan yang dihapus. Anda tidak perlu menghapus tag sebelum menghapus grup laporan.

Untuk menghapus satu atau beberapa tag dari grup laporan, lihatMengedit tag untuk grup laporan (AWS CLI). Perbarui `tags` bagian dalam JSON-formatted data dengan daftar tag yang diperbarui yang tidak berisi tag yang ingin Anda hapus. Jika Anda ingin menghapus semua tag, perbarui `tags` bagian ke:

```
"tags: []"
```

Memperbarui grup laporan

Saat memperbarui grup laporan, Anda dapat menentukan informasi tentang apakah akan mengekspor data hasil pengujian mentah ke file dalam bucket Amazon S3. Jika Anda memilih untuk mengekspor ke bucket S3, Anda dapat menentukan yang berikut untuk grup laporan Anda:

- Apakah file hasil pengujian mentah dikompresi dalam file ZIP.
- Apakah file hasil tes mentah dienkripsi. Anda dapat menentukan enkripsi dengan salah satu dari berikut ini:
  - Sebuah Kunci yang dikelola AWS untuk Amazon S3.
  - Kunci yang dikelola pelanggan yang Anda buat dan konfigurasikan.

Untuk informasi selengkapnya, lihat Enkripsi data.

Jika Anda menggunakan AWS CLI untuk memperbarui grup laporan, Anda juga dapat memperbarui atau menambahkan tag. Untuk informasi selengkapnya, lihat Menandai grup laporan di AWS CodeBuild.

> **Note**
>
> Per CodeBuild an layanan yang ditentukan dalam proyek digunakan untuk izin untuk mengunggah ke bucket S3.

Topik

- Memperbarui grup laporan (konsol)
- Memperbarui grup laporan (CLI)

Memperbarui grup laporan (konsol)

Gunakan prosedur berikut untuk memperbarui grup laporan menggunakan Konsol Manajemen AWS.

Untuk memperbarui grup laporan

1. Buka AWS CodeBuild konsol di https://console.aws.amazon.com/codesuite/codebuild/home.
2. Di panel navigasi, pilih Grup laporan.
3. Pilih grup laporan yang ingin Anda perbarui.
4. Pilih Edit.
5. Pilih atau hapus Cadangan ke Amazon S3. Jika Anda memilih opsi ini, tentukan pengaturan ekspor Anda:
   a. Untuk nama bucket S3, masukkan nama bucket S3.
   b. Untuk prefiks Path, masukkan path di bucket S3 tempat Anda ingin mengunggah hasil pengujian.
   c. Pilih Kompres data hasil tes dalam file zip untuk mengompres file data hasil tes mentah Anda.
   d. Perluas Konfigurasi tambahan untuk menampilkan opsi enkripsi. Pilih salah satu cara berikut:
      - K AWS unci terkelola default untuk menggunakan Kunci yang dikelola AWS untuk Amazon S3. Untuk informasi selengkapnya, lihat CMK yang dikelola pelanggan di Panduan AWS Key Management Service Pengguna. Ini adalah opsi enkripsi default.
      - Pilih kunci khusus untuk menggunakan kunci yang dikelola pelanggan yang Anda buat dan konfigurasikan. Untuk kunci AWS KMS enkripsi, masukkan ARN kunci enkripsi Anda. Formatnya adalah `arn:aws:kms:<region-id>: <aws-account-id>:key/<key-id>` . Untuk informasi selengkapnya, lihat Membuat kunci KMS di Panduan AWS Key Management Service Pengguna.
      - Nonaktifkan enkripsi artefak untuk menonaktifkan enkripsi. Anda dapat memilih ini jika Anda ingin membagikan hasil tes Anda, atau mempublikasikannya ke situs web statis. (Situs web dinamis dapat menjalankan kode untuk mendekripsi hasil pengujian.)

Memperbarui grup laporan (CLI)

Gunakan prosedur berikut untuk memperbarui grup laporan menggunakan AWS CLI.

Untuk memperbarui grup laporan

1. Buat file bernama `UpdateReportGroupInput.json`.
2. Salin yang berikut ini ke`UpdateReportGroupInput.json`:

```
{
    "arn": "",
    "exportConfig": {
        "exportConfigType": "S3",
        "s3Destination": {
            "bucket": "bucket-name",
            "path": "path",
            "packaging": "NONE | ZIP",
            "encryptionDisabled": "false",
            "encryptionKey": "your-key"
         }
     },
     "tags": [
        {
            "key": "tag-key",
            "value": "tag-value"
        }
     ]
}
```

3. Masukkan ARN grup laporan Anda di `arn` baris (misalnya,`"arn":"arn:aws:codebuild:region:123456789012:report-group/report-group-1"`).
4. Perbarui `UpdateReportGroupInput.json` dengan pembaruan yang ingin Anda terapkan ke grup laporan Anda.

   - Jika Anda ingin memperbarui grup laporan untuk mengekspor file hasil pengujian mentah ke bucket S3, perbarui `exportConfig` bagian tersebut. Ganti `bucket-name` dengan nama bucket S3 Anda dan `path` dengan jalur di bucket S3 Anda yang ingin Anda ekspor file. Jika Anda ingin mengompres file yang diekspor, untuk`packaging`, tentukan`ZIP`. Jika tidak, tentukan`NONE`. Gunakan `encryptionDisabled` untuk menentukan apakah akan

mengenkripsi file yang diekspor. Jika Anda mengenkripsi file yang diekspor, masukkan kunci yang dikelola pelanggan Anda.

- Jika Anda ingin memperbarui grup laporan sehingga tidak mengekspor file hasil pengujian mentah ke bucket S3, perbarui `exportConfig` bagian dengan JSON berikut:

```
{
  "exportConfig": {
      "exportConfigType": "NO_EXPORT"
  }
}
```

- Jika Anda ingin memperbarui tag grup laporan, perbarui `tags` bagian tersebut. Anda dapat mengubah, menambah, atau menghapus tag. Jika Anda ingin menghapus semua tag, perbarui dengan JSON berikut:

```
"tags": []
```

5. Jalankan perintah berikut:

```
aws codebuild update-report-group \
--cli-input-json file://UpdateReportGroupInput.json
```

Kerangka uji

Topik di bagian ini menunjukkan cara mengatur pelaporan pengujian AWS CodeBuild untuk berbagai kerangka pengujian.

Topik

- Siapkan pelaporan tes dengan Jasmine
- Siapkan pelaporan pengujian dengan Jest
- Siapkan pelaporan pengujian dengan pytest
- Siapkan pelaporan pengujian dengan RSpec

Siapkan pelaporan tes dengan Jasmine

Prosedur berikut menunjukkan cara mengatur pelaporan pengujian AWS CodeBuild dengan kerangka pengujian JasmineBDD.

Prosedur ini membutuhkan prasyarat berikut:

- Anda memiliki CodeBuild proyek yang sudah ada.
- Proyek Anda adalah Node.js proyek yang diatur untuk menggunakan kerangka pengujian Jasmine.

Tambahkan `jasmine-reporters` paket ke `devDependencies` bagian `package.json` file proyek Anda. Paket ini memiliki koleksi kelas JavaScript reporter yang dapat digunakan dengan Jasmine.

```
npm install --save-dev jasmine-reporters
```

Jika belum ada, tambahkan `test` skrip ke `package.json` file proyek Anda. `test`Skrip memastikan bahwa Jasmine dipanggil saat npm test dijalankan.

```
{
  "scripts": {
    "test": "npx jasmine"
  }
}
```

CodeBuild mendukung reporter tes Jasmine berikut:

JUnitXmlReporter

Digunakan untuk menghasilkan laporan dalam `JunitXml` format.

NUnitXmlReporter

Digunakan untuk menghasilkan laporan dalam `NunitXml` format.

Sebuah Node.js proyek dengan Jasmine akan, secara default, memiliki `spec` sub-direktori, yang berisi konfigurasi Jasmine dan skrip pengujian.

Untuk mengonfigurasi Jasmine untuk menghasilkan laporan dalam `JunitXML` format, buat instance `JUnitXmlReporter` reporter dengan menambahkan kode berikut ke pengujian Anda.

```
var reporters = require('jasmine-reporters');

var junitReporter = new reporters.JUnitXmlReporter({
  savePath: <test report directory>,
  filePrefix: <report filename>,
```

```
  consolidateAll: true
});

jasmine.getEnv().addReporter(junitReporter);
```

Untuk mengonfigurasi Jasmine untuk menghasilkan laporan dalam `NunitXML` format, buat instance `NUnitXmlReporter` reporter dengan menambahkan kode berikut ke pengujian Anda.

```
var reporters = require('jasmine-reporters');

var nunitReporter = new reporters.NUnitXmlReporter({
  savePath: <test report directory>,
  filePrefix: <report filename>,
  consolidateAll: true
});

jasmine.getEnv().addReporter(nunitReporter)
```

Laporan pengujian diekspor ke file yang ditentukan oleh*<test report directory>*/*<report filename>*.

Dalam `buildspec.yml` file Anda, add/update bagian berikut.

```
version: 0.2

phases:
  pre_build:
    commands:
      - npm install
  build:
    commands:
      - npm build
      - npm test

reports:
  jasmine_reports:
    files:
      - <report filename>
    file-format: JUNITXML
    base-directory: <test report directory>
```

Jika Anda menggunakan format `NunitXml` laporan, ubah nil `file-format` ainya menjadi berikut.

```
    file-format: NUNITXML
```

Siapkan pelaporan pengujian dengan Jest

Prosedur berikut menunjukkan cara mengatur pelaporan pengujian AWS CodeBuild dengan kerangka pengujian Jest.

Prosedur ini membutuhkan prasyarat berikut:

- Anda memiliki CodeBuild proyek yang sudah ada.
- Proyek Anda adalah Node.js proyek yang disiapkan untuk menggunakan kerangka pengujian Jest.

Tambahkan `jest-junit`paket ke `devDependencies` bagian `package.json` file proyek Anda. CodeBuild menggunakan paket ini untuk menghasilkan laporan dalam `JunitXml` format.

```
npm install --save-dev jest-junit
```

Jika belum ada, tambahkan `test` skrip ke `package.json` file proyek Anda. `test`Skrip memastikan bahwa Jest dipanggil saat npm test dijalankan.

```
{
  "scripts": {
    "test": "jest"
  }
}
```

Konfigurasikan Jest untuk menggunakan `JunitXml` reporter dengan menambahkan yang berikut ini ke file konfigurasi Jest Anda. Jika proyek Anda tidak memiliki file konfigurasi Jest, buat file bernama `jest.config.js` di root proyek Anda dan tambahkan yang berikut ini. Laporan pengujian diekspor ke file yang ditentukan oleh*<test report directory>*/*<report filename>*.

```
module.exports = {
  reporters: [
    'default',
    [ 'jest-junit', {
      outputDirectory: <test report directory>,
      outputName: <report filename>,
    } ]
```

```
  ]
};
```

Dalam `buildspec.yml` file Anda, add/update bagian berikut.

```
version: 0.2

phases:
  pre_build:
    commands:
      - npm install
  build:
    commands:
      - npm build
      - npm test

reports:
  jest_reports:
    files:
      - <report filename>
    file-format: JUNITXML
    base-directory: <test report directory>
```

Siapkan pelaporan pengujian dengan pytest

Prosedur berikut menunjukkan cara mengatur pelaporan pengujian AWS CodeBuild dengan kerangka pengujian pytest.

Prosedur ini membutuhkan prasyarat berikut:

- Anda memiliki CodeBuild proyek yang sudah ada.
- Proyek Anda adalah proyek Python yang disiapkan untuk menggunakan kerangka pengujian pytest.

Tambahkan entri berikut ke salah satu `build` atau `post_build` fase `buildspec.yml` file Anda. Kode ini secara otomatis menemukan tes di direktori saat ini dan mengekspor laporan pengujian ke file yang ditentukan oleh*<test report directory>*/. *<report filename>* Laporan menggunakan `JunitXml` format.

```
      - python -m pytest --junitxml=<test report directory>/<report filename>
```

Dalam `buildspec.yml` file Anda, add/update bagian berikut.

```
version: 0.2

phases:
  install:
    runtime-versions:
      python: 3.7
    commands:
      - pip3 install pytest
  build:
    commands:
      - python -m pytest --junitxml=<test report directory>/<report filename>

reports:
  pytest_reports:
    files:
      - <report filename>
    base-directory: <test report directory>
    file-format: JUNITXML
```

Siapkan pelaporan pengujian dengan RSpec

Prosedur berikut menunjukkan cara mengatur pelaporan pengujian AWS CodeBuild dengan kerangka pengujian RSpec.

Prosedur ini membutuhkan prasyarat berikut:

- Anda memiliki CodeBuild proyek yang sudah ada.
- Proyek Anda adalah proyek Ruby yang disiapkan untuk menggunakan kerangka pengujian RSpec.

Add/update berikut ini dalam `buildspec.yml` file Anda. Kode ini menjalankan tes di *<test source directory>* direktori dan mengekspor laporan pengujian ke file yang ditentukan oleh*<test report directory>*/*<report filename>*. Laporan menggunakan `JunitXml` format.

```
version: 0.2

phases:
  install:
    runtime-versions:
```

```
      ruby: 2.6
  pre_build:
    commands:
      - gem install rspec
      - gem install rspec_junit_formatter
  build:
    commands:
      - rspec <test source directory>/* --format RspecJunitFormatter --out <test report
 directory>/<report filename>
reports:
    rspec_reports:
        files:
            - <report filename>
        base-directory: <test report directory>
        file-format: JUNITXML
```

Lihat laporan pengujian

Anda dapat melihat detail tentang laporan pengujian, seperti informasi tentang kasus pengujian, nomor lulus dan gagal, dan berapa lama waktu yang dibutuhkan untuk menjalankannya. Anda dapat melihat laporan pengujian yang dikelompokkan berdasarkan build run, grup laporan, atau AWS akun Anda. Pilih laporan pengujian di konsol untuk melihat detail dan hasil kasus ujiannya.

Anda dapat melihat laporan pengujian yang belum kedaluwarsa. Laporan pengujian kedaluwarsa 30 hari setelah dibuat. Anda tidak dapat melihat laporan kedaluwarsa di CodeBuild.

Topik

- Melihat laporan pengujian untuk build
- Melihat laporan pengujian untuk grup laporan
- Lihat laporan pengujian di AWS akun

Melihat laporan pengujian untuk build

Untuk melihat laporan pengujian untuk build

1. Buka AWS CodeBuild konsol di https://console.aws.amazon.com/codesuite/codebuild/home.
2. Temukan build yang ingin Anda lihat. Jika Anda mengetahui proyek yang menjalankan build yang membuat laporan pengujian:

1. Di panel navigasi, pilih Bang un proyek, lalu pilih proyek dengan build yang menjalankan laporan pengujian yang ingin Anda lihat.
2. Pilih Riwayat build, lalu pilih build yang berjalan membuat laporan yang ingin Anda lihat.

Anda juga dapat menemukan build di riwayat build untuk AWS akun Anda:

1. Di panel navigasi, pilih Ri wayat build, lalu pilih build yang membuat laporan pengujian yang ingin Anda lihat.

3. Di halaman build, pilih Laporan, lalu pilih laporan pengujian untuk melihat detailnya.

Melihat laporan pengujian untuk grup laporan

Untuk melihat laporan pengujian dalam grup laporan

1. Buka AWS CodeBuild konsol di https://console.aws.amazon.com/codesuite/codebuild/home.
2. Di panel navigasi, pilih Grup laporan.
3. Pilih grup laporan yang berisi laporan pengujian yang ingin Anda lihat.
4. Pilih laporan pengujian untuk melihat detailnya.

Lihat laporan pengujian di AWS akun

Untuk melihat laporan pengujian di AWS akun

1. Buka AWS CodeBuild konsol di https://console.aws.amazon.com/codesuite/codebuild/home.
2. Di panel navigasi, pilih Ri wayat laporan.
3. Pilih laporan pengujian untuk melihat detailnya.

Izin laporan uji

Topik ini menjelaskan informasi penting tentang izin yang terkait dengan pelaporan pengujian.

Topik

- Peran IAM untuk laporan pengujian

- Izin untuk operasi pelaporan pengujian
- Contoh izin pelaporan uji

Peran IAM untuk laporan pengujian

Untuk menjalankan laporan pengujian, dan memperbarui proyek untuk menyertakan laporan pengujian, peran IAM Anda memerlukan izin berikut. Izin ini disertakan dalam kebijakan terkel AWS ola yang telah ditentukan sebelumnya. Jika Anda ingin menambahkan pelaporan pengujian ke proyek build yang ada, Anda harus menambahkan izin ini sendiri.

- `CreateReportGroup`
- `CreateReport`
- `UpdateReport`
- `BatchPutTestCases`

Untuk menjalankan laporan cakupan kode, peran IAM Anda juga harus menyertakan `BatchPutCodeCoverages` izin tersebut.

> **Note**
>
> `BatchPutTestCases`,`CreateReport`,`UpdateReport`, dan `BatchPutCodeCoverages` bukan izin publik. Anda tidak dapat memanggil AWS CLI perintah atau metode SDK yang sesuai untuk izin ini.

Untuk memastikan Anda memiliki izin ini, Anda dapat melampirkan kebijakan berikut ke peran IAM Anda:

```
{
    "Effect": "Allow",
    "Resource": [
        "*"
    ],
    "Action": [
        "codebuild:CreateReportGroup",
        "codebuild:CreateReport",
        "codebuild:UpdateReport",
```

```
        "codebuild:BatchPutTestCases",
        "codebuild:BatchPutCodeCoverages"
    ]
}
```

Sebaiknya batasi kebijakan ini hanya untuk grup laporan yang harus Anda gunakan. Berikut ini membatasi izin hanya untuk grup laporan dengan dua ARN dalam kebijakan:

```
{
    "Effect": "Allow",
    "Resource": [
        "arn:aws:codebuild:your-region:your-aws-account-id:report-group/report-group-
name-1",
        "arn:aws:codebuild:your-region:your-aws-account-id:report-group/report-group-
name-2"
    ],
    "Action": [
        "codebuild:CreateReportGroup",
        "codebuild:CreateReport",
        "codebuild:UpdateReport",
        "codebuild:BatchPutTestCases",
        "codebuild:BatchPutCodeCoverages"
    ]
}
```

Berikut ini membatasi izin untuk hanya melaporkan grup yang dibuat dengan menjalankan build proyek bernama: `my-project`

```
{
    "Effect": "Allow",
    "Resource": [
        "arn:aws:codebuild:your-region:your-aws-account-id:report-group/my-project-*"
    ],
    "Action": [
        "codebuild:CreateReportGroup",
        "codebuild:CreateReport",
        "codebuild:UpdateReport",
        "codebuild:BatchPutTestCases",
        "codebuild:BatchPutCodeCoverages"
    ]
}
```

> **Note**
>
> Per CodeBuild an layanan yang ditentukan dalam proyek digunakan untuk izin untuk mengunggah ke bucket S3.

Izin untuk operasi pelaporan pengujian

Anda dapat menentukan izin untuk operasi CodeBuild API pelaporan pengujian berikut:

- `BatchGetReportGroups`
- `BatchGetReports`
- `CreateReportGroup`
- `DeleteReportGroup`
- `DeleteReport`
- `DescribeTestCases`
- `ListReportGroups`
- `ListReports`
- `ListReportsForReportGroup`
- `UpdateReportGroup`

Untuk informasi selengkapnya, lihat AWS CodeBuild referensi izin.

Contoh izin pelaporan uji

Untuk informasi tentang kebijakan sampel yang terkait dengan pelaporan pengujian, lihat berikut ini:

- Izinkan pengguna mengubah grup laporan
- Izinkan pengguna membuat grup laporan
- Izinkan pengguna untuk menghapus laporan
- Izinkan pengguna untuk menghapus grup laporan
- Memungkinkan pengguna untuk mendapatkan informasi tentang grup laporan
- Memungkinkan pengguna untuk mendapatkan informasi tentang laporan
- Izinkan pengguna untuk mendapatkan daftar grup laporan

- Memungkinkan pengguna untuk mendapatkan daftar laporan
- Memungkinkan pengguna untuk mendapatkan daftar laporan untuk grup laporan
- Memungkinkan pengguna untuk mendapatkan daftar kasus uji untuk laporan

Status laporan pengujian

Status laporan pengujian dapat berupa salah satu dari berikut ini:

- `GENERATING`: Pengoperasian kasus uji masih berlangsung.
- `DELETING`: Laporan pengujian sedang dihapus. Ketika laporan pengujian dihapus, kasus ujiannya juga dihapus. File data hasil pengujian mentah yang diekspor ke bucket S3 tidak dihapus.
- `INCOMPLETE`: Laporan pengujian belum selesai. Status ini dapat dikembalikan karena salah satu alasan berikut:
  - Masalah dengan konfigurasi grup laporan yang menentukan kasus uji laporan ini. Misalnya, jalur ke kasus uji di bawah grup laporan di file buildspec mungkin salah.
  - Pengguna IAM yang menjalankan build tidak memiliki izin untuk menjalankan pengujian. Untuk informasi selengkapnya, lihat Izin laporan uji.
  - Pembuatan tidak selesai karena kesalahan yang tidak terkait dengan pengujian.
- `SUCCEEDED`: Semua kasus uji berhasil.
- `FAILED`: Beberapa kasus uji tidak berhasil.

Setiap kasus uji mengembalikan status. Status untuk kasus uji dapat berupa salah satu dari berikut ini:

- `SUCCEEDED`: Kasus uji lulus.
- `FAILED`: Kasus uji gagal.
- `ERROR`: Kasus uji menghasilkan kesalahan yang tidak terduga.
- `SKIPPED`: Kasus uji tidak berjalan.
- `UNKNOWN`: Kasus uji mengembalikan status selain`SUCCEEDED`,`FAILED`,`ERROR`, atau`SKIPPED`.

Laporan pengujian dapat memiliki maksimum 500 hasil uji kasus. Jika lebih dari 500 kasus uji dijalankan, CodeBuild memprioritaskan pengujian dengan status `FAILED` dan memotong hasil kasus uji.

Gunakan AWS CodeBuild dengan Amazon Virtual Private Cloud

Biasanya, AWS CodeBuild tidak dapat mengakses sumber daya dalam VPC. Untuk mengaktifkan akses, Anda harus memberikan informasi VPC-specific konfigurasi tambahan dalam konfigurasi CodeBuild proyek Anda. Ini termasuk ID VPC, ID subnet VPC, dan ID grup keamanan VPC. VPC-enabled build kemudian dapat mengakses sumber daya di dalam VPC Anda. Untuk informasi selengkapnya tentang menyiapkan VPC di Amazon VPC, lihat Panduan Pengguna Amazon VPC.

Topik

- Kasus penggunaan
- Praktik terbaik untuk VPC
- Keterbatasan VPC
- Izinkan akses Amazon VPC di proyek Anda CodeBuild
- Memecahkan masalah pengaturan VPC Anda
- Gunakan titik akhir VPC
- Gunakan AWS CodeBuild dengan server proxy terkelola
- Gunakan AWS CodeBuild dengan server proxy
- CloudFormation Template VPC

Kasus penggunaan

Konektivitas VPC dari AWS CodeBuild build memungkinkan untuk:

- Jalankan pengujian integrasi dari build Anda terhadap data dalam database Amazon RDS yang terisolasi di subnet pribadi.
- Kueri data di ElastiCache klaster Amazon langsung dari pengujian.
- Berinteraksi dengan layanan web internal yang dihosting di Amazon EC2, Amazon ECS, atau layanan yang menggunakan Elastic Load Balancing internal.
- Ambil dependensi dari repositori artefak internal yang dihosting sendiri, seperti PyPI untuk Python, Maven untuk Java, dan npm for. Node.js
- Akses objek dalam bucket S3 yang dikonfigurasi untuk mengizinkan akses melalui titik akhir VPC Amazon saja.

- Kueri layanan web eksternal yang memerlukan alamat IP tetap melalui alamat IP Elastis dari gateway NAT atau instance NAT yang terkait dengan subnet Anda.

Build Anda dapat mengakses sumber daya apa pun yang di-host di VPC Anda.

Praktik terbaik untuk VPC

Gunakan daftar periksa ini saat Anda menyiapkan VPC untuk dikerjakan. CodeBuild

- Siapkan VPC Anda dengan subnet publik dan pribadi, dan gateway NAT. Gateway NAT harus berada di subnet publik. Untuk informasi selengkapnya, lihat VPC dengan subnet publik dan pribadi (NAT) di Panduan Pengguna Amazon VPC.

  > **Important**
  >
  > Anda memerlukan gateway NAT atau instance NAT untuk digunakan CodeBuild dengan VPC Anda sehingga CodeBuild dapat mencapai titik akhir publik (misalnya, untuk menjalankan perintah CLI saat menjalankan build). Anda tidak dapat menggunakan gateway internet alih-alih gateway NAT atau instans NAT karena CodeBuild tidak mendukung penetapan alamat IP Elastis ke antarmuka jaringan yang dibuatnya, dan menetapkan alamat IP publik secara otomatis tidak didukung oleh Amazon EC2 untuk antarmuka jaringan apa pun yang dibuat di luar peluncuran instans Amazon EC2.

- Sertakan beberapa Availability Zone dengan VPC Anda.
- Pastikan grup keamanan Anda tidak memiliki lalu lintas masuk (ingress) yang diizinkan ke build Anda. CodeBuild tidak memiliki persyaratan khusus untuk lalu lintas keluar, tetapi Anda harus mengizinkan akses ke sumber daya Internet apa pun yang diperlukan untuk build Anda, seperti GitHub atau Amazon S3.

  Untuk informasi selengkapnya, lihat Aturan grup keamanan di Panduan Pengguna Amazon VPC.
- Siapkan subnet terpisah untuk build Anda.
- Ketika Anda mengatur CodeBuild proyek Anda untuk mengakses VPC Anda, pilih subnet pribadi saja.

Untuk informasi selengkapnya tentang menyiapkan VPC di Amazon VPC, lihat Panduan Pengguna Amazon VPC.

Untuk informasi selengkapnya tentang penggunaan CloudFormation untuk mengonfigurasi VPC agar menggunakan fitur CodeBuild VPC, lihat. CloudFormation Template VPC

Keterbatasan VPC

- Konektivitas VPC dari tidak CodeBuild didukung untuk VPC bersama.

Izinkan akses Amazon VPC di proyek Anda CodeBuild

Sertakan pengaturan ini dalam konfigurasi VPC Anda:

- Untuk ID VPC, pilih ID VPC yang digunakan. CodeBuild
- Untuk Subnet, pilih subnet pribadi dengan terjemahan NAT yang mencakup atau memiliki rute ke sumber daya yang digunakan oleh. CodeBuild
- Untuk Grup Keamanan, pilih grup keamanan yang CodeBuild digunakan untuk memungkinkan akses ke sumber daya di VPC.

Untuk menggunakan konsol untuk membuat proyek build, lihatBuat proyek build (konsol). Saat Anda membuat atau mengubah CodeBuild proyek, di VPC, pilih ID VPC, subnet, dan grup keamanan Anda.

Untuk menggunakan AWS CLI untuk membuat proyek build, lihatBuat proyek build (AWS CLI). Jika Anda menggunakan AWS CLI with CodeBuild, peran layanan yang digunakan oleh CodeBuild untuk berinteraksi dengan layanan atas nama pengguna IAM harus memiliki kebijakan yang dilampirkan. Untuk informasi, lihat Izinkan CodeBuild akses ke AWS layanan yang diperlukan untuk membuat antarmuka jaringan VPC.

*vpcConfig*Objek harus mencakup Anda*vpcId*,*securityGroupIds*, dan*subnets*.

- *vpcId*: Diperlukan. ID VPC yang CodeBuild menggunakan. Jalankan perintah ini untuk mendapatkan daftar semua ID VPC Amazon di Wilayah Anda:

```
aws ec2 describe-vpcs
```

- *subnets*: Diperlukan. ID subnet yang menyertakan sumber daya yang digunakan oleh CodeBuild. Jalankan perintah ini mendapatkan ID ini:

```
aws ec2 describe-subnets --filters "Name=vpc-id,Values=<vpc-id>" --region us-east-1
```

> **Note**
>
> Ganti `us-east-1` dengan Wilayah Anda.

- *`securityGroupIds`*: Diperlukan. ID grup keamanan yang digunakan oleh CodeBuild untuk memungkinkan akses ke sumber daya di VPC. Jalankan perintah ini untuk mendapatkan ID ini:

```
aws ec2 describe-security-groups --filters "Name=vpc-id,Values=<vpc-id>" --region us-
east-1
```

> **Note**
>
> Ganti `us-east-1` dengan Wilayah Anda.

Memecahkan masalah pengaturan VPC Anda

Gunakan informasi yang muncul dalam pesan kesalahan untuk membantu Anda mengidentifikasi, mendiagnosis, dan mengatasi masalah.

Berikut ini adalah beberapa panduan untuk membantu Anda saat memecahkan masalah kesalahan CodeBuild VPC umum:. `Build does not have internet connectivity. Please check subnet network configuration`

1. Pastikan gateway internet Anda terpasang ke VPC.
2. Pastikan bahwa tabel rute untuk subnet publik Anda menunjuk ke gateway internet.
3. Pastikan ACL jaringan Anda memungkinkan lalu lintas mengalir.
4. Pastikan grup keamanan Anda mengizinkan lalu lintas mengalir.
5. Memecahkan masalah gateway NAT Anda.
6. Pastikan bahwa tabel rute untuk subnet pribadi menunjuk ke gateway NAT.
7. Pastikan bahwa peran layanan yang digunakan oleh CodeBuild untuk berinteraksi dengan layanan atas nama pengguna IAM memiliki izin dalam kebijakan ini. Untuk informasi

selengkapnya, lihat Memungkinkan CodeBuild untuk berinteraksi dengan orang lain AWS layanan.

Jika izin tidak CodeBuild ada, Anda mungkin menerima kesalahan yang mengatakan,`Unexpected EC2 error: UnauthorizedOperation`. Kesalahan ini dapat terjadi jika CodeBuild tidak memiliki izin Amazon EC2 yang diperlukan untuk bekerja dengan VPC.

Gunakan titik akhir VPC

Anda dapat meningkatkan keamanan build Anda dengan mengonfigurasi AWS CodeBuild untuk menggunakan titik akhir VPC antarmuka. Endpoint antarmuka didukung oleh PrivateLink, teknologi yang dapat Anda gunakan untuk mengakses Amazon EC2 secara pribadi CodeBuild dan dengan menggunakan alamat IP pribadi. PrivateLink membatasi semua lalu lintas jaringan antara instans terkelola Anda CodeBuild, dan Amazon EC2 ke jaringan Amazon. (Instans terkelola tidak memiliki akses ke internet.) Selain itu, Anda tidak memerlukan gateway internet, perangkat NAT, atau gateway pribadi virtual. Anda tidak diwajibkan untuk mengonfigurasi PrivateLink, tetapi sebaiknya direkomendasikan. Untuk informasi selengkapnya tentang PrivateLink dan titik akhir VPC, lihat Apa itu? AWS PrivateLink .

Sebelum Anda membuat titik akhir VPC

Sebelum Anda mengonfigurasi titik akhir VPC AWS CodeBuild, perhatikan batasan dan batasan berikut.

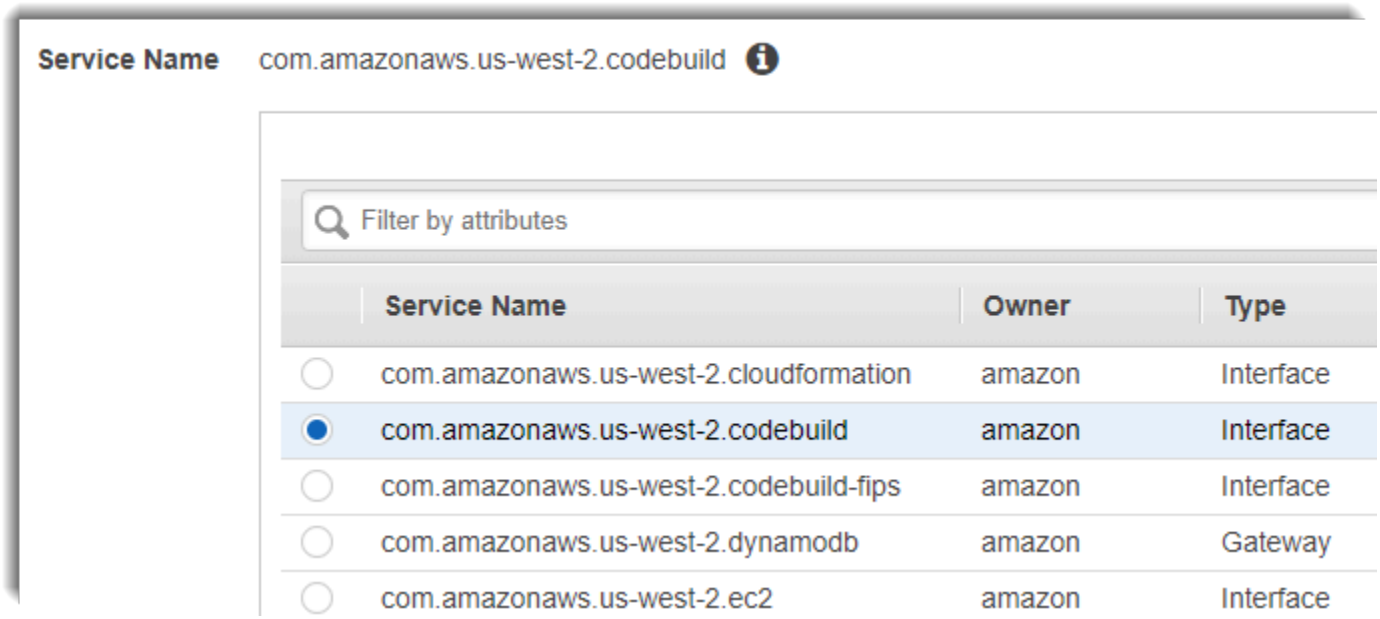
> **Note**
>
> Gunakan gateway NAT jika Anda ingin menggunakan CodeBuild dengan AWS layanan yang tidak mendukung koneksi Amazon PrivateLink VPC.

- Titik akhir VPC mendukung Amazon-provided DNS melalui Amazon Route 53 saja. Jika Anda ingin menggunakan DNS Anda sendiri, Anda dapat menggunakan penerusan DNS bersyarat. Untuk informasi selengkapnya, lihat Set opsi DHCP dalam Panduan Pengguna Amazon VPC.
- Titik akhir VPC saat ini tidak mendukung permintaan lintas wilayah. Pastikan Anda membuat titik akhir di AWS Wilayah yang sama dengan bucket S3 yang menyimpan input dan output build Anda. Anda dapat menggunakan konsol Amazon S3 atau perintah get-bucket-location untuk menemukan

lokasi bucket Anda. Gunakan endpoint Region-specific Amazon S3 untuk mengakses bucket Anda (misalnya,). `<bucket-name>.s3-us-west-2.amazonaws.com` Untuk informasi selengkapnya tentang Region-specific titik akhir Amazon S3, lihat Layanan Penyimpanan Sederhana Amazon di. Referensi Umum Amazon Web Jika Anda menggunakan AWS CLI untuk membuat permintaan ke Amazon S3, setel Wilayah default Anda ke Wilayah yang sama tempat bucket dibuat, atau gunakan `--region` parameter dalam permintaan Anda.

Buat titik akhir VPC untuk CodeBuild

Ikuti petunjuk dalam Membuat titik akhir antarmuka untuk membuat titik akhir`com.amazonaws.region.codebuild`. Ini adalah titik akhir VPC untuk. AWS CodeBuild



*region*mewakili pengenal wilayah untuk AWS Wilayah yang didukung oleh CodeBuild, seperti `us-east-2` untuk Wilayah Timur AS (Ohio). Untuk daftar AWS Wilayah yang didukung, lihat CodeBuilddi Referensi AWS Umum. Titik akhir diisi sebelumnya dengan Wilayah yang Anda tentukan saat Anda masuk. AWS Jika Anda mengubah Wilayah, titik akhir VPC akan diperbarui.

Membuat kebijakan titik akhir VPC untuk CodeBuild

Anda dapat membuat kebijakan untuk titik akhir VPC Amazon yang dapat Anda tentukan: AWS CodeBuild

- Prinsipal yang dapat melakukan tindakan.
- Tindakan yang dapat dilakukan.

- Sumber daya yang dapat memiliki tindakan yang dilakukan pada mereka.

Contoh kebijakan berikut menetapkan bahwa semua prinsipal hanya dapat memulai dan melihat build untuk proyek. `project-name`

```
{
    "Statement": [
        {
            "Action": [
                "codebuild:ListBuildsForProject",
                "codebuild:StartBuild",
                "codebuild:BatchGetBuilds"
            ],
            "Effect": "Allow",
            "Resource": "arn:aws:codebuild:region-ID:account-ID:project/project-name",
            "Principal": "*"
        }
    ]
}
```

Untuk informasi selengkapnya, lihat Mengontrol Akses ke Layanan dengan titik akhir VPC dalam Panduan Pengguna Amazon VPC.

Gunakan AWS CodeBuild dengan server proxy terkelola

Untuk menjalankan armada kapasitas AWS CodeBuild cadangan di server proxy terkelola, Anda harus mengonfigurasi server proxy untuk mengizinkan atau menolak lalu lintas ke dan dari situs eksternal menggunakan aturan proxy. Perhatikan bahwa menjalankan armada kapasitas cadangan di server proxy terkelola tidak didukung untuk VPC, Windows, atau macOS.

> **Important**
>
> Ada biaya tambahan berdasarkan durasi konfigurasi proxy yang ada di armada. Untuk informasi selengkapnya, lihat https://aws.amazon.com/codebuild/pricing/.

Topik

- Konfigurasikan konfigurasi proxy terkelola untuk armada kapasitas cadangan

- Jalankan armada kapasitas CodeBuild cadangan

Konfigurasikan konfigurasi proxy terkelola untuk armada kapasitas cadangan

Untuk mengonfigurasi server proxy terkelola untuk armada kapasitas cadangan, Anda harus mengaktifkan fitur ini saat membuat armada di konsol atau menggunakan AWS CLI. Ada beberapa properti yang perlu Anda definisikan:

Tentukan konfigurasi proxy - opsional

Konfigurasi proxy yang menerapkan kontrol akses jaringan ke instance kapasitas cadangan Anda.

Perilaku default

Mendefinisikan perilaku lalu lintas keluar.

Izinkan

Mengizinkan lalu lintas keluar ke semua tujuan secara default.

Menyangkal

Menolak lalu lintas keluar ke semua tujuan secara default.

Aturan proxy

Menentukan domain tujuan untuk membatasi kontrol akses jaringan ke.

Untuk menentukan konfigurasi proxy di konsol Anda, lihat Buat armada kapasitas yang dicadangkan petunjuknya. Untuk menentukan konfigurasi proxy menggunakan AWS CLI, Anda dapat melakukannya dengan memodifikasi sintaks JSON berikut dan menyimpan hasil Anda:

```
"proxyConfiguration": {
    "defaultBehavior": "ALLOW_ALL" | "DENY_ALL",
    "orderedProxyRules": [
        {
            "type": "DOMAIN" | "IP",
            "effect": "ALLOW" | "DENY",
            "entities": [
                "destination"
            ]
        }
```

```
    ]
}
```

File JSON Anda mungkin terlihat mirip dengan yang berikut ini:

```
"proxyConfiguration": {
    "defaultBehavior": "DENY_ALL",
    "orderedProxyRules": [
        {
            "type": "DOMAIN",
            "effect": "ALLOW",
            "entities": [
                "github.com"
            ]
        }
    ]
}
```

Jalankan armada kapasitas CodeBuild cadangan

Saat menjalankan armada kapasitas AWS CodeBuild cadangan dengan server proxy terkelola Anda, secara otomatis CodeBuild akan mengatur variabel `HTTP_PROXY` dan `HTTPS_PROXY` lingkungannya dengan alamat proxy terkelola. Jika perangkat lunak dependensi Anda memiliki konfigurasinya sendiri dan tidak mematuhi variabel lingkungan, Anda dapat merujuk ke nilai-nilai ini dan memperbarui konfigurasi perangkat lunak Anda dalam perintah build untuk merutekan lalu lintas build dengan benar melalui proxy terkelola. Lihat informasi yang lebih lengkap di Membuat proyek pembangunan di AWS CodeBuild dan Ubah pengaturan proyek build di AWS CodeBuild.

Gunakan AWS CodeBuild dengan server proxy

Anda dapat menggunakan AWS CodeBuild dengan server proxy untuk mengatur lalu lintas HTTP dan HTTPS ke dan dari internet. Untuk menjalankan CodeBuild dengan server proxy, Anda menginstal server proxy di subnet publik dan CodeBuild di subnet pribadi di VPC.

Ada dua kasus penggunaan utama untuk berjalan CodeBuild di server proxy:

- Ini menghilangkan penggunaan gateway NAT atau instance NAT di VPC Anda.
- Ini memungkinkan Anda menentukan URL yang dapat diakses oleh instance di server proxy dan URL yang ditolak aksesnya oleh server proxy.

Anda dapat menggunakan CodeBuild dengan dua jenis server proxy. Untuk keduanya, server proxy berjalan di subnet publik dan CodeBuild berjalan di subnet pribadi.

- Proksi eksplisit: Jika Anda menggunakan server proxy eksplisit, Anda harus mengkonfigurasi `NO_PROXY`HTTP_PROXY,, dan variabel `HTTPS_PROXY` lingkungan CodeBuild di tingkat proyek. Untuk informasi selengkapnya, lihat Ubah pengaturan proyek build di AWS CodeBuild dan Membuat proyek pembangunan di AWS CodeBuild.
- Proxy transparan: Jika Anda menggunakan server proxy transparan, tidak diperlukan konfigurasi khusus.

Topik

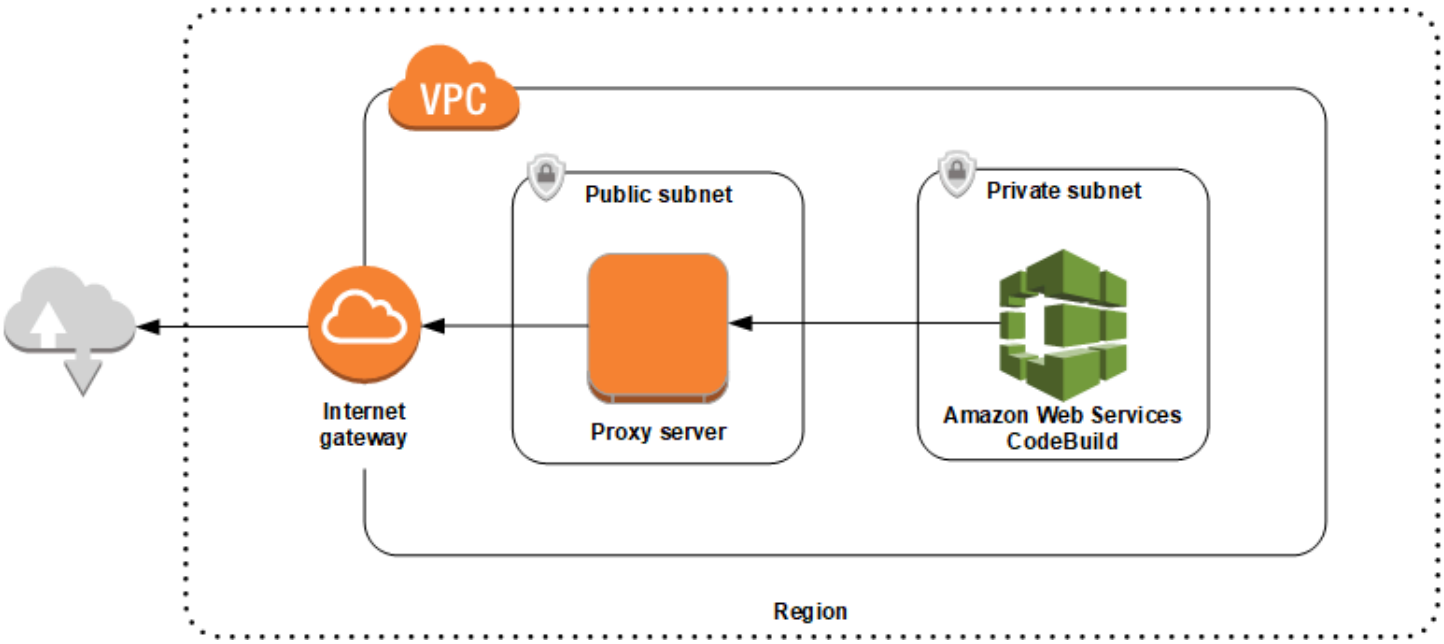
- Menyiapkan komponen yang diperlukan untuk berjalan CodeBuild di server proxy
- Jalan CodeBuild kan di server proxy eksplisit
- J CodeBuild alankan di server proxy transparan
- Jalankan manajer paket dan alat lainnya di server proxy

Menyiapkan komponen yang diperlukan untuk berjalan CodeBuild di server proxy

Anda memerlukan komponen ini untuk berjalan AWS CodeBuild di server proxy transparan atau eksplisit:

- SEBUAH VPC.
- Satu subnet publik di VPC Anda untuk server proxy.
- Satu subnet pribadi di VPC Anda untuk. CodeBuild
- Gerbang internet yang memungkinkan komunikasi antara VPC dan internet.

Diagram berikut menunjukkan bagaimana komponen berinteraksi.



Menyiapkan VPC, subnet, dan gateway jaringan

Langkah-langkah berikut diperlukan untuk berjalan AWS CodeBuild di server proxy transparan atau eksplisit.

1. Buat sebuah VPC. Untuk selengkapnya, lihat Membuat VPC di Panduan Pengguna Amazon VPC.
2. Buat dua subnet di VPC Anda. Salah satunya adalah subnet publik bernama `Public Subnet` di mana server proxy Anda berjalan. Yang lainnya adalah subnet pribadi bernama `Private Subnet` di mana CodeBuild berjalan.

   Untuk selengkapnya, lihat Membuat subnet di V PC Anda.
3. Buat dan lampirkan gateway internet ke VPC Anda. Untuk informasi selengkapnya, lihat Membuat dan melampirkan gateway internet.
4. Tambahkan aturan ke tabel rute default yang merutekan lalu lintas keluar dari VPC (0.0.0. 0/0) ke gateway internet. Untuk selengkapnya, lihat Men ambahkan dan menghapus rute dari tabel rute.
5. Tambahkan aturan ke grup keamanan default VPC Anda yang memungkinkan masuk lalu lintas SSH (TCP 22) dari VPC Anda (0.0.0. 0/0).
6. Ikuti petunjuk dalam Mel uncurkan instance menggunakan wizard instance peluncuran di Panduan Pengguna Amazon EC2 untuk meluncurkan instans Amazon Linux. Saat Anda menjalankan wizard, pilih opsi berikut:
   - Di Pilih Jenis Instans, pilih Amazon Linux Amazon Machine Image (AMI).

- Di Subnet, pilih subnet publik yang Anda buat sebelumnya dalam topik ini. Jika Anda menggunakan nama yang disarankan, itu adalah Subnet Publik.
- Di IP Auto-assign Publik, pilih Aktifkan.
- Pada halaman Konfigurasi Grup Keamanan, untuk Tetapkan grup keamanan, pilih Pilih grup keamanan yang ada. Selanjutnya, pilih grup keamanan default.
- Setelah Anda memilih Luncur kan, pilih pasangan kunci yang ada atau buat satu.

Pilih pengaturan default untuk semua opsi lainnya.

7. Setelah instans EC2 Anda berjalan, nonaktifkan source/destination pemeriksaan. Untuk informasi, lihat Men onaktifkan Source/Destination pemeriksaan di Panduan Pengguna Amazon VPC.
8. Buat tabel rute di VPC Anda. Tambahkan aturan ke tabel rute yang merutekan lalu lintas yang ditujukan untuk internet ke server proxy Anda. Kaitkan tabel rute ini dengan subnet pribadi Anda. Ini diperlukan agar permintaan keluar dari instance di subnet pribadi Anda, di mana CodeBuild berjalan, selalu dirutekan melalui server proxy.

Instal dan konfigurasikan server proxy

Ada banyak server proxy yang bisa dipilih. Server proxy open-source, Squid, digunakan di sini untuk menunjukkan bagaimana AWS CodeBuild berjalan di server proxy. Anda dapat menerapkan konsep yang sama ke server proxy lainnya.

Untuk menginstal Squid, gunakan repo yum dengan menjalankan perintah berikut:

```
sudo yum update -y
sudo yum install -y squid
```

Setelah Anda menginstal Squid, edit `squid.conf` file-nya menggunakan instruksi nanti dalam topik ini.

Konfigurasikan Squid untuk lalu lintas HTTPS

Untuk HTTPS, lalu lintas HTTP dienkapsulasi dalam koneksi Transport Layer Security (TLS). Squid menggunakan fitur yang di SslPeekAndSplice panggil untuk mengambil Server Name Indication (SNI) dari inisiasi TLS yang berisi host internet yang diminta. Ini diperlukan agar Squid tidak perlu membuka enkripsi lalu lintas HTTPS. Untuk mengaktifkan SslPeekAndSplice, Squid memerlukan sertifikat. Buat sertifikat ini menggunakan OpenSSL:

```
sudo mkdir /etc/squid/ssl
cd /etc/squid/ssl
sudo openssl genrsa -out squid.key 2048
sudo openssl req -new -key squid.key -out squid.csr -subj "/C=XX/ST=XX/L=squid/O=squid/
CN=squid"
sudo openssl x509 -req -days 3650 -in squid.csr -signkey squid.key -out squid.crt
sudo cat squid.key squid.crt | sudo tee squid.pem
```

> **Note**
>
> Untuk HTTP, Squid tidak memerlukan konfigurasi. Dari semua pesan HTTP/1.1 permintaan, ia dapat mengambil bidang header host, yang menentukan host internet yang diminta.

Jalan CodeBuild kan di server proxy eksplisit

Untuk berjalan AWS CodeBuild di server proxy eksplisit, Anda harus mengonfigurasi server proxy untuk mengizinkan atau menolak lalu lintas ke dan dari situs eksternal, dan kemudian mengkonfigurasi variabel `HTTP_PROXY` dan `HTTPS_PROXY` lingkungan.

Topik

- Konfigurasikan Squid sebagai server proxy eksplisit
- Buat CodeBuild proyek
- Contoh file squid .conf server proxy eksplisit

Konfigurasikan Squid sebagai server proxy eksplisit

Untuk mengonfigurasi server proxy Squid agar eksplisit, Anda harus membuat modifikasi berikut pada `/etc/squid/squid.conf` file-nya:

- Hapus aturan daftar kontrol akses default (ACL) berikut.

```
acl localnet src 10.0.0.0/8
acl localnet src 172.16.0.0/12
acl localnet src 192.168.0.0/16
acl localnet src fc00::/7
acl localnet src fe80::/10
```

Tambahkan yang berikut ini sebagai pengganti aturan ACL default yang Anda hapus. Baris pertama memungkinkan permintaan dari VPC Anda. Dua baris berikutnya memberikan akses server proxy Anda ke URL tujuan yang mungkin digunakan oleh AWS CodeBuild. Edit ekspresi reguler di baris terakhir untuk menentukan bucket S3 atau CodeCommit repositori di Wil AWS ayah. Contoh:

- Jika sumber Anda adalah Amazon S3, gunakan perintah acl download_src dstdom_regex .*s3\.us-west-1\.amazonaws\.com untuk memberikan akses ke bucket S3 di Wil `us-west-1` ayah.
- Jika sumber Anda adalah AWS CodeCommit, gunakan `git-codecommit.<your-region>.amazonaws.com` untuk menambahkan AWS Wilayah ke daftar izin.

```
acl localnet src 10.1.0.0/16 #Only allow requests from within the VPC
acl allowed_sites dstdomain .github.com #Allows to download source from GitHub
acl allowed_sites dstdomain .bitbucket.com #Allows to download source from Bitbucket
acl download_src dstdom_regex .*\.amazonaws\.com #Allows to download source from
 Amazon S3 or CodeCommit
```

- Ganti `http_access allow localnet` dengan yang berikut:

```
http_access allow localnet allowed_sites
http_access allow localnet download_src
```

- Jika Anda ingin build Anda mengunggah log dan artefak, lakukan salah satu hal berikut:

  1. Sebelum `http_access deny all` pernyataan, masukkan pernyataan berikut. Mereka memungkinkan CodeBuild untuk mengakses CloudWatch dan Amazon S3. Ak CloudWatch ses ke diperlukan agar CodeBuild dapat membuat CloudWatch log. Akses ke Amazon S3 diperlukan untuk mengunggah artefak dan caching Amazon S3.

     -
     ```
     https_port 3130 cert=/etc/squid/ssl/squid.pem ssl-bump intercept
     acl SSL_port port 443
     http_access allow SSL_port
     acl allowed_https_sites ssl::server_name .amazonaws.com
     acl step1 at_step SslBump1
     acl step2 at_step SslBump2
     acl step3 at_step SslBump3
     ssl_bump peek step1 all
     ssl_bump peek step2 allowed_https_sites
     ssl_bump splice step3 allowed_https_sites
     ssl_bump terminate step2 all
     ```

- Setelah Anda menyimpan`squid.conf`, jalankan perintah berikut:

```
sudo iptables -t nat -A PREROUTING -p tcp --dport 443 -j REDIRECT --to-port 3130
sudo service squid restart
```

2. Tambahkan `proxy` ke file buildspec Anda. Untuk informasi selengkapnya, lihat Sintaks Buildspec.

```
version: 0.2
proxy:
  upload-artifacts: yes
  logs: yes
phases:
  build:
    commands:
      - command
```

> **Note**
>
> Jika Anda menerima kesalahan RequestError batas waktu, lihatRequestError kesalahan batas waktu saat berjalan CodeBuild di server proxy.

Untuk informasi selengkapnya, lihat Contoh file `squid` .conf server proxy eksplisit dalam topik ini.

Buat CodeBuild proyek

Untuk menjalankan AWS CodeBuild dengan server proxy eksplisit Anda, atur variabel `HTTP_PROXY` dan `HTTPS_PROXY` lingkungan dengan alamat IP pribadi dari instance EC2 yang Anda buat untuk server proxy dan port 3128 di tingkat proyek. Alamat IP pribadi terlihat seperti`http://`*`your-ec2-private-ip-address`*`:3128`. Untuk informasi selengkapnya, lihat Membuat proyek pembangunan di AWS CodeBuild dan Ubah pengaturan proyek build di AWS CodeBuild.

Gunakan perintah berikut untuk melihat log akses proxy Squid:

```
sudo tail -f /var/log/squid/access.log
```

Contoh file **squid** .conf server proxy eksplisit

Berikut ini adalah contoh `squid.conf` file yang dikonfigurasi untuk server proxy eksplisit.

```
 acl localnet src 10.0.0.0/16 #Only allow requests from within the VPC
 # add all URLS to be whitelisted for download source and commands to be run in build
environment
 acl allowed_sites dstdomain .github.com    #Allows to download source from github
 acl allowed_sites dstdomain .bitbucket.com #Allows to download source from bitbucket
 acl allowed_sites dstdomain ppa.launchpad.net #Allows to run apt-get in build
environment
 acl download_src dstdom_regex .*\.amazonaws\.com #Allows to download source from S3
or CodeCommit
 acl SSL_ports port 443
 acl Safe_ports port 80  # http
 acl Safe_ports port 21  # ftp
 acl Safe_ports port 443  # https
 acl Safe_ports port 70  # gopher
 acl Safe_ports port 210  # wais
 acl Safe_ports port 1025-65535 # unregistered ports
 acl Safe_ports port 280  # http-mgmt
 acl Safe_ports port 488  # gss-http
 acl Safe_ports port 591  # filemaker
 acl Safe_ports port 777  # multiling http
 acl CONNECT method CONNECT
 #
 # Recommended minimum Access Permission configuration:
 #
 # Deny requests to certain unsafe ports
 http_access deny !Safe_ports
 # Deny CONNECT to other than secure SSL ports
 http_access deny CONNECT !SSL_ports
 # Only allow cachemgr access from localhost
 http_access allow localhost manager
 http_access deny manager
 # We strongly recommend the following be uncommented to protect innocent
 # web applications running on the proxy server who think the only
 # one who can access services on "localhost" is a local user
 #http_access deny to_localhost
 #
 # INSERT YOUR OWN RULE(S) HERE TO ALLOW ACCESS FROM YOUR CLIENTS
 #
 # Example rule allowing access from your local networks.
 # Adapt localnet in the ACL section to list your (internal) IP networks
```

```
  # from where browsing should be allowed
  http_access allow localnet allowed_sites
  http_access allow localnet download_src
  http_access allow localhost
  # Add this for CodeBuild to access CWL end point, caching and upload artifacts S3
bucket end point
  https_port 3130 cert=/etc/squid/ssl/squid.pem ssl-bump intercept
  acl SSL_port port 443
  http_access allow SSL_port
  acl allowed_https_sites ssl::server_name .amazonaws.com
  acl step1 at_step SslBump1
  acl step2 at_step SslBump2
  acl step3 at_step SslBump3
  ssl_bump peek step1 all
  ssl_bump peek step2 allowed_https_sites
  ssl_bump splice step3 allowed_https_sites
  ssl_bump terminate step2 all
  # And finally deny all other access to this proxy
  http_access deny all
  # Squid normally listens to port 3128
  http_port 3128
  # Uncomment and adjust the following to add a disk cache directory.
  #cache_dir ufs /var/spool/squid 100 16 256
  # Leave coredumps in the first cache dir
  coredump_dir /var/spool/squid
  #
  # Add any of your own refresh_pattern entries above these.
  #
  refresh_pattern ^ftp:  1440 20% 10080
  refresh_pattern ^gopher: 1440 0% 1440
  refresh_pattern -i (/cgi-bin/|\?) 0 0% 0
  refresh_pattern .  0 20% 4320
```

J CodeBuild alankan di server proxy transparan

Untuk berjalan AWS CodeBuild di server proxy transparan, Anda harus mengkonfigurasi server proxy dengan akses ke situs web dan domain yang berinteraksi dengannya.

Topik

- Konfigurasikan Squid sebagai server proxy transparan
- Buat CodeBuild proyek

Konfigurasikan Squid sebagai server proxy transparan

Untuk mengonfigurasi server proxy agar transparan, Anda harus memberinya akses ke domain dan situs web yang ingin Anda akses. Untuk menjalankan AWS CodeBuild dengan server proxy transparan, Anda harus memberinya akses ke`amazonaws.com`. Anda juga harus memberikan akses ke CodeBuild penggunaan situs web lain. Ini bervariasi, tergantung pada bagaimana Anda membuat CodeBuild proyek Anda. Contoh situs web adalah untuk repositori seperti GitHub, Bitbucket, Yum, dan Maven. Untuk memberikan Squid akses ke domain dan situs web tertentu, gunakan perintah yang mirip dengan berikut ini untuk memperbarui `squid.conf` file. Perintah sampel ini memberikan akses ke`amazonaws.com`,`github.com`, dan`bitbucket.com`. Anda dapat mengedit sampel ini untuk memberikan akses ke situs web lain.

```
cat | sudo tee /etc/squid/squid.conf #EOF
visible_hostname squid
#Handling HTTP requests
http_port 3129 intercept
acl allowed_http_sites dstdomain .amazonaws.com
#acl allowed_http_sites dstdomain domain_name [uncomment this line to add another
 domain]
http_access allow allowed_http_sites
#Handling HTTPS requests
https_port 3130 cert=/etc/squid/ssl/squid.pem ssl-bump intercept
acl SSL_port port 443
http_access allow SSL_port
acl allowed_https_sites ssl::server_name .amazonaws.com
acl allowed_https_sites ssl::server_name .github.com
acl allowed_https_sites ssl::server_name .bitbucket.com
#acl allowed_https_sites ssl::server_name [uncomment this line to add another website]
acl step1 at_step SslBump1
acl step2 at_step SslBump2
acl step3 at_step SslBump3
ssl_bump peek step1 all
ssl_bump peek step2 allowed_https_sites
ssl_bump splice step3 allowed_https_sites
ssl_bump terminate step2 all
http_access deny all
EOF
```

Permintaan masuk dari instance di subnet pribadi harus dialihkan ke port Squid. Squid mendengarkan pada port 3129 untuk lalu lintas HTTP (bukan 80) dan 3130 untuk lalu lintas HTTPS (bukan 443). Gunakan iptables perintah untuk merutekan lalu lintas:

```
sudo iptables -t nat -A PREROUTING -p tcp --dport 80 -j REDIRECT --to-port 3129
sudo iptables -t nat -A PREROUTING -p tcp --dport 443 -j REDIRECT --to-port 3130
sudo service iptables save
sudo service squid start
```

Buat CodeBuild proyek

Setelah Anda mengkonfigurasi server proxy Anda, Anda dapat menggunakannya dengan AWS CodeBuild subnet pribadi tanpa konfigurasi lebih lanjut. Setiap permintaan HTTP dan HTTPS melewati server proxy publik. Gunakan perintah berikut untuk melihat log akses proxy Squid:

```
sudo tail -f /var/log/squid/access.log
```

Jalankan manajer paket dan alat lainnya di server proxy

Gunakan prosedur berikut untuk menjalankan manajer paket dan alat lainnya di server proxy.

Untuk menjalankan alat, seperti manajer paket, di server proxy

1. Tambahkan alat ke daftar izinkan di server proxy Anda dengan menambahkan pernyataan ke `squid.conf` file Anda.
2. Tambahkan baris ke file buildspec Anda yang menunjuk ke titik akhir pribadi server proxy Anda.

Contoh berikut menunjukkan bagaimana melakukan ini untuk`apt-get`,`curl`, dan`maven`. Jika Anda menggunakan alat yang berbeda, prinsip yang sama berlaku. Tambahkan ke daftar izinkan dalam `squid.conf` file dan tambahkan perintah ke file buildspec Anda untuk mengetahui titik CodeBuild akhir server proxy Anda.

Untuk menjalankan **`apt-get`** di server proxy

1. Tambahkan pernyataan berikut ke `squid.conf` file Anda untuk ditambahkan `apt-get` ke daftar izinkan di server proxy Anda. Tiga baris pertama memungkinkan `apt-get` untuk berjalan di lingkungan build.

   ```
   acl allowed_sites dstdomain ppa.launchpad.net # Required for apt-get to run in the
    build environment
   acl apt_get dstdom_regex .*\.launchpad.net # Required for CodeBuild to run apt-get
    in the build environment
   ```

```
acl apt_get dstdom_regex .*\.ubuntu.com    # Required for CodeBuild to run apt-get
 in the build environment
http_access allow localnet allowed_sites
http_access allow localnet apt_get
```

2. Tambahkan pernyataan berikut di file buildspec Anda sehingga `apt-get` perintah mencari konfigurasi proxy di. `/etc/apt/apt.conf.d/00proxy`

```
echo 'Acquire::http::Proxy "http://<private-ip-of-proxy-server>:3128";
 Acquire::https::Proxy "http://<private-ip-of-proxy-server>:3128";
 Acquire::ftp::Proxy "http://<private-ip-of-proxy-server>:3128";' > /etc/apt/
apt.conf.d/00proxy
```

Untuk menjalankan **`curl`** di server proxy

1. Tambahkan yang berikut ini ke `squid.conf` file Anda untuk ditambahkan `curl` ke daftar izinkan di lingkungan build Anda.

```
acl allowed_sites dstdomain ppa.launchpad.net # Required to run apt-get in the
 build environment
acl allowed_sites dstdomain google.com # Required for access to a webiste. This
 example uses www.google.com.
http_access allow localnet allowed_sites
http_access allow localnet apt_get
```

2. Tambahkan pernyataan berikut dalam file buildspec Anda sehingga `curl` menggunakan server proxy pribadi untuk mengakses situs web yang Anda tambahkan ke. `squid.conf` Dalam contoh ini, situs web adalah`google.com`.

```
curl -x <private-ip-of-proxy-server>:3128 https://www.google.com
```

Untuk menjalankan **`maven`** di server proxy

1. Tambahkan yang berikut ini ke `squid.conf` file Anda untuk ditambahkan `maven` ke daftar izinkan di lingkungan build Anda.

```
acl allowed_sites dstdomain ppa.launchpad.net # Required to run apt-get in the
 build environment
```

```
acl maven dstdom_regex .*\.maven.org # Allows access to the maven repository in the
 build environment
http_access allow localnet allowed_sites
http_access allow localnet maven
```

2. Tambahkan pernyataan berikut ke file buildspec Anda.

```
maven clean install -DproxySet=true -DproxyHost=<private-ip-of-proxy-server> -
DproxyPort=3128
```

CloudFormation Template VPC

CloudFormation memungkinkan Anda membuat dan menyediakan penerapan AWS infrastruktur secara dapat diprediksi dan berulang kali, dengan menggunakan file template untuk membuat dan menghapus kumpulan sumber daya bersama-sama sebagai satu unit (tumpukan). Untuk informasi selengkapnya, silakan lihat Panduan Pengguna CloudFormation.

Berikut ini adalah template CloudFormation YAML untuk mengonfigurasi VPC untuk digunakan. AWS CodeBuild File ini juga tersedia di samples.zip.

```
Description:  This template deploys a VPC, with a pair of public and private subnets
 spread
  across two Availability Zones. It deploys an internet gateway, with a default
  route on the public subnets. It deploys a pair of NAT gateways (one in each AZ),
  and default routes for them in the private subnets.

Parameters:
  EnvironmentName:
    Description: An environment name that is prefixed to resource names
    Type: String

  VpcCIDR:
    Description: Please enter the IP range (CIDR notation) for this VPC
    Type: String
    Default: 10.192.0.0/16

  PublicSubnet1CIDR:
    Description: Please enter the IP range (CIDR notation) for the public subnet in the
 first Availability Zone
    Type: String
    Default: 10.192.10.0/24
```

```
  PublicSubnet2CIDR:
    Description: Please enter the IP range (CIDR notation) for the public subnet in the
 second Availability Zone
    Type: String
    Default: 10.192.11.0/24

  PrivateSubnet1CIDR:
    Description: Please enter the IP range (CIDR notation) for the private subnet in
 the first Availability Zone
    Type: String
    Default: 10.192.20.0/24

  PrivateSubnet2CIDR:
    Description: Please enter the IP range (CIDR notation) for the private subnet in
 the second Availability Zone
    Type: String
    Default: 10.192.21.0/24

Resources:
  VPC:
    Type: AWS::EC2::VPC
    Properties:
      CidrBlock: !Ref VpcCIDR
      EnableDnsSupport: true
      EnableDnsHostnames: true
      Tags:
        - Key: Name
          Value: !Ref EnvironmentName

  InternetGateway:
    Type: AWS::EC2::InternetGateway
    Properties:
      Tags:
        - Key: Name
          Value: !Ref EnvironmentName

  InternetGatewayAttachment:
    Type: AWS::EC2::VPCGatewayAttachment
    Properties:
      InternetGatewayId: !Ref InternetGateway
      VpcId: !Ref VPC

  PublicSubnet1:
```

```
    Type: AWS::EC2::Subnet
    Properties:
      VpcId: !Ref VPC
      AvailabilityZone: !Select [ 0, !GetAZs '' ]
      CidrBlock: !Ref PublicSubnet1CIDR
      MapPublicIpOnLaunch: true
      Tags:
        - Key: Name
          Value: !Sub ${EnvironmentName} Public Subnet (AZ1)

  PublicSubnet2:
    Type: AWS::EC2::Subnet
    Properties:
      VpcId: !Ref VPC
      AvailabilityZone: !Select [ 1, !GetAZs  '' ]
      CidrBlock: !Ref PublicSubnet2CIDR
      MapPublicIpOnLaunch: true
      Tags:
        - Key: Name
          Value: !Sub ${EnvironmentName} Public Subnet (AZ2)

  PrivateSubnet1:
    Type: AWS::EC2::Subnet
    Properties:
      VpcId: !Ref VPC
      AvailabilityZone: !Select [ 0, !GetAZs  '' ]
      CidrBlock: !Ref PrivateSubnet1CIDR
      MapPublicIpOnLaunch: false
      Tags:
        - Key: Name
          Value: !Sub ${EnvironmentName} Private Subnet (AZ1)

  PrivateSubnet2:
    Type: AWS::EC2::Subnet
    Properties:
      VpcId: !Ref VPC
      AvailabilityZone: !Select [ 1, !GetAZs  '' ]
      CidrBlock: !Ref PrivateSubnet2CIDR
      MapPublicIpOnLaunch: false
      Tags:
        - Key: Name
          Value: !Sub ${EnvironmentName} Private Subnet (AZ2)

  NatGateway1EIP:
```

```
    Type: AWS::EC2::EIP
    DependsOn: InternetGatewayAttachment
    Properties:
      Domain: vpc

  NatGateway2EIP:
    Type: AWS::EC2::EIP
    DependsOn: InternetGatewayAttachment
    Properties:
      Domain: vpc

  NatGateway1:
    Type: AWS::EC2::NatGateway
    Properties:
      AllocationId: !GetAtt NatGateway1EIP.AllocationId
      SubnetId: !Ref PublicSubnet1

  NatGateway2:
    Type: AWS::EC2::NatGateway
    Properties:
      AllocationId: !GetAtt NatGateway2EIP.AllocationId
      SubnetId: !Ref PublicSubnet2

  PublicRouteTable:
    Type: AWS::EC2::RouteTable
    Properties:
      VpcId: !Ref VPC
      Tags:
        - Key: Name
          Value: !Sub ${EnvironmentName} Public Routes

  DefaultPublicRoute:
    Type: AWS::EC2::Route
    DependsOn: InternetGatewayAttachment
    Properties:
      RouteTableId: !Ref PublicRouteTable
      DestinationCidrBlock: 0.0.0.0/0
      GatewayId: !Ref InternetGateway

  PublicSubnet1RouteTableAssociation:
    Type: AWS::EC2::SubnetRouteTableAssociation
    Properties:
      RouteTableId: !Ref PublicRouteTable
      SubnetId: !Ref PublicSubnet1
```

```
  PublicSubnet2RouteTableAssociation:
    Type: AWS::EC2::SubnetRouteTableAssociation
    Properties:
      RouteTableId: !Ref PublicRouteTable
      SubnetId: !Ref PublicSubnet2


  PrivateRouteTable1:
    Type: AWS::EC2::RouteTable
    Properties:
      VpcId: !Ref VPC
      Tags:
        - Key: Name
          Value: !Sub ${EnvironmentName} Private Routes (AZ1)

  DefaultPrivateRoute1:
    Type: AWS::EC2::Route
    Properties:
      RouteTableId: !Ref PrivateRouteTable1
      DestinationCidrBlock: 0.0.0.0/0
      NatGatewayId: !Ref NatGateway1

  PrivateSubnet1RouteTableAssociation:
    Type: AWS::EC2::SubnetRouteTableAssociation
    Properties:
      RouteTableId: !Ref PrivateRouteTable1
      SubnetId: !Ref PrivateSubnet1

  PrivateRouteTable2:
    Type: AWS::EC2::RouteTable
    Properties:
      VpcId: !Ref VPC
      Tags:
        - Key: Name
          Value: !Sub ${EnvironmentName} Private Routes (AZ2)

  DefaultPrivateRoute2:
    Type: AWS::EC2::Route
    Properties:
      RouteTableId: !Ref PrivateRouteTable2
      DestinationCidrBlock: 0.0.0.0/0
      NatGatewayId: !Ref NatGateway2
```

```
  PrivateSubnet2RouteTableAssociation:
    Type: AWS::EC2::SubnetRouteTableAssociation
    Properties:
      RouteTableId: !Ref PrivateRouteTable2
      SubnetId: !Ref PrivateSubnet2

  NoIngressSecurityGroup:
    Type: AWS::EC2::SecurityGroup
    Properties:
      GroupName: "no-ingress-sg"
      GroupDescription: "Security group with no ingress rule"
      VpcId: !Ref VPC

Outputs:
  VPC:
    Description: A reference to the created VPC
    Value: !Ref VPC

  PublicSubnets:
    Description: A list of the public subnets
    Value: !Join [ ",", [ !Ref PublicSubnet1, !Ref PublicSubnet2 ]]

  PrivateSubnets:
    Description: A list of the private subnets
    Value: !Join [ ",", [ !Ref PrivateSubnet1, !Ref PrivateSubnet2 ]]

  PublicSubnet1:
    Description: A reference to the public subnet in the 1st Availability Zone
    Value: !Ref PublicSubnet1

  PublicSubnet2:
    Description: A reference to the public subnet in the 2nd Availability Zone
    Value: !Ref PublicSubnet2

  PrivateSubnet1:
    Description: A reference to the private subnet in the 1st Availability Zone
    Value: !Ref PrivateSubnet1

  PrivateSubnet2:
    Description: A reference to the private subnet in the 2nd Availability Zone
    Value: !Ref PrivateSubnet2

  NoIngressSecurityGroup:
    Description: Security group with no ingress rule
```

```
    Value: !Ref NoIngressSecurityGroup
```

Penebangan dan pemantauan di AWS CodeBuild

Pencatatan dan pemantauan adalah bagian penting untuk menjaga keandalan, ketersediaan, dan kinerja AWS CodeBuild dan AWS solusi Anda. Anda harus mengumpulkan data pemantauan dari semua bagian AWS solusi Anda sehingga Anda dapat lebih mudah men-debug kegagalan multi-titik, jika terjadi. AWS menyediakan alat-alat berikut untuk memantau CodeBuild sumber daya dan membangun Anda dan untuk menanggapi potensi insiden.

Topik

- Log AWS CodeBuild Panggilan API dengan AWS CloudTrail
- Memantau CodeBuild build dengan CloudWatch

Log AWS CodeBuild Panggilan API dengan AWS CloudTrail

AWS CodeBuild terintegrasi dengan AWS CloudTrail, layanan yang menyediakan catatan tindakan yang diambil oleh pengguna, peran, atau AWS layanan di CodeBuild. CloudTrail menangkap semua panggilan API untuk acara CodeBuild as, termasuk panggilan dari CodeBuild konsol dan dari panggilan kode ke CodeBuild API. Jika Anda membuat jejak, Anda dapat mengaktifkan pengiriman CloudTrail acara secara terus menerus ke bucket S3, termasuk acara untuk CodeBuild. Jika Anda tidak mengonfigurasi jejak, Anda masih dapat melihat peristiwa terbaru di CloudTrail konsol di Ri wayat acara. Dengan menggunakan informasi yang dikumpulkan oleh CloudTrail, Anda dapat menentukan permintaan yang dibuat CodeBuild, alamat IP dari mana permintaan dibuat, siapa yang membuat permintaan, kapan itu dibuat, dan detail tambahan.

Untuk mempelajari selengkap CloudTrail nya, lihat Panduan AWS CloudTrail Pengguna.

Topik

- Tentang AWS CodeBuild informasi di CloudTrail
- Tentang AWS CodeBuild entri file log

Tentang AWS CodeBuild informasi di CloudTrail

CloudTrail diaktifkan pada AWS akun Anda saat Anda membuat akun. Ketika aktivitas terjadi di CodeBuild, aktivitas tersebut dicatat dalam suatu CloudTrail peristiwa bersama dengan peristiwa AWS layanan lainnya di ri wayat acara. Anda dapat melihat, mencari, dan mengunduh acara terbaru

di AWS akun Anda. Untuk informasi selengkapnya, lihat Melihat CloudTrail acara dengan riwayat acara di Panduan AWS CloudTrail Pengguna.

Untuk catatan acara yang berkelanjutan di AWS akun Anda, termasuk acara untuk CodeBuild, buat jejak. Jejak memungkinkan CloudTrail untuk mengirimkan file log ke bucket S3. Secara default, ketika Anda membuat jejak di konsol, jejak ini diterapkan ke semua Wilayah. Jejak mencatat peristiwa dari semua wilayah di AWS partisi dan mengirimkan file log ke bucket S3 yang Anda tentukan. Anda dapat mengonfigurasi AWS layanan lain untuk menganalisis lebih lanjut dan bertindak berdasarkan data peristiwa yang dikumpulkan dalam CloudTrail log. Untuk informasi selengkapnya, lihat:

- Ikhtisar untuk membuat jejak
- CloudTrail layanan dan integrasi yang didukung
- Mengkonfigurasi notifikasi Amazon SNS untuk CloudTrail
- M CloudTrail enerima file log dari beberapa wilayah dan Mener CloudTrail ima file log dari beberapa akun

Semua CodeBuild tindakan dicatat oleh CloudTrail dan didokumentasikan dalam Refer CodeBuild ensi API. Misalnya, panggilan ke tindakan `CreateProject` (in the AWS CLI,`create-project`), `StartBuild` (in the AWS CLI,`start-project`), dan `UpdateProject` (in the AWS CLI,`update-project`) menghasilkan entri dalam file CloudTrail log.

Setiap entri peristiwa atau log berisi informasi tentang entitas yang membuat permintaan tersebut. Informasi identitas membantu Anda menentukan hal berikut:

- Baik permintaan tersebut dibuat dengan kredensial pengguna atau root.
- Apakah permintaan tersebut dibuat dengan kredensial keamanan sementara untuk satu peran atau pengguna gabungan.
- Apakah permintaan itu dibuat oleh AWS layanan lain.

Untuk informasi selengkapnya, lihat elemen CloudTrail UserIdenti ty di Panduan AWS CloudTrail Pengguna.

Tentang AWS CodeBuild entri file log

Trail adalah konfigurasi yang memungkinkan pengiriman peristiwa sebagai file log ke bucket S3 yang Anda tentukan. CloudTrail file log berisi satu atau lebih entri log. Peristiwa mewakili permintaan tunggal dari sumber apa pun dan mencakup informasi tentang tindakan yang diminta, tanggal dan

waktu tindakan, parameter permintaan, dan sebagainya. CloudTrail file log bukanlah jejak tumpukan yang diurutkan dari panggilan API publik, jadi file tersebut tidak muncul dalam urutan tertentu.

> **Note**
>
> Untuk melindungi informasi sensitif, berikut ini disembunyikan di CodeBuild log:
>
> - AWS ID kunci akses. Untuk informasi selengkapnya, lihat Mengelola Kunci Akses untuk Pengguna IAM di Panduan AWS Identity and Access Management Pengguna.
> - String ditentukan menggunakan Parameter Store. Untuk informasi selengkapnya, lihat Panduan Konsol Penyimpanan Parameter Manajer Sistem dan Panduan Konsol Penyimpanan Parameter Manajer Sistem di Panduan Pengguna Amazon EC2 Systems Manager.
> - String ditentukan menggunakan AWS Secrets Manager. Untuk informasi selengkapnya, lihat Manajemen kunci.

Contoh berikut menunjukkan entri CloudTrail log yang menunjukkan pembuatan proyek build di CodeBuild.

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "FederatedUser",
    "principalId": "account-ID:user-name",
    "arn": "arn:aws:sts::account-ID:federated-user/user-name",
    "accountId": "account-ID",
    "accessKeyId": "access-key-ID",
    "sessionContext": {
      "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "2016-09-06T17:59:10Z"
      },
      "sessionIssuer": {
        "type": "IAMUser",
        "principalId": "access-key-ID",
        "arn": "arn:aws:iam::account-ID:user/user-name",
        "accountId": "account-ID",
        "userName": "user-name"
      }
    }
```

```
  },
  "eventTime": "2016-09-06T17:59:11Z",
  "eventSource": "codebuild.amazonaws.com",
  "eventName": "CreateProject",
  "awsRegion": "region-ID",
  "sourceIPAddress": "127.0.0.1",
  "userAgent": "user-agent",
  "requestParameters": {
    "awsActId": "account-ID"
  },
  "responseElements": {
    "project": {
      "environment": {
        "image": "image-ID",
        "computeType": "BUILD_GENERAL1_SMALL",
        "type": "LINUX_CONTAINER",
        "environmentVariables": []
      },
      "name": "codebuild-demo-project",
      "description": "This is my demo project",
      "arn": "arn:aws:codebuild:region-ID:account-ID:project/codebuild-demo-
project:project-ID",
      "encryptionKey": "arn:aws:kms:region-ID:key-ID",
      "timeoutInMinutes": 10,
      "artifacts": {
        "location": "arn:aws:s3:::codebuild-region-ID-account-ID-output-bucket",
        "type": "S3",
        "packaging": "ZIP",
        "outputName": "MyOutputArtifact.zip"
      },
      "serviceRole": "arn:aws:iam::account-ID:role/CodeBuildServiceRole",
      "lastModified": "Sep 6, 2016 10:59:11 AM",
      "source": {
        "type": "GITHUB",
        "location": "https://github.com/my-repo.git"
      },
      "created": "Sep 6, 2016 10:59:11 AM"
    }
  },
  "requestID": "9d32b228-745b-11e6-98bb-23b67EXAMPLE",
  "eventID": "581f7dd1-8d2e-40b0-aeee-0dbf7EXAMPLE",
  "eventType": "AwsApiCall",
  "recipientAccountId": "account-ID"
}
```

Memantau CodeBuild build dengan CloudWatch

Anda dapat menggunakan Amazon CloudWatch untuk menonton build Anda, melaporkan ketika ada sesuatu yang salah, dan mengambil tindakan otomatis jika diperlukan. Anda dapat memantau build Anda di dua tingkat:

Tingkat proyek

Metrik ini untuk semua build dalam proyek yang ditentukan. Untuk melihat metrik untuk proyek, `ProjectName` tentukan dimensi di CloudWatch.

AWS tingkat akun

Metrik ini untuk semua build di akun. Untuk melihat metrik di tingkat AWS akun, jangan masukkan dimensi di CloudWatch. Metrik pemanfaatan sumber daya build tidak tersedia di tingkat AWS akun.

CloudWatch metrik menunjukkan perilaku build Anda dari waktu ke waktu. Misalnya, Anda dapat memantau:

- Berapa banyak build yang dicoba dalam proyek build atau AWS akun dari waktu ke waktu.
- Berapa banyak build yang berhasil dalam proyek build atau AWS akun dari waktu ke waktu.
- Berapa banyak build yang gagal dalam proyek build atau AWS akun dari waktu ke waktu.
- Berapa banyak waktu yang CodeBuild dihabiskan untuk menjalankan build dalam proyek build atau AWS akun dari waktu ke waktu.
- Membangun pemanfaatan sumber daya untuk build atau seluruh proyek build. Membangun metrik pemanfaatan sumber daya mencakup metrik seperti CPU, memori, dan pemanfaatan penyimpanan.

Untuk informasi selengkapnya, lihat Lihat CodeBuild metrik.

CodeBuild CloudWatch metrik

Metrik berikut dapat dilacak per AWS akun atau proyek build. Untuk informasi selengkapnya tentang menggunakan CloudWatch dengan CodeBuild, lihatMemantau CodeBuild build dengan CloudWatch.

BuildDuration

Mengukur durasi `BUILD` fase build.

Unit: detik

CloudWatch Statistik yang valid: Rata-rata (disarankan), Maksimum, Minimum

Bangunan

Mengukur jumlah build yang dipicu.

Unit: Hitungan

CloudWatch Statistik yang valid: Jumlah

DownloadSourceDuration

Mengukur durasi `DOWNLOAD_SOURCE` fase build.

Unit: detik

CloudWatch Statistik yang valid: Rata-rata (disarankan), Maksimum, Minimum

Durasi

Mengukur durasi semua build dari waktu ke waktu.

Unit: detik

CloudWatch Statistik yang valid: Rata-rata (disarankan), Maksimum, Minimum

FailedBuilds

Mengukur jumlah build yang gagal karena kesalahan klien atau batas waktu.

Unit: Hitungan

CloudWatch Statistik yang valid: Jumlah

FinalizingDuration

Mengukur durasi `FINALIZING` fase build.

Unit: detik

CloudWatch Statistik yang valid: Rata-rata (disarankan), Maksimum, Minimum

InstallDuration

Mengukur durasi `INSTALL` fase build.

Unit: detik

CloudWatch Statistik yang valid: Rata-rata (disarankan), Maksimum, Minimum

PostBuildDuration

Mengukur durasi `POST_BUILD` fase build

Unit: detik

CloudWatch Statistik yang valid: Rata-rata (disarankan), Maksimum, Minimum

PreBuildDuration

Mengukur durasi `PRE_BUILD` fase build.

Unit: detik

CloudWatch Statistik yang valid: Rata-rata (disarankan), Maksimum, Minimum

ProvisioningDuration

Mengukur durasi `PROVISIONING` fase build.

Unit: detik

CloudWatch Statistik yang valid: Rata-rata (disarankan), Maksimum, Minimum

QueuedDuration

Mengukur durasi `QUEUED` fase build.

Unit: detik

CloudWatch Statistik yang valid: Rata-rata (disarankan), Maksimum, Minimum

SubmittedDuration

Mengukur durasi `SUBMITTED` fase build.

Unit: detik

CloudWatch Statistik yang valid: Rata-rata (disarankan), Maksimum, Minimum

SucceededBuilds

Mengukur jumlah build yang berhasil.

Unit: Hitungan

CloudWatch Statistik yang valid: Jumlah

UploadArtifactsDuration

Mengukur durasi `UPLOAD_ARTIFACTS` fase build.

Unit: detik

CloudWatch Statistik yang valid: Rata-rata (disarankan), Maksimum, Minimum

CodeBuild CloudWatch metrik pemanfaatan sumber daya

> **Note**
>
> CodeBuild Metrik pemanfaatan sumber daya hanya tersedia di wilayah berikut:
>
> - Wilayah Asia Pacific (Tokyo)
> - Wilayah Asia Pasifik (Seoul)
> - Wilayah Asia Pasifik (Mumbai)
> - Wilayah Asia Pasifik (Singapura)
> - Wilayah Asia Pacific (Sydney)
> - Wilayah Kanada (Pusat)
> - Wilayah Eropa (Frankfurt)
> - Wilayah Eropa (Irlandia)
> - Wilayah Eropa (London)
> - Wilayah Eropa (Paris)
> - Wilayah Amerika Selatan (Sao Paulo)
> - Wilayah AS Timur (N. Virginia)
> - Wilayah US East (Ohio)

- Wilayah US West (N California)
- Wilayah US West (Oregon)

Metrik pemanfaatan sumber daya berikut dapat dilacak. Untuk informasi selengkapnya tentang menggunakan CloudWatch dengan CodeBuild, lihatMemantau CodeBuild build dengan CloudWatch.

CPUdimanfaatkan

Jumlah unit CPU pemrosesan yang dialokasikan yang digunakan oleh wadah build.

Unit: Unit CPU

CloudWatch Statistik yang valid: Rata-rata (disarankan), Maksimum, Minimum

CPUUtilizedPercent

Persentase pemrosesan yang dialokasikan yang digunakan oleh wadah build.

Unit: Persen

CloudWatch Statistik yang valid: Rata-rata (disarankan), Maksimum, Minimum

MemoryUtilized

Jumlah megabyte memori yang digunakan oleh wadah build.

Unit: Megabyte

CloudWatch Statistik yang valid: Rata-rata (disarankan), Maksimum, Minimum

MemoryUtilizedPercent

Persentase memori yang dialokasikan yang digunakan oleh wadah build.

Unit: Persen

CloudWatch Statistik yang valid: Rata-rata (disarankan), Maksimum, Minimum

StorageReadBytes

Kecepatan baca penyimpanan yang digunakan oleh wadah build.

Unit: Bytes/second

CloudWatch Statistik yang valid: Rata-rata (disarankan), Maksimum, Minimum

StorageWriteBytes

Kecepatan tulis penyimpanan yang digunakan oleh wadah build.

Unit: Bytes/second

CloudWatch Statistik yang valid: Rata-rata (disarankan), Maksimum, Minimum

CodeBuild CloudWatch dimensi

CodeBuild menyediakan dimensi CloudWatch metrik berikut. Jika tidak ada yang ditentukan, metriknya adalah untuk AWS akun saat ini.

BuildId, BuildNumber, ProjectName

Metrik disediakan untuk pengenal build, nomor build, dan nama proyek.

ProjectName

Metrik disediakan untuk nama proyek.

CodeBuild CloudWatch alarm

Anda dapat menggunakan CloudWatch konsol untuk membuat alarm berdasarkan CodeBuild metrik sehingga Anda dapat bereaksi jika terjadi kesalahan dengan build Anda. Dua metrik yang paling berguna dengan alarm dijelaskan dalam peluru berikut. Untuk informasi selengkapnya tentang menggunakan CloudWatch dengan CodeBuild, lihatMemantau CodeBuild build dengan CloudWatch.

- `FailedBuild`. Anda dapat membuat alarm yang dipicu ketika sejumlah build gagal terdeteksi dalam jumlah detik yang telah ditentukan. Di CloudWatch, Anda menentukan jumlah detik dan berapa banyak build yang gagal memicu alarm.
- `Duration`. Anda dapat membuat alarm yang dipicu saat pembuatan membutuhkan waktu lebih lama dari yang diharapkan. Anda menentukan berapa detik yang harus berlalu setelah build dimulai dan sebelum build selesai sebelum alarm dipicu.

Untuk informasi tentang cara membuat alarm untuk CodeBuild metrik, lihatMonitor CodeBuild dibangun dengan alarm CloudWatch. Untuk informasi selengkapnya tentang alarm, lihat Membuat CloudWatch alarm Amazon di Panduan CloudWatch Pengguna Amazon.

Lihat CodeBuild metrik

AWS CodeBuild memantau fungsi atas nama Anda dan melaporkan metrik melalui Amazon CloudWatch. Metrik ini mencakup jumlah total build, build yang gagal, build yang berhasil, dan durasi build.

Anda dapat menggunakan CodeBuild konsol atau CloudWatch konsol untuk memantau metrik. CodeBuild Prosedur berikut menunjukkan cara melihat metrik.

Topik

- Lihat metrik build (CodeBuild konsol)
- Lihat metrik build (CloudWatch konsol Amazon)

Lihat metrik build (CodeBuild konsol)

> **Note**
>
> Anda tidak dapat menyesuaikan metrik atau grafik yang digunakan untuk menampilkannya di konsol. CodeBuild Jika Anda ingin menyesuaikan tampilan, gunakan CloudWatch konsol Amazon untuk melihat metrik build Anda.

Account-level metrik

Untuk melihat AWS metrik tingkat akun

1. Masuk ke Konsol Manajemen AWS dan buka AWS CodeBuild konsol di https://console.aws.amazon.com/codesuite/codebuild/home.
2. Di panel navigasi, pilih Metrik akun.

Project-level metrik

Untuk melihat metrik tingkat proyek

1. Masuk ke Konsol Manajemen AWS dan buka AWS CodeBuild konsol di https://console.aws.amazon.com/codesuite/codebuild/home.
2. Di panel navigasi, pilih Membangun proyek.

3. Dalam daftar proyek build, di kolom Nama, pilih proyek tempat Anda ingin melihat metrik.
4. Pilih tab Metrik.

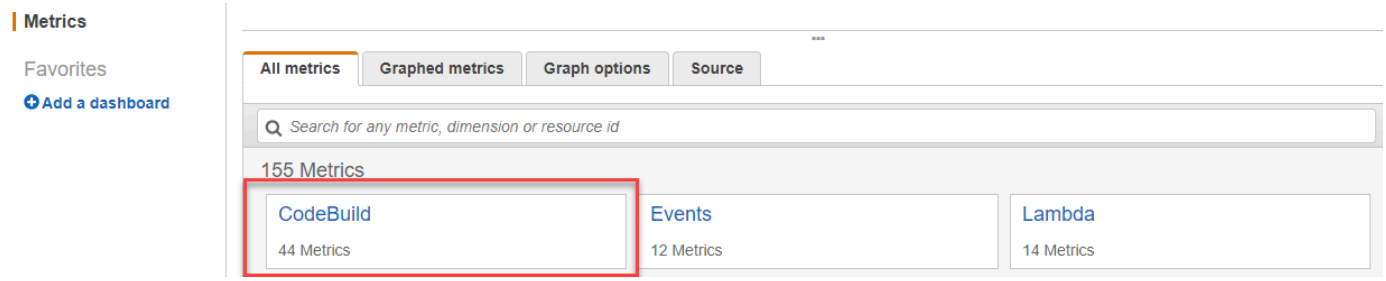
Lihat metrik build (CloudWatch konsol Amazon)

Anda dapat menyesuaikan metrik dan grafik yang digunakan untuk menampilkannya dengan konsol. CloudWatch

Account-level metrik

Untuk melihat metrik tingkat akun

1. Masuk ke Konsol Manajemen AWS dan buka CloudWatch konsol di https://console.aws.amazon.com/cloudwatch/.
2. Pada panel navigasi, silakan pilih Metrik.
3. Pada tab Semua metrik, pilih CodeBuild.

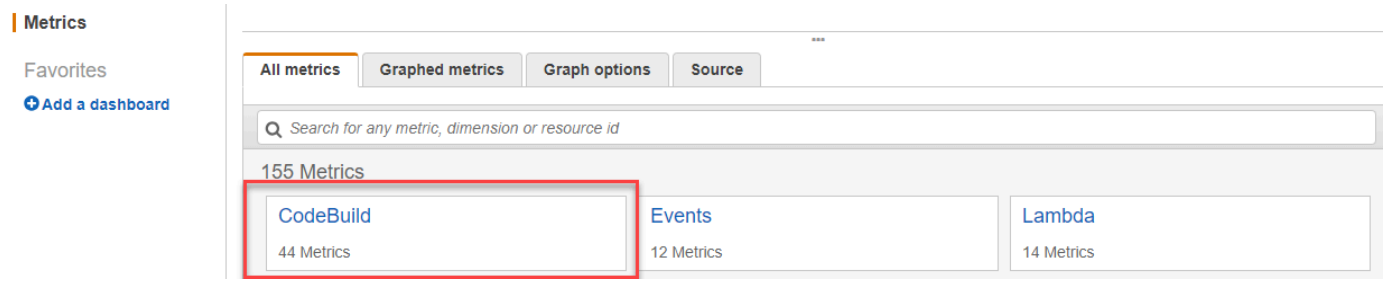


4. Silahkan Anda memilih Metrik Akun.
5. Pilih satu atau beberapa proyek dan metrik. Untuk setiap proyek, Anda dapat memilih metrik SucceededBuilds, FailedBuilds, Builds, dan Duration. Semua proyek dan kombinasi metrik yang dipilih ditampilkan dalam grafik pada halaman.

Project-level metrik

Untuk melihat metrik tingkat proyek

1. Masuk ke Konsol Manajemen AWS dan buka CloudWatch konsol di https://console.aws.amazon.com/cloudwatch/.
2. Pada panel navigasi, silakan pilih Metrik.
3. Pada tab Semua metrik, pilih CodeBuild.



4. Pilih Berdasarkan Proyek.
5. Pilih satu atau lebih proyek dan kombinasi metrik. Untuk setiap proyek, Anda dapat memilih metrik SucceededBuilds, FailedBuilds, Builds, dan Duration. Semua proyek dan kombinasi metrik yang dipilih ditampilkan dalam grafik pada halaman.
6. (Opsional) Anda dapat menyesuaikan metrik dan grafik Anda. Misalnya, dari daftar drop-down di kolom Statistik, Anda dapat memilih statistik yang berbeda untuk ditampilkan. Atau dari menu tarik-turun di kolom Periode, Anda dapat memilih periode waktu yang berbeda untuk digunakan untuk memantau metrik.

   Untuk informasi selengkapnya, lihat Metrik grafik dan Lihat metrik yang tersedia di CloudWatch Panduan Pengguna Amazon.

Lihat metrik pemanfaatan CodeBuild sumber daya

AWS CodeBuild monitor membangun pemanfaatan sumber daya atas nama Anda dan melaporkan metrik melalui Amazon. CloudWatch Ini termasuk metrik seperti CPU, memori, dan pemanfaatan penyimpanan.

> **Note**
>
> CodeBuild Metrik pemanfaatan sumber daya hanya dicatat untuk build yang berjalan selama lebih dari satu menit.

Anda dapat menggunakan CodeBuild konsol atau CloudWatch konsol untuk memantau metrik pemanfaatan sumber daya. CodeBuild

> **Note**
>
> CodeBuild Metrik pemanfaatan sumber daya hanya tersedia di wilayah berikut:

- Wilayah Asia Pacific (Tokyo)
- Wilayah Asia Pasifik (Seoul)
- Wilayah Asia Pasifik (Mumbai)
- Wilayah Asia Pasifik (Singapura)
- Wilayah Asia Pacific (Sydney)
- Wilayah Kanada (Pusat)
- Wilayah Eropa (Frankfurt)
- Wilayah Eropa (Irlandia)
- Wilayah Eropa (London)
- Wilayah Eropa (Paris)
- Wilayah Amerika Selatan (Sao Paulo)
- Wilayah AS Timur (N. Virginia)
- Wilayah US East (Ohio)
- Wilayah US West (N California)
- Wilayah US West (Oregon)

Prosedur berikut menunjukkan cara mengakses metrik pemanfaatan sumber daya Anda.

Topik

- Akses metrik pemanfaatan sumber daya (konsol) CodeBuild
- Akses metrik pemanfaatan sumber daya (konsol Amazon CloudWatch)

Akses metrik pemanfaatan sumber daya (konsol) CodeBuild

> **Note**
>
> Anda tidak dapat menyesuaikan metrik atau grafik yang digunakan untuk menampilkannya di konsol. CodeBuild Jika Anda ingin menyesuaikan tampilan, gunakan CloudWatch konsol Amazon untuk melihat metrik build Anda.

Project-level metrik pemanfaatan sumber daya

Untuk mengakses metrik pemanfaatan sumber daya tingkat proyek

1. Masuk ke Konsol Manajemen AWS dan buka AWS CodeBuild konsol di https://console.aws.amazon.com/codesuite/codebuild/home.
2. Di panel navigasi, pilih Membangun proyek.
3. Dalam daftar proyek pembangunan, di kolom Nama, pilih proyek yang ingin Anda lihat metrik pemanfaatannya.
4. Pilih tab Metrik. Metrik pemanfaatan sumber daya ditampilkan di bagian Metrik pemanfaatan sumber daya.
5. Untuk melihat metrik pemanfaatan sumber daya tingkat proyek di CloudWatch konsol, pilih Lihat CloudWatch di bagian Metrik pemanfaatan sumber daya.

Build-level metrik pemanfaatan sumber daya

Untuk mengakses metrik pemanfaatan sumber daya tingkat build

1. Masuk ke Konsol Manajemen AWS dan buka AWS CodeBuild konsol di https://console.aws.amazon.com/codesuite/codebuild/home.
2. Di panel navigasi, pilih Build history.
3. Dalam daftar build, di kolom Build run, pilih build yang ingin Anda lihat metrik pemanfaatannya.
4. Pilih tab Pemanfaatan sumber daya.
5. Untuk melihat metrik pemanfaatan sumber daya tingkat build di CloudWatch konsol, pilih Lihat CloudWatch di bagian Metrik pemanfaatan sumber daya.

Akses metrik pemanfaatan sumber daya (konsol Amazon CloudWatch)

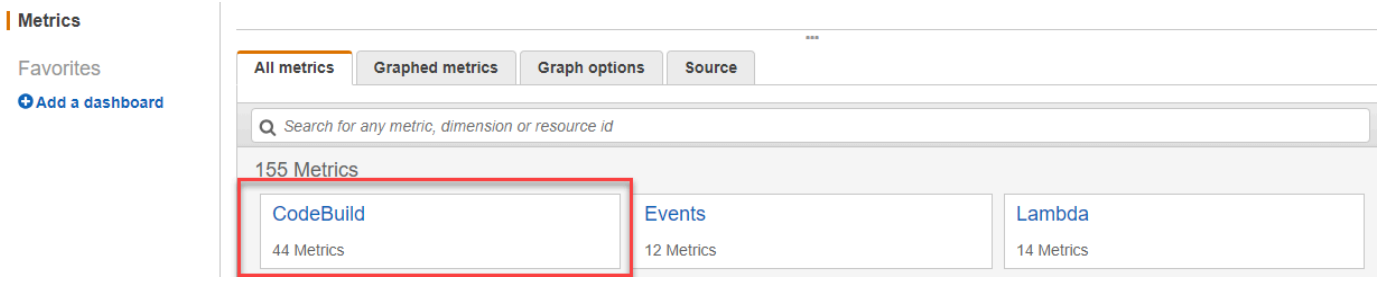
CloudWatch Konsol Amazon dapat digunakan untuk mengakses metrik pemanfaatan CodeBuild sumber daya.

Project-level metrik pemanfaatan sumber daya

Untuk mengakses metrik pemanfaatan sumber daya tingkat proyek

1. Masuk ke Konsol Manajemen AWS dan buka CloudWatch konsol di https://console.aws.amazon.com/cloudwatch/.

2. Pada panel navigasi, silakan pilih Metrik.
3. Pada tab Semua metrik, pilih CodeBuild.



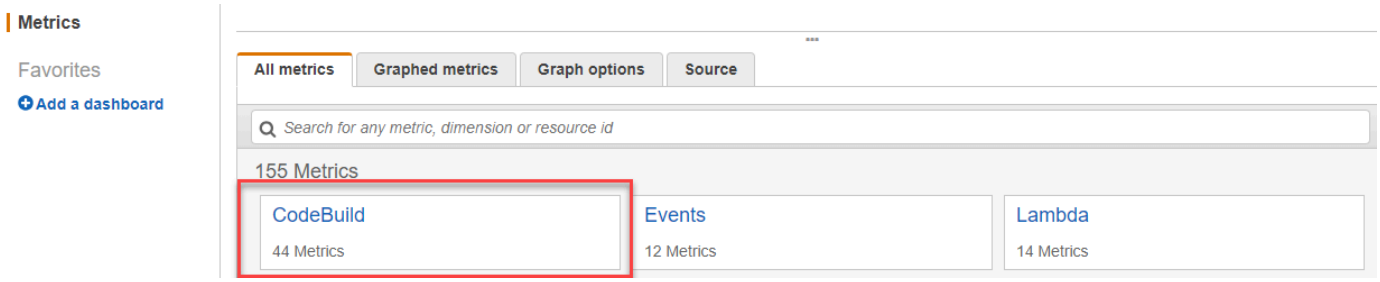
4. Pilih Berdasarkan Proyek.
5. Pilih satu atau beberapa proyek dan kombinasi metrik untuk ditambahkan ke grafik. Semua proyek dan kombinasi metrik yang dipilih ditampilkan dalam grafik pada halaman.
6. (Opsional) Anda dapat menyesuaikan metrik dan grafik Anda dari tab Metrik Grafik. Misalnya, dari daftar drop-down di kolom Statistik, Anda dapat memilih statistik yang berbeda untuk ditampilkan. Atau dari menu tarik-turun di kolom Periode, Anda dapat memilih periode waktu yang berbeda untuk digunakan untuk memantau metrik.

   Untuk informasi selengkapnya, lihat Metrik grafik dan Melihat metrik yang tersedia di Panduan Pengguna Amazon CloudWatch .

Build-level metrik pemanfaatan sumber daya

Untuk mengakses metrik pemanfaatan sumber daya tingkat build

1. Masuk ke Konsol Manajemen AWS dan buka CloudWatch konsol di https://console.aws.amazon.com/cloudwatch/.
2. Pada panel navigasi, silakan pilih Metrik.
3. Pada tab Semua metrik, pilih CodeBuild.



4. Pilih BuildId, BuildNumber, ProjectName.

5. Pilih satu atau beberapa kombinasi build dan metrik untuk ditambahkan ke grafik. Semua kombinasi build dan metrik yang dipilih ditampilkan dalam grafik pada halaman.
6. (Opsional) Anda dapat menyesuaikan metrik dan grafik Anda dari tab Metrik Grafik. Misalnya, dari daftar drop-down di kolom Statistik, Anda dapat memilih statistik yang berbeda untuk ditampilkan. Atau dari menu tarik-turun di kolom Periode, Anda dapat memilih periode waktu yang berbeda untuk digunakan untuk memantau metrik.

   Untuk informasi selengkapnya, lihat Metrik grafik dan Melihat metrik yang tersedia di Panduan Pengguna Amazon CloudWatch .

Monitor CodeBuild dibangun dengan alarm CloudWatch

Anda dapat membuat CloudWatch alarm untuk build Anda. Alarm mengawasi satu metrik selama periode waktu yang Anda tentukan dan melakukan satu atau beberapa tindakan berdasarkan nilai metrik relatif terhadap ambang batas tertentu selama beberapa periode waktu. Menggunakan fungsionalitas CloudWatch alarm asli, Anda dapat menentukan tindakan apa pun yang didukung CloudWatch ketika ambang batas terlampaui. Misalnya, Anda dapat menentukan bahwa notifikasi Amazon SNS dikirim ketika lebih dari tiga build di akun Anda gagal dalam lima belas menit.

Untuk membuat CloudWatch alarm untuk CodeBuild metrik

1. Masuk ke Konsol Manajemen AWS dan buka CloudWatch konsol di https://console.aws.amazon.com/cloudwatch/.
2. Di panel navigasi, pilih Alarm.
3. Pilih Buat Alarm.
4. Di bawah CloudWatch Metrik berdasarkan Kategori, pilih CodeBuildMetrik. Jika Anda tahu bahwa Anda hanya menginginkan metrik tingkat proyek, pilih Berdasarkan Proyek. Jika Anda tahu bahwa Anda hanya menginginkan metrik tingkat akun, pilih Metrik Akun.
5. Pada Buat Alarm, jika belum dipilih, pilih Pilih Metrik.
6. Pilih metrik yang ingin Anda buat alarm. Pilihannya adalah Berdasarkan Proyek atau Metrik Akun.
7. Pilih Berikutnya atau Tentukan Alarm dan kemudian buat alarm Anda. Untuk informasi selengkapnya, lihat Membuat CloudWatch alarm Amazon di Panduan CloudWatch Pengguna Amazon. Untuk informasi selengkapnya tentang mengatur notifikasi Amazon SNS saat alarm dipicu, lihat Mengatur notifikasi Amazon SNS di Panduan Pengembang Amazon SNS.
8. Pilih Buat Alarm.

Keamanan di AWS CodeBuild

Keamanan cloud di AWS adalah prioritas tertinggi. Sebagai AWS pelanggan, Anda mendapat manfaat dari pusat data dan arsitektur jaringan yang dibangun untuk memenuhi persyaratan organisasi yang paling sensitif terhadap keamanan.

Keamanan adalah tanggung jawab bersama antara Anda AWS dan Anda. Model tanggung jawab bersama menggambarkan hal ini sebagai keamanan dari cloud dan keamanan di cloud:

- Keamanan cloud — AWS bertanggung jawab untuk melindungi infrastruktur yang menjalankan AWS layanan di AWS Cloud. AWS juga memberi Anda layanan yang dapat Anda gunakan dengan aman. Third-party Auditor secara teratur menguji dan memverifikasi efektivitas keamanan kami sebagai bagian dari Program AWS Kepatuhan. Untuk mempelajari tentang program kepatuhan yang berlaku AWS CodeBuild, lihat AWS Layanan dalam Lingkup berdasarkan Program Kepatuhan.
- Keamanan di cloud — Tanggung jawab Anda ditentukan oleh AWS layanan yang Anda gunakan. Anda juga bertanggung jawab atas faktor lain, termasuk sensitivitas data Anda, persyaratan perusahaan Anda, serta undang-undang dan peraturan yang berlaku.

Dokumentasi ini membantu Anda memahami cara menerapkan model tanggung jawab bersama saat menggunakan CodeBuild. Topik berikut menunjukkan cara mengonfigurasi CodeBuild untuk memenuhi tujuan keamanan dan kepatuhan Anda. Anda juga belajar cara menggunakan AWS layanan lain yang membantu Anda memantau dan mengamankan CodeBuild sumber daya Anda.

Topik

- Perlindungan data di AWS CodeBuild
- Identitas dan manajemen akses di AWS CodeBuild
- Validasi kepatuhan untuk AWS CodeBuild
- Ketahanan di AWS CodeBuild
- Keamanan Infrastruktur di AWS CodeBuild
- Akses penyedia sumber Anda di CodeBuild
- Cross-service pencegahan wakil bingung

Perlindungan data di AWS CodeBuild

Model tanggung jawab bersama AWS berlaku untuk perlindungan data di AWS CodeBuild. Seperti yang dijelaskan dalam model AWS ini, bertanggung jawab untuk melindungi infrastruktur global yang menjalankan semua AWS Cloud. Anda bertanggung jawab untuk mempertahankan kendali atas konten yang di-host pada infrastruktur ini. Anda juga bertanggung jawab atas tugas-tugas konfigurasi dan manajemen keamanan untuk Layanan AWS yang Anda gunakan. Untuk informasi selengkapnya tentang privasi data, lihat FAQ Privasi Data AWS . Untuk informasi tentang perlindungan data di Eropa, lihat Pusat Peraturan Umum Perlindungan Data (GDPR).

Untuk tujuan perlindungan data, kami menyarankan Anda melindungi Akun AWS kredensil dan mengatur pengguna individu dengan AWS IAM Identity Center atau AWS Identity and Access Management (IAM). Dengan cara itu, setiap pengguna hanya diberi izin yang diperlukan untuk memenuhi tanggung jawab tugasnya. Kami juga menyarankan supaya Anda mengamankan data dengan cara-cara berikut:

- Gunakan autentikasi multi-faktor (MFA) pada setiap akun.
- Gunakan SSL/TLS untuk berkomunikasi dengan AWS sumber daya. Kami mensyaratkan TLS 1.2 dan menganjurkan TLS 1.3.
- Siapkan API dan logging aktivitas pengguna dengan AWS CloudTrail. Untuk informasi tentang penggunaan CloudTrail jejak untuk menangkap AWS aktivitas, lihat Bekerja dengan CloudTrail jejak di AWS CloudTrail Panduan Pengguna.
- Gunakan solusi AWS enkripsi, bersama dengan semua kontrol keamanan default di dalamnya Layanan AWS.
- Gunakan layanan keamanan terkelola tingkat lanjut seperti Amazon Macie, yang membantu menemukan dan mengamankan data sensitif yang disimpan di Amazon S3.
- Jika Anda memerlukan modul kriptografi tervalidasi FIPS 140-3 saat mengakses AWS melalui antarmuka baris perintah atau API, gunakan titik akhir FIPS. Lihat informasi selengkapnya tentang titik akhir FIPS yang tersedia di Standar Pemrosesan Informasi Federal (FIPS) 140-3.

Kami sangat merekomendasikan agar Anda tidak pernah memasukkan informasi identifikasi yang sensitif, seperti nomor rekening pelanggan Anda, ke dalam tanda atau bidang isian bebas seperti bidang Nama. Ini termasuk saat Anda bekerja dengan CodeBuild atau lainnya Layanan AWS menggunakan konsol, API AWS CLI, atau AWS SDK. Data apa pun yang Anda masukkan ke dalam tanda atau bidang isian bebas yang digunakan untuk nama dapat digunakan untuk log penagihan atau log diagnostik. Saat Anda memberikan URL ke server eksternal, kami sangat menganjurkan

supaya Anda tidak menyertakan informasi kredensial di dalam URL untuk memvalidasi permintaan Anda ke server itu.

Untuk melindungi informasi sensitif, berikut ini disembunyikan di CodeBuild log:

- String ditentukan menggunakan Parameter Store dalam variabel lingkungan CodeBuild proyek atau bagian `env/parameter-store` buildspec. Untuk informasi selengkapnya, lihat panduan konsol Systems Manager Parameter Store dan Systems Manager Parameter Store di Panduan Pengguna Amazon EC2 Systems Manager.
- String ditentukan menggunakan variabel AWS Secrets Manager lingkungan CodeBuild proyek atau bagian `env/secrets-manager` buildspec. Untuk informasi selengkapnya, lihat Manajemen kunci.

> **Note**
>
> Masking cocok dengan nilai persis yang disimpan di Amazon EC2 Systems Manager Parameter Store atau. AWS Secrets Manager Jika perintah build mengubah rahasia sebelum muncul di log, hasilnya adalah string berbeda yang tidak disamarkan. Misalnya, menyandikannya dengan Base64, membalikkannya, mengubah kasus, atau menggabungkannya dengan teks lain menciptakan nilai berbeda yang dapat muncul di log build.
>
> Untuk menjaga nilai tetap tertutup, simpan dalam bentuk yang digunakan build Anda. Misalnya, jika build Anda memerlukan Base64-encoded nilai, simpan Base64-encoded string secara langsung di Amazon EC2 Systems Manager Parameter Store AWS Secrets Manager atau Amazon EC2 Systems Manager. Dengan cara ini, nilai yang dicatat cocok dengan nilai yang disimpan.

Untuk informasi selengkapnya tentang perlindungan data, lihat postingan blog model tanggung tawab bersama AWS dan Peraturan Perlindungan Data Umum (GDPR) di Blog Keamanan AWS .

Topik

- Enkripsi data
- Manajemen kunci
- Privasi lalu lintas

Enkripsi data

Enkripsi adalah bagian penting dari CodeBuild keamanan. Beberapa enkripsi, seperti untuk data dalam perjalanan, disediakan secara default dan tidak mengharuskan Anda untuk melakukan apa pun. Enkripsi lain, seperti untuk data saat istirahat, Anda dapat mengonfigurasi saat membuat proyek atau membangun.

- Enkripsi data saat istirahat - Artefak build, seperti cache, log, file data laporan pengujian mentah yang diekspor, dan hasil build, dienkripsi secara default menggunakan. Kunci yang dikelola AWS Jika Anda tidak ingin menggunakan kunci KMS ini, Anda harus membuat dan mengkonfigurasi kunci yang dikelola pelanggan. Untuk informasi lebih lanjut Membuat Kunci KMS dan konsep Layanan Manajemen AWS Kunci di Panduan AWS Key Management Service Pengguna.
  - Anda dapat menyimpan pengenal kunci AWS KMS yang CodeBuild digunakan untuk mengenkripsi artefak keluaran build dalam variabel lingkungan. `CODEBUILD_KMS_KEY_ID` Untuk informasi selengkapnya, lihat Variabel lingkungan di lingkungan build
  - Anda dapat menentukan kunci terkelola pelanggan saat membuat proyek build. Untuk informasi selengkapnya, lihat Set the Encryption Key Using the Console dan Mengatur kunci enkripsi menggunakan CLI.

  Volume Amazon Elastic Block Store dari armada build Anda dienkripsi secara default menggunakan. Kunci yang dikelola AWS
- Enkripsi data dalam transit - Semua komunikasi antara pelanggan dan dan antara CodeBuild CodeBuild dan dependensi hilirnya dilindungi menggunakan koneksi TLS yang ditandatangani menggunakan proses penandatanganan Signature Version 4. Semua CodeBuild titik akhir menggunakan SHA-256 sertifikat yang dikelola oleh AWS Private Certificate Authority. Untuk informasi selengkapnya, lihat Proses penandatanganan Signature Version 4 dan Tentang ACM PCA.
- Enkripsi artefak build - Peran CodeBuild layanan yang terkait dengan proyek build memerlukan akses ke kunci KMS untuk mengenkripsi artefak keluaran build-nya. Secara default, CodeBuild gunakan Kunci yang dikelola AWS untuk Amazon S3 di akun Anda AWS . Jika Anda tidak ingin menggunakan ini Kunci yang dikelola AWS, Anda harus membuat dan mengkonfigurasi kunci yang dikelola pelanggan. Untuk informasi selengkapnya, lihat Enkripsi keluaran build dan Membuat kunci di Panduan AWS KMS Pengembang.

Manajemen kunci

Anda dapat melindungi konten Anda dari penggunaan yang tidak sah melalui enkripsi. Simpan kunci enkripsi Anda AWS Secrets Manager, lalu berikan izin peran CodeBuild layanan yang terkait dengan proyek build untuk mendapatkan kunci enkripsi dari akun Secrets Manager Anda. Untuk informasi lebih lanjut, lihatEnkripsi output build menggunakan kunci yang dikelola pelanggan,, Membuat proyek pembangunan di AWS CodeBuildJalankan AWS CodeBuild build secara manual, dan Tutorial: Menyimpan dan mengambil rahasia.

Gunakan variabel `CODEBUILD_KMS_KEY_ID` lingkungan dalam perintah build untuk mendapatkan pengenal AWS KMS kunci. Untuk informasi selengkapnya, lihat Variabel lingkungan di lingkungan build.

Anda dapat menggunakan Secrets Manager untuk melindungi kredensil ke registri pribadi yang menyimpan image Docker yang digunakan untuk lingkungan runtime Anda. Untuk informasi selengkapnya, lihat Registri pribadi dengan AWS Secrets Manager sampel untuk CodeBuild.

Privasi lalu lintas

Anda dapat meningkatkan keamanan build Anda dengan mengonfigurasi CodeBuild untuk menggunakan titik akhir VPC antarmuka. Untuk melakukan ini, Anda tidak memerlukan gateway internet, perangkat NAT, atau gateway pribadi virtual. Hal ini juga tidak diperlukan untuk mengkonfigurasi PrivateLink, meskipun dianjurkan. Untuk informasi selengkapnya, lihat Gunakan titik akhir VPC. Untuk informasi selengkapnya tentang PrivateLink dan titik akhir VPC, lihat AWS PrivateLinkdan Mengakses AWS layanan melalui. PrivateLink

Identitas dan manajemen akses di AWS CodeBuild

Akses ke AWS CodeBuild membutuhkan kredensi. Kredensyal tersebut harus memiliki izin untuk mengakses AWS sumber daya, seperti menyimpan dan mengambil artefak build di bucket S3 dan melihat Log Amazon untuk build. CloudWatch Bagian berikut menjelaskan bagaimana Anda dapat menggunakan AWS Identity and Access Management(IAM) dan CodeBuild untuk membantu mengamankan akses ke sumber daya Anda:

Ikhtisar mengelola izin akses ke AWS CodeBuild sumber daya

Setiap AWS sumber daya dimiliki oleh AWS akun, dan izin untuk membuat atau mengakses sumber daya diatur oleh kebijakan izin. Administrator akun dapat melampirkan kebijakan izin ke identitas IAM (yaitu, pengguna, grup, dan peran).

> **Note**
>
> Administrator akun (atau pengguna administrator) adalah pengguna dengan hak akses administrator. Untuk informasi selengkapnya, lihat Praktik Terbaik IAM dalam Panduan Pengguna IAM.

Ketika memberikan izin, Anda memutuskan orang yang mendapatkan izin, sumber daya yang dapat diakses, dan tindakan yang dapat dilakukan di sumber daya tersebut.

Topik

- AWS CodeBuild sumber daya dan operasi
- Memahami kepemilikan sumber daya
- Mengelola akses ke sumber daya
- Menentukan elemen kebijakan: Tindakan, efek, dan penanggung jawab

AWS CodeBuild sumber daya dan operasi

Di AWS CodeBuild, sumber daya utama adalah proyek pembangunan. Dalam sebuah kebijakan, Anda menggunakan Amazon Resource Name (ARN) untuk mengidentifikasi sumber daya tempat kebijakan berlaku. Build juga merupakan sumber daya dan memiliki ARN yang terkait dengannya. Untuk informasi selengkapnya, lihat Nama Sumber Daya Amazon (ARN) dan Ruang Nama AWS Layanan di bagian. Referensi Umum Amazon Web Services

Jenis Sumber Daya	Format ARN
Bangun proyek	`arn:aws:codebuild:` *region-ID* `:`*account-ID* `:project/` *project-name*
Build	`arn:aws:codebuild:` *region-ID* `:`*account-ID* `:build/`*build-ID*
Grup laporan	`arn:aws:codebuild:` *region-ID* `:`*account-ID* `:report-group/` *report-group-name*
Laporan	`arn:aws:codebuild:` *region-ID* `:`*account-ID* `:report/`*report-ID*

Jenis Sumber Daya	Format ARN
Armada	`arn:aws:codebuild:` *`region-ID`* `:` *`account-ID`* `:fleet/` *`fleet-ID`*
Semua CodeBuild sumber daya	`arn:aws:codebuild:*`
Semua CodeBuild sumber daya yang dimiliki oleh akun yang ditentukan di AWS Wilayah yang ditentukan	`arn:aws:codebuild:` *`region-ID`* `:` *`account-ID`* `:*`

Important

Saat menggunakan fitur kapasitas cadangan, data yang di-cache pada instance armada, termasuk file sumber, lapisan Docker, dan direktori cache yang ditentukan dalam buildspec, dapat diakses oleh proyek lain dalam akun yang sama. Ini dirancang dan memungkinkan proyek dalam akun yang sama untuk berbagi instance armada.

Note

Sebagian besar AWS layanan memperlakukan titik dua (:) atau garis miring (/) sebagai karakter yang sama di ARN. Namun, CodeBuild menggunakan kecocokan yang tepat dalam pola dan aturan sumber daya. Pastikan untuk menggunakan karakter yang benar saat membuat pola peristiwa sehingga cocok dengan sintaks ARN di sumber daya.

Misalnya, Anda dapat menunjukkan proyek build (*`myBuildProject`*) tertentu dalam pernyataan Anda menggunakan ARN sebagai berikut:

```
"Resource": "arn:aws:codebuild:us-east-2:123456789012:project/myBuildProject"
```

Untuk menentukan semua sumber daya, atau jika tindakan API tidak mendukung ARN, gunakan karakter wildcard (*) dalam `Resource` elemen sebagai berikut:

```
"Resource": "*"
```

Beberapa tindakan CodeBuild API menerima beberapa sumber daya (misalnya,`BatchGetProjects`). Untuk menentukan beberapa sumber daya dalam satu pernyataan, pisahkan ARN mereka dengan koma, sebagai berikut:

```
"Resource": [
  "arn:aws:codebuild:us-east-2:123456789012:project/myBuildProject",
  "arn:aws:codebuild:us-east-2:123456789012:project/myOtherBuildProject"
]
```

CodeBuild menyediakan satu set operasi untuk bekerja dengan CodeBuild sumber daya. Untuk daftar, lihat AWS CodeBuild referensi izin.

Memahami kepemilikan sumber daya

AWS Akun memiliki sumber daya yang dibuat di akun, terlepas dari siapa yang membuat sumber daya. Secara khusus, pemilik sumber daya adalah AWS akun entitas utama (yaitu, akun root, pengguna, atau peran IAM) yang mengautentikasi permintaan pembuatan sumber daya. Contoh berikut menggambarkan cara kerjanya:

- Jika Anda menggunakan kredensi akun root AWS akun Anda untuk membuat aturan, AWS akun Anda adalah pemilik sumber daya. CodeBuild
- Jika Anda membuat pengguna di AWS akun Anda dan memberikan izin untuk membuat CodeBuild sumber daya kepada pengguna tersebut, pengguna dapat membuat CodeBuild sumber daya. Namun, AWS akun Anda, tempat pengguna berada, memiliki CodeBuild sumber daya.
- Jika Anda membuat peran IAM di AWS akun Anda dengan izin untuk membuat CodeBuild sumber daya, siapa pun yang dapat mengambil peran tersebut dapat membuat CodeBuild sumber daya. AWS Akun Anda, tempat peran itu berada, memiliki CodeBuild sumber daya.

Mengelola akses ke sumber daya

Kebijakan izin menjelaskan siapa yang memiliki akses ke sumber daya mana.

> **Note**
>
> Bagian ini membahas penggunaan IAM di AWS CodeBuild. Bagian ini tidak memberikan informasi yang mendetail tentang layanan IAM. Untuk dokumentasi IAM lengkap, lihat Apa

yang Dimaksud dengan IAM? dalam Panduan Pengguna IAM. Untuk informasi tentang sintaksis dan deskripsi kebijakan IAM, lihat Referensi Kebijakan IAM AWS dalam Panduan Pengguna IAM.

Kebijakan yang terlampir pada identitas IAM disebut sebagai kebijakan berbasis identitas (kebijakan IAM). Kebijakan yang melekat pada sumber daya disebut sebagai kebijakan berbasis sumber daya. CodeBuild mendukung kebijakan berbasis identitas, dan kebijakan berbasis sumber daya untuk API baca saja tertentu untuk tujuan berbagi sumber daya lintas akun.

Akses aman ke bucket S3

Kami sangat menyarankan agar Anda menyertakan izin berikut dalam peran IAM Anda untuk memverifikasi bucket S3 yang terkait dengan CodeBuild proyek Anda dimiliki oleh Anda atau seseorang yang Anda percayai. Izin ini tidak disertakan dalam kebijakan dan peran AWS terkelola. Anda harus menambahkannya sendiri.

- `s3:GetBucketAcl`
- `s3:GetBucketLocation`

Jika pemilik bucket S3 yang digunakan oleh project Anda berubah, Anda harus memverifikasi bahwa Anda masih memiliki bucket dan memperbarui izin dalam peran IAM Anda jika tidak. Untuk informasi selengkapnya, lihat Izinkan pengguna berinteraksi dengan CodeBuild dan Memungkinkan CodeBuild untuk berinteraksi dengan orang lain AWS layanan.

Menentukan elemen kebijakan: Tindakan, efek, dan penanggung jawab

Untuk setiap AWS CodeBuild sumber daya, layanan mendefinisikan satu set operasi API. Untuk memberikan izin untuk operasi API ini, CodeBuild tentukan serangkaian tindakan yang dapat Anda tentukan dalam kebijakan. Beberapa operasi API dapat memerlukan izin untuk lebih dari satu tindakan untuk melakukan operasi API. Untuk informasi selengkapnya, lihat AWS CodeBuild sumber daya dan operasi dan AWS CodeBuild referensi izin.

Berikut ini adalah elemen-elemen kebijakan dasar:

- Sumber daya – Anda menggunakan Amazon Resource Name (ARN) untuk mengidentifikasi sumber daya yang diberlakukan oleh kebijakan tersebut.

- Tindakan – Anda menggunakan kata kunci tindakan untuk mengidentifikasi operasi sumber daya yang ingin Anda izinkan atau tolak. Misalnya, izin `codebuild:CreateProject` memungkinkan pengguna melakukan operasi `CreateProject`.
- Pengaruh – Anda menetapkan pengaruh, baik mengizinkan atau menolak, ketika pengguna meminta tindakan tertentu. Jika Anda tidak secara eksplisit memberikan akses ke (mengizinkan) sumber daya, akses akan ditolak secara implisit. Anda juga dapat secara eksplisit menolak akses ke sumber daya. Anda dapat melakukan ini untuk memastikan pengguna tidak dapat mengakses sumber daya, meskipun kebijakan lain memberikan akses.
- Prinsipal – Dalam kebijakan berbasis identitas (kebijakan IAM), pengguna yang kebijakannya terlampir adalah prinsipal implisit. Untuk kebijakan berbasis sumber daya, Anda menentukan pengguna, akun, layanan, atau entitas lain yang ingin Anda terima izinnya.

Untuk mempelajari selengkapnya tentang sintaksis dan deskripsi kebijakan IAM, lihat Referensi Kebijakan IAM AWS dalam Panduan Pengguna IAM.

Untuk tabel yang menunjukkan semua tindakan CodeBuild API dan sumber daya yang diterapkan, lihatAWS CodeBuild referensi izin.

Menggunakan kebijakan berbasis identitas untuk AWS CodeBuild

Topik ini memberikan contoh kebijakan berbasis identitas yang menunjukkan bagaimana administrator akun dapat melampirkan kebijakan izin ke identitas IAM (yaitu, pengguna, grup, dan peran) dan dengan demikian memberikan izin untuk melakukan operasi AWS CodeBuild sumber daya.

> **Important**
>
> Kami menyarankan Anda terlebih dahulu meninjau topik pengantar yang menjelaskan konsep dasar dan opsi yang tersedia untuk mengelola akses ke CodeBuild sumber daya Anda. Untuk informasi selengkapnya, lihat Ikhtisar mengelola izin akses ke AWS CodeBuild sumber daya.

Topik

- Izin yang diperlukan untuk menggunakan AWS CodeBuild konsol
- Izin diperlukan untuk AWS CodeBuild untuk terhubung ke Amazon Elastic Container Registry
- Izin yang diperlukan untuk AWS CodeBuild konsol untuk terhubung ke penyedia sumber

- AWS kebijakan terkelola (standar) untuk AWS CodeBuild
- CodeBuild kebijakan dan pemberitahuan terkelola
- CodeBuild update ke AWS kebijakan terkelola
- Customer-managed contoh kebijakan

Berikut ini menunjukkan contoh kebijakan izin yang memungkinkan pengguna mendapatkan informasi tentang proyek build hanya di `us-east-2` wilayah `123456789012` untuk akun proyek build apa pun yang dimulai dengan nama: `my`

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:BatchGetProjects",
      "Resource": "arn:aws:codebuild:us-east-2:111122223333:project/my*"
    }
  ]
}
```

Izin yang diperlukan untuk menggunakan AWS CodeBuild konsol

Pengguna yang menggunakan AWS CodeBuild konsol harus memiliki seperangkat izin minimum yang memungkinkan pengguna menjelaskan AWS sumber daya lain untuk AWS akun tersebut. Anda harus memiliki izin dari layanan berikut:

- AWS CodeBuild
- Amazon CloudWatch
- CodeCommit (jika Anda menyimpan kode sumber Anda di AWS CodeCommit repositori)
- Amazon Elastic Container Registry (Amazon ECR) (jika Anda menggunakan lingkungan build yang mengandalkan image Docker di repositori Amazon ECR)

> **Note**
>
> Per 26 Juli 2022, kebijakan IAM default telah diperbarui. Untuk informasi selengkapnya, lihat Izin diperlukan untuk AWS CodeBuild untuk terhubung ke Amazon Elastic Container Registry.

- Amazon Elastic Container Service (Amazon ECS) (jika Anda menggunakan lingkungan build yang mengandalkan image Docker di repositori Amazon ECR)
- AWS Identity and Access Management (IAM)
- AWS Key Management Service (AWS KMS)
- Amazon Simple Storage Service (Amazon S3)

Jika Anda membuat kebijakan IAM yang lebih ketat dari izin minimum yang diperlukan, konsol tidak akan berfungsi sebagaimana dimaksudkan.

Izin diperlukan untuk AWS CodeBuild untuk terhubung ke Amazon Elastic Container Registry

Per 26 Juli 2022, AWS CodeBuild telah memperbarui kebijakan IAM default untuk izin Amazon ECR. Izin berikut telah dihapus dari kebijakan default:

```
"ecr:PutImage",
"ecr:InitiateLayerUpload",
"ecr:UploadLayerPart",
"ecr:CompleteLayerUpload"
```

Untuk CodeBuild proyek yang dibuat sebelum 26 Juli 2022, kami sarankan Anda memperbarui kebijakan Anda dengan kebijakan ECR Amazon berikut:

```
"Action": [
    "ecr:BatchCheckLayerAvailability",
    "ecr:GetDownloadUrlForLayer",
    "ecr:BatchGetImage"
]
```

Untuk informasi selengkapnya tentang memperbarui kebijakan Anda, lihatIzinkan pengguna berinteraksi dengan CodeBuild.

Izin yang diperlukan untuk AWS CodeBuild konsol untuk terhubung ke penyedia sumber

AWS CodeBuild Konsol menggunakan tindakan API berikut untuk terhubung ke penyedia sumber (misalnya, GitHub repositori).

- `codebuild:ListConnectedOAuthAccounts`
- `codebuild:ListRepositories`
- `codebuild:PersistOAuthToken`
- `codebuild:ImportSourceCredentials`

Anda dapat mengaitkan penyedia sumber (seperti GitHub repositori) dengan proyek build menggunakan konsol. AWS CodeBuild Untuk melakukannya, Anda harus terlebih dahulu menambahkan tindakan API sebelumnya ke kebijakan akses IAM yang terkait dengan pengguna yang Anda gunakan untuk mengakses konsol. AWS CodeBuild

Tindakan `ListConnectedOAuthAccountsListRepositories`,, dan `PersistOAuthToken` API tidak dimaksudkan untuk dipanggil oleh kode Anda. Oleh karena itu, tindakan API ini tidak termasuk dalam AWS CLI dan AWS SDK.

AWS kebijakan terkelola (standar) untuk AWS CodeBuild

AWS mengatasi banyak kasus penggunaan umum dengan menyediakan kebijakan IAM mandiri yang dibuat dan dikelola oleh. AWS Kebijakan AWS terkelola ini memberikan izin yang diperlukan untuk kasus penggunaan umum sehingga Anda dapat menghindari keharusan menyelidiki izin apa yang diperlukan. Kebijakan terkelola CodeBuild juga memberikan izin untuk melakukan operasi di layanan lain, seperti IAM, Amazon EC2 AWS CodeCommit, Amazon ECR, Amazon SNS, CloudWatch dan Amazon Events, sebagaimana diperlukan untuk tanggung jawab bagi pengguna yang telah diberikan kebijakan tersebut. Misalnya, `AWSCodeBuildAdminAccess` kebijakan tersebut adalah kebijakan pengguna tingkat administratif yang memungkinkan pengguna dengan kebijakan ini membuat dan mengelola aturan CloudWatch Acara untuk pembuatan proyek dan topik Amazon SNS untuk pemberitahuan tentang peristiwa terkait proyek (topik yang namanya diawali`arn:aws:codebuild:`), serta mengelola proyek dan grup laporan di. CodeBuild Untuk informasi selengkapnya, lihat Kebijakan Terkelola AWS dalam Panduan Pengguna IAM.

Kebijakan AWS terkelola berikut, yang dapat Anda lampirkan ke pengguna di akun Anda, khusus untuk AWS CodeBuild.

AWSCodeBuildAdminAccess

Menyediakan akses penuh untuk CodeBuild menyertakan izin untuk mengelola proyek CodeBuild pembangunan.

AWSCodeBuildDeveloperAccess

Menyediakan akses ke CodeBuild tetapi tidak mengizinkan administrasi proyek build.

AWSCodeBuildReadOnlyAccess

Menyediakan akses hanya-baca ke. CodeBuild

Untuk mengakses artefak keluaran build yang CodeBuild dibuat, Anda juga harus melampirkan kebijakan AWS terkelola bernama`AmazonS3ReadOnlyAccess`.

Untuk membuat dan mengelola peran CodeBuild layanan, Anda juga harus melampirkan kebijakan AWS terkelola bernama`IAMFullAccess`.

Anda juga dapat membuat kebijakan IAM khusus Anda sendiri untuk memberikan izin untuk CodeBuild tindakan dan sumber daya. Anda dapat menyematkan kebijakan khusus ini untuk pengguna atau grup yang memerlukan izin tersebut.

Topik

- AWSCodeBuildAdminAccess
- AWSCodeBuildDeveloperAccess
- AWSCodeBuildReadOnlyAccess

AWSCodeBuildAdminAccess

`AWSCodeBuildAdminAccess`Kebijakan ini menyediakan akses penuh ke CodeBuild, termasuk izin untuk mengelola proyek CodeBuild build. Terapkan kebijakan ini hanya kepada pengguna tingkat administratif untuk memberi mereka kontrol penuh atas CodeBuild proyek, grup laporan, dan sumber daya terkait di AWS akun Anda, termasuk kemampuan untuk menghapus proyek dan grup laporan.

Untuk kebijakan terkelola lengkap, lihat AWSCodeBuildAdminAccessdi referensi kebijakan terkelola IAM.

AWSCodeBuildDeveloperAccess

`AWSCodeBuildDeveloperAccess`Kebijakan ini memungkinkan akses ke semua fungsionalitas CodeBuild dan memproyeksikan dan melaporkan sumber daya terkait grup. Kebijakan ini tidak mengizinkan pengguna untuk menghapus CodeBuild proyek atau grup laporan, atau sumber daya terkait di AWS layanan lain, seperti CloudWatch Acara. Kami merekomendasikan bahwa Anda menerapkan kebijakan ini untuk sebagian besar pengguna.

Untuk kebijakan terkelola lengkap, lihat AWSCodeBuildDeveloperAccessdi referensi kebijakan terkelola IAM.

AWSCodeBuildReadOnlyAccess

`AWSCodeBuildReadOnlyAccess`Kebijakan ini memberikan akses hanya-baca ke CodeBuild dan sumber daya terkait di layanan lain. AWS Terapkan kebijakan ini kepada pengguna yang dapat melihat dan menjalankan build, melihat proyek, dan melihat grup laporan, tetapi tidak dapat membuat perubahan apa pun padanya.

Untuk kebijakan terkelola lengkap, lihat AWSCodeBuildReadOnlyAccessdi referensi kebijakan terkelola IAM.

CodeBuild kebijakan dan pemberitahuan terkelola

CodeBuild mendukung pemberitahuan, yang dapat memberi tahu pengguna tentang perubahan penting untuk membangun proyek. Kebijakan terkelola untuk CodeBuild menyertakan pernyataan kebijakan untuk fungsionalitas notifikasi. Untuk informasi selengkapnya, lihat Apa itu notifikasi?.

Izin yang terkait dengan notifikasi dalam kebijakan terkelola hanya-baca

Kebijakan terkelola `AWSCodeBuildReadOnlyAccess` mencakup pernyataan berikut untuk mengizinkan akses penuh ke notifikasi. Pengguna yang menerapkan kebijakan terkelola ini dapat melihat pemberitahuan untuk sumber daya, tetapi tidak dapat membuat, mengelola, atau berlangganan ke sumber daya tersebut.

```
{
     "Sid": "CodeStarNotificationsPowerUserAccess",
     "Effect": "Allow",
     "Action": [
         "codestar-notifications:DescribeNotificationRule"
     ],
     "Resource": "*",
     "Condition" : {
```

```
            "ArnLike" : {"codestar-notifications:NotificationsForResource" :
"arn:aws:codebuild:*:*:project/*"}
        }
    },
    {
        "Sid": "CodeStarNotificationsListAccess",
        "Effect": "Allow",
        "Action": [
            "codestar-notifications:ListNotificationRules",
            "codestar-notifications:ListEventTypes",
            "codestar-notifications:ListTargets"
        ],
        "Resource": "*"
    }
```

Izin terkait notifikasi dalam kebijakan terkelola lainnya

Kebijakan terkelola `AWSCodeBuildDeveloperAccess` mencakup pernyataan berikut untuk mengizinkan pengguna membuat, mengedit, dan berlangganan notifikasi. Mereka tidak dapat menghapus aturan notifikasi atau mengelola tag untuk sumber daya.

```
    {
        "Sid": "CodeStarNotificationsReadWriteAccess",
        "Effect": "Allow",
        "Action": [
            "codestar-notifications:CreateNotificationRule",
            "codestar-notifications:DescribeNotificationRule",
            "codestar-notifications:UpdateNotificationRule",
            "codestar-notifications:Subscribe",
            "codestar-notifications:Unsubscribe"
        ],
        "Resource": "*",
        "Condition" : {
            "ArnLike" : {"codestar-notifications:NotificationsForResource" :
"arn:aws:codebuild:*:*:project/*"}
        }
    },
    {
        "Sid": "CodeStarNotificationsListAccess",
        "Effect": "Allow",
        "Action": [
            "codestar-notifications:ListNotificationRules",
            "codestar-notifications:ListTargets",
```

```
            "codestar-notifications:ListTagsforResource",
            "codestar-notifications:ListEventTypes"
        ],
        "Resource": "*"
    },
    {
        "Sid": "SNSTopicListAccess",
        "Effect": "Allow",
        "Action": [
            "sns:ListTopics"
        ],
        "Resource": "*"
    },
    {
        "Sid": "CodeStarNotificationsChatbotAccess",
        "Effect": "Allow",
        "Action": [
            "chatbot:DescribeSlackChannelConfigurations",
            "chatbot:ListMicrosoftTeamsChannelConfigurations"
          ],
       "Resource": "*"
    }
```

Untuk informasi selengkapnya tentang IAM dan notifikasi, lihat Identity and Access Managementuntuk AWS CodeStar Pemberitahuan.

CodeBuild update ke AWS kebijakan terkelola

Lihat detail tentang pembaruan kebijakan AWS terkelola CodeBuild sejak layanan ini mulai melacak perubahan ini. Untuk peringatan otomatis tentang perubahan pada halaman ini, berlangganan RSS feed on. AWS CodeBuild Riwayat dokumen Panduan Pengguna

Ubah	Deskripsi	Date
`AWSCodeBuildAdminAccess` dan `AWSCodeBuildDeveloperAccess` — Perbarui kebijakan yang ada	CodeBuild menambahkan `ssmmessages:OpenDataChannel` izin ke kebijakan ini untuk mendukung debugging build interaktif Session Manager.	Desember 1, 2025

Ubah	Deskripsi	Date
	`AWSCodeBuildDeveloperAccess` Kebijakan `AWSCodeBuildAdminAccess` dan sekarang menyertakan `ssmmessages:OpenDataChannel` tindakan untuk sumber daya sesi Manajer Sesi (`arn:aws:ssm:*:*:session/*`) untuk mendukung penegakan SiGv4 pada API ini WebSocket .	
`AWSCodeBuildAdminAccess` ,`AWSCodeBuildDeveloperAccess` , dan `AWSCodeBuildReadOnlyAccess` — Perbarui ke kebijakan yang ada	CodeBuild memperbarui sumber daya ke kebijakan ini. `AWSCodeBuildReadOnlyAccess` Kebijakan `AWSCodeBuildAdminAccess` ,`AWSCodeBuildDeveloperAccess` , dan telah diubah untuk memperbarui sumber daya yang ada. Sumber daya asli `arn:aws:codebuild:*` telah diperbarui ke`arn:aws:codebuild:*:*:project/*` .	November 15, 2024

Ubah	Deskripsi	Date
`AWSCodeBuildAdminAccess` ,`AWSCodeBuildDeveloperAccess` , dan `AWSCodeBuildReadOnlyAccess` — Perbarui ke kebijakan yang ada	CodeBuild menambahkan sumber daya ke kebijakan ini untuk mendukung AWS CodeConnections rebranding. `AWSCodeBuildReadOnlyAccess` Kebijakan `AWSCodeBuildAdminAccess` ,`AWSCodeBuildDeveloperAccess` , dan telah diubah untuk menambahkan sumber daya,`arn:aws:codeconnections:*:*:*` .	April 18, 2024
`AWSCodeBuildAdminAccess` dan `AWSCodeBuildDeveloperAccess` — Perbarui kebijakan yang ada	CodeBuild menambahkan izin ke kebijakan ini untuk mendukung jenis pemberitahuan tambahan menggunakan Pengembang Amazon Q di aplikasi obrolan. `AWSCodeBuildDeveloperAccess` Kebijakan `AWSCodeBuildAdminAccess` dan telah diubah untuk menambahkan izin,`chatbot:ListMicrosoftTeamsChannelConfigurations` .	16 Mei 2023
CodeBuild mulai melacak perubahan	CodeBuild mulai melacak perubahan untuk kebijakan yang AWS dikelola.	Mei 16, 2021

Customer-managed contoh kebijakan

Di bagian ini, Anda dapat menemukan contoh kebijakan pengguna yang memberikan izin untuk AWS CodeBuild tindakan. Kebijakan ini berfungsi saat Anda menggunakan CodeBuild API, AWS SDK, atau AWS CLI. Saat menggunakan konsol, Anda harus memberikan izin tambahan khusus konsol. Untuk informasi, lihat Izin yang diperlukan untuk menggunakan AWS CodeBuild konsol.

Anda dapat menggunakan contoh kebijakan IAM berikut untuk membatasi CodeBuild akses bagi pengguna dan peran Anda.

Topik

- Memungkinkan pengguna untuk mendapatkan informasi tentang proyek build
- Memungkinkan pengguna untuk mendapatkan informasi tentang armada
- Memungkinkan pengguna untuk mendapatkan informasi tentang grup laporan
- Memungkinkan pengguna untuk mendapatkan informasi tentang laporan
- Izinkan pengguna untuk membuat proyek build
- Memungkinkan pengguna untuk membuat armada
- Izinkan pengguna membuat grup laporan
- Memungkinkan pengguna untuk menghapus armada
- Izinkan pengguna untuk menghapus grup laporan
- Izinkan pengguna untuk menghapus laporan
- Izinkan pengguna untuk menghapus proyek build
- Izinkan pengguna untuk mendapatkan daftar nama proyek build
- Memungkinkan pengguna untuk mengubah informasi tentang proyek build
- Memungkinkan pengguna untuk mengubah armada
- Izinkan pengguna mengubah grup laporan
- Memungkinkan pengguna untuk mendapatkan informasi tentang build
- Izinkan pengguna mendapatkan daftar ID build untuk proyek build
- Izinkan pengguna mendapatkan daftar ID build
- Memungkinkan pengguna untuk mendapatkan daftar armada
- Izinkan pengguna untuk mendapatkan daftar grup laporan
- Memungkinkan pengguna untuk mendapatkan daftar laporan

- Memungkinkan pengguna untuk mendapatkan daftar laporan untuk grup laporan
- Memungkinkan pengguna untuk mendapatkan daftar kasus uji untuk laporan
- Izinkan pengguna untuk mulai menjalankan build
- Izinkan pengguna untuk mencoba menghentikan build
- Izinkan pengguna mencoba menghapus build
- Izinkan pengguna untuk mendapatkan informasi tentang gambar Docker yang dikelola oleh CodeBuild
- Izinkan pengguna menambahkan kebijakan izin untuk peran layanan armada
- Izinkan CodeBuild akses ke AWS layanan yang diperlukan untuk membuat antarmuka jaringan VPC
- Gunakan pernyataan penolakan untuk mencegah AWS CodeBuild dari memutuskan sambungan dari penyedia sumber

Memungkinkan pengguna untuk mendapatkan informasi tentang proyek build

Contoh pernyataan kebijakan berikut memungkinkan pengguna untuk mendapatkan informasi tentang proyek build di `us-east-2` Wilayah `123456789012` untuk memperhitungkan setiap proyek build yang dimulai dengan nama`my`:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:BatchGetProjects",
      "Resource": "arn:aws:codebuild:us-east-2:111122223333:project/my*"
    }
  ]
}
```

Memungkinkan pengguna untuk mendapatkan informasi tentang armada

Contoh pernyataan kebijakan berikut memungkinkan pengguna untuk mendapatkan informasi tentang armada di `us-east-2` Wilayah untuk akun`123456789012`:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:BatchGetFleets",
      "Resource": "arn:aws:codebuild:us-east-2:111122223333:fleet/*"
    }
  ]
}
```

Memungkinkan pengguna untuk mendapatkan informasi tentang grup laporan

Contoh pernyataan kebijakan berikut memungkinkan pengguna untuk mendapatkan informasi tentang grup laporan di `us-east-2` Wilayah untuk akun`123456789012`:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:BatchGetReportGroups",
      "Resource": "arn:aws:codebuild:us-east-2:111122223333:report-group/*"
    }
  ]
}
```

Memungkinkan pengguna untuk mendapatkan informasi tentang laporan

Contoh pernyataan kebijakan berikut memungkinkan pengguna untuk mendapatkan informasi tentang laporan di `us-east-2` Wilayah untuk akun`123456789012`:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:BatchGetReports",
      "Resource": "arn:aws:codebuild:us-east-2:111122223333:report-group/*"
    }
  ]
}
```

Izinkan pengguna untuk membuat proyek build

Contoh pernyataan kebijakan berikut memungkinkan pengguna untuk membuat proyek build dengan nama apa pun tetapi hanya di `us-east-2` Wilayah untuk akun `123456789012` dan hanya menggunakan peran CodeBuild layanan yang ditentukan:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:CreateProject",
      "Resource": "arn:aws:codebuild:us-east-2:111122223333:project/*"
    },
    {
      "Effect": "Allow",
      "Action": "iam:PassRole",
      "Resource": "arn:aws:iam::111122223333:role/CodeBuildServiceRole"
    }
  ]
}
```

Contoh pernyataan kebijakan berikut memungkinkan pengguna untuk membuat proyek build dengan nama apa pun tetapi hanya di `us-east-2` Wilayah untuk akun `123456789012` dan hanya

menggunakan peran CodeBuild layanan yang ditentukan. Ini juga memberlakukan bahwa pengguna hanya dapat menggunakan peran layanan yang ditentukan dengan AWS CodeBuild dan bukan AWS layanan lain.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:CreateProject",
      "Resource": "arn:aws:codebuild:us-east-2:111122223333:project/*"
    },
    {
      "Effect": "Allow",
      "Action": "iam:PassRole",
      "Resource": "arn:aws:iam::111122223333:role/CodeBuildServiceRole",
      "Condition": {
          "StringEquals": {"iam:PassedToService": "codebuild.amazonaws.com"}
      }
    }
  ]
}
```

Memungkinkan pengguna untuk membuat armada

Contoh pernyataan kebijakan berikut memungkinkan pengguna untuk membuat armada di `us-east-2` Wilayah untuk akun`123456789012`:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:CreateFleet",
      "Resource": "arn:aws:codebuild:us-east-2:111122223333:fleet/*"
    }
```

```
  ]
}
```

Izinkan pengguna membuat grup laporan

Contoh pernyataan kebijakan berikut memungkinkan pengguna untuk membuat grup laporan di `us-east-2` Wilayah untuk akun`123456789012`:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:CreateReportGroup",
      "Resource": "arn:aws:codebuild:us-east-2:111122223333:report-group/*"
    }
  ]
}
```

Memungkinkan pengguna untuk menghapus armada

Contoh pernyataan kebijakan berikut memungkinkan pengguna untuk menghapus armada di `us-east-2` Wilayah untuk akun`123456789012`:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:DeleteFleet",
      "Resource": "arn:aws:codebuild:us-east-2:111122223333:fleet/*"
    }
  ]
}
```

Izinkan pengguna untuk menghapus grup laporan

Contoh pernyataan kebijakan berikut memungkinkan pengguna untuk menghapus grup laporan di `us-east-2` Wilayah untuk akun`123456789012`:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:DeleteReportGroup",
      "Resource": "arn:aws:codebuild:us-east-2:111122223333:report-group/*"
    }
  ]
}
```

Izinkan pengguna untuk menghapus laporan

Contoh pernyataan kebijakan berikut memungkinkan pengguna untuk menghapus laporan di `us-east-2` Wilayah untuk akun`123456789012`:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:DeleteReport",
      "Resource": "arn:aws:codebuild:us-east-2:111122223333:report-group/*"
    }
  ]
}
```

Izinkan pengguna untuk menghapus proyek build

Contoh pernyataan kebijakan berikut memungkinkan pengguna untuk menghapus proyek build di `us-east-2` Wilayah untuk akun `123456789012` untuk setiap proyek build yang dimulai dengan nama`my`:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:DeleteProject",
      "Resource": "arn:aws:codebuild:us-east-2:111122223333:project/my*"
    }
  ]
}
```

Izinkan pengguna untuk mendapatkan daftar nama proyek build

Contoh pernyataan kebijakan berikut memungkinkan pengguna untuk mendapatkan daftar nama proyek build untuk akun yang sama:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:ListProjects",
      "Resource": "*"
    }
  ]
}
```

Memungkinkan pengguna untuk mengubah informasi tentang proyek build

Contoh pernyataan kebijakan berikut memungkinkan pengguna untuk mengubah informasi tentang proyek build dengan nama apa pun tetapi hanya di `us-east-2` Wilayah untuk akun `123456789012` dan hanya menggunakan peran AWS CodeBuild layanan yang ditentukan:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:UpdateProject",
      "Resource": "arn:aws:codebuild:us-east-2:111122223333:project/*"
    },
    {
      "Effect": "Allow",
      "Action": "iam:PassRole",
      "Resource": "arn:aws:iam::111122223333:role/CodeBuildServiceRole"
    }
  ]
}
```

Memungkinkan pengguna untuk mengubah armada

Contoh pernyataan kebijakan berikut memungkinkan pengguna untuk mengubah armada di `us-east-2` Wilayah untuk akun`123456789012`:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:UpdateFleet",
      "Resource": "arn:aws:codebuild:us-east-2:111122223333:fleet/*"
    }
  ]
```

```
}
```

Izinkan pengguna mengubah grup laporan

Contoh pernyataan kebijakan berikut memungkinkan pengguna untuk mengubah grup laporan di `us-east-2` Wilayah untuk akun`123456789012`:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:UpdateReportGroup",
      "Resource": "arn:aws:codebuild:us-east-2:111122223333:report-group/*"
    }
  ]
}
```

Memungkinkan pengguna untuk mendapatkan informasi tentang build

Contoh pernyataan kebijakan berikut memungkinkan pengguna untuk mendapatkan informasi tentang build di `us-east-2` Wilayah `123456789012` untuk akun proyek build bernama `my-build-project` dan`my-other-build-project`:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:BatchGetBuilds",
      "Resource": [
        "arn:aws:codebuild:us-east-2:111122223333:project/my-build-project",
        "arn:aws:codebuild:us-east-2:111122223333:project/my-other-build-project"
      ]
    }
```

```
  ]
}
```

Izinkan pengguna mendapatkan daftar ID build untuk proyek build

Contoh pernyataan kebijakan berikut memungkinkan pengguna untuk mendapatkan daftar ID build di `us-east-2` Region `123456789012` untuk akun proyek build bernama `my-build-project` dan`my-other-build-project`:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:ListBuildsForProject",
      "Resource": [
        "arn:aws:codebuild:us-east-2:111122223333:project/my-build-project",
        "arn:aws:codebuild:us-east-2:111122223333:project/my-other-build-project"
      ]
    }
  ]
}
```

Izinkan pengguna mendapatkan daftar ID build

Contoh pernyataan kebijakan berikut memungkinkan pengguna mendapatkan daftar semua ID build untuk akun yang sama:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:ListBuilds",
      "Resource": "*"
```

```
    }
  ]
}
```

Memungkinkan pengguna untuk mendapatkan daftar armada

Contoh pernyataan kebijakan berikut memungkinkan pengguna untuk mendapatkan daftar armada di `us-east-2` Wilayah untuk akun`123456789012`:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:ListFleets",
      "Resource": "*"
    }
  ]
}
```

Izinkan pengguna untuk mendapatkan daftar grup laporan

Contoh pernyataan kebijakan berikut memungkinkan pengguna untuk mendapatkan daftar grup laporan di `us-east-2` Wilayah untuk akun`123456789012`:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:ListReportGroups",
      "Resource": "*"
    }
  ]
}
```

Memungkinkan pengguna untuk mendapatkan daftar laporan

Contoh pernyataan kebijakan berikut memungkinkan pengguna untuk mendapatkan daftar laporan di `us-east-2` Wilayah untuk akun`123456789012`:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:ListReports",
      "Resource": "*"
    }
  ]
}
```

Memungkinkan pengguna untuk mendapatkan daftar laporan untuk grup laporan

Contoh pernyataan kebijakan berikut memungkinkan pengguna untuk mendapatkan daftar laporan untuk grup laporan di `us-east-2` Wilayah untuk akun`123456789012`:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:ListReportsForReportGroup",
      "Resource": "arn:aws:codebuild:us-east-2:111122223333:report-group/*"
    }
  ]
}
```

Memungkinkan pengguna untuk mendapatkan daftar kasus uji untuk laporan

Contoh pernyataan kebijakan berikut memungkinkan pengguna untuk mendapatkan daftar kasus uji untuk laporan di `us-east-2` Wilayah untuk akun`123456789012`:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:DescribeTestCases",
      "Resource": "arn:aws:codebuild:us-east-2:111122223333:report-group/*"
    }
  ]
}
```

Izinkan pengguna untuk mulai menjalankan build

Contoh pernyataan kebijakan berikut memungkinkan pengguna menjalankan build di `us-east-2` Region `123456789012` untuk akun proyek build yang dimulai dengan nama: `my`

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:StartBuild",
      "Resource": "arn:aws:codebuild:us-east-2:111122223333:project/my*"
    }
  ]
}
```

Izinkan pengguna untuk mencoba menghentikan build

Contoh pernyataan kebijakan berikut memungkinkan pengguna untuk mencoba menghentikan menjalankan build hanya di `us-east-2` wilayah `123456789012` untuk memperhitungkan setiap proyek build yang dimulai dengan nama: `my`

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:StopBuild",
      "Resource": "arn:aws:codebuild:us-east-2:111122223333:project/my*"
    }
  ]
}
```

Izinkan pengguna mencoba menghapus build

Contoh pernyataan kebijakan berikut memungkinkan pengguna untuk mencoba menghapus build hanya di `us-east-2` Region untuk akun `123456789012` untuk setiap proyek build yang dimulai dengan nama: `my`

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:BatchDeleteBuilds",
      "Resource": "arn:aws:codebuild:us-east-2:111122223333:project/my*"
    }
  ]
}
```

Izinkan pengguna untuk mendapatkan informasi tentang gambar Docker yang dikelola oleh CodeBuild

Contoh pernyataan kebijakan berikut memungkinkan pengguna untuk mendapatkan informasi tentang semua gambar Docker yang dikelola oleh CodeBuild:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:ListCuratedEnvironmentImages",
      "Resource": "*"
    }
  ]
}
```

Izinkan pengguna menambahkan kebijakan izin untuk peran layanan armada

Contoh pernyataan kebijakan sumber daya berikut memungkinkan pengguna untuk menambahkan kebijakan izin VPC untuk peran layanan armada:

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "CodeBuildFleetVpcCreateNI",
            "Effect": "Allow",
            "Action": [
                "ec2:CreateNetworkInterface"
            ],
            "Resource": [
                "arn:aws:ec2:us-west-2:111122223333:subnet/subnet-id-1",
                "arn:aws:ec2:us-west-2:111122223333:security-group/security-
group-id-1",
                "arn:aws:ec2:us-west-2:111122223333:network-interface/*"
```

```
            ]
        },
        {
            "Sid": "CodeBuildFleetVpcPermission",
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeDhcpOptions",
                "ec2:DescribeNetworkInterfaces",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeSubnets",
                "ec2:DescribeVpcs",
                "ec2:ModifyNetworkInterfaceAttribute",
                "ec2:DeleteNetworkInterface"
            ],
            "Resource": "*"
        },
        {
            "Sid": "CodeBuildFleetVpcNIPermission",
            "Effect": "Allow",
            "Action": [
                "ec2:CreateNetworkInterfacePermission"
            ],
            "Resource": "arn:aws:ec2:us-west-2:111122223333:network-interface/*",
            "Condition": {
                "StringEquals": {
                    "ec2:Subnet": [
                        "arn:aws:ec2:us-west-2:111122223333:subnet/subnet-id-1"
                    ]
                }
            }
        }
    ]
}
```

Contoh pernyataan kebijakan sumber daya berikut memungkinkan pengguna menambahkan kebijakan izin Amazon Managed Image (AMI) khusus untuk peran layanan armada:

JSON

```
{
    "Version":"2012-10-17",
```

```
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ec2:DescribeImages",
            "Resource": "*"
        }
    ]
}
```

Contoh pernyataan kebijakan kepercayaan berikut memungkinkan pengguna untuk menambahkan kebijakan izin untuk peran layanan armada:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "CodeBuildFleetVPCTrustPolicy",
      "Effect": "Allow",
      "Principal": {
        "Service": "codebuild.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "111122223333"
        }
      }
    }
  ]
}
```

Izinkan CodeBuild akses ke AWS layanan yang diperlukan untuk membuat antarmuka jaringan VPC

Contoh pernyataan kebijakan berikut memberikan AWS CodeBuild izin untuk membuat antarmuka jaringan di VPC dengan dua subnet:

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateNetworkInterface",
                "ec2:DescribeDhcpOptions",
                "ec2:DescribeNetworkInterfaces",
                "ec2:DeleteNetworkInterface",
                "ec2:DescribeSubnets",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeVpcs"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateNetworkInterfacePermission"
            ],
            "Resource": "arn:aws:ec2:us-west-2:111122223333:network-interface/*",
            "Condition": {
                "StringEquals": {
                    "ec2:AuthorizedService": "codebuild.amazonaws.com"
                },
                "ArnEquals": {
                    "ec2:Subnet": [
                        "arn:aws:ec2:us-west-2:111122223333:subnet/subnet-id-1",
                        "arn:aws:ec2:us-west-2:111122223333:subnet/subnet-id-2"
                    ]
                }
            }
        }
    ]
}
```

Gunakan pernyataan penolakan untuk mencegah AWS CodeBuild dari memutuskan sambungan dari penyedia sumber

Contoh pernyataan kebijakan berikut menggunakan pernyataan penolakan untuk mencegah terputus AWS CodeBuild dari penyedia sumber. Ini menggunakan`codebuild:DeleteOAuthToken`, yang merupakan kebalikan dari `codebuild:PersistOAuthToken` dan`codebuild:ImportSourceCredentials`, untuk terhubung dengan penyedia sumber. Untuk informasi selengkapnya, lihat Izin yang diperlukan untuk AWS CodeBuild konsol untuk terhubung ke penyedia sumber.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": "codebuild:DeleteOAuthToken",
      "Resource": "*"
    }
  ]
}
```

AWS CodeBuild referensi izin

Anda dapat menggunakan kunci kondisi AWS-wide dalam AWS CodeBuild kebijakan Anda untuk menyatakan kondisi. Untuk daftar, lihat Kunci yang Tersedia di Panduan Pengguna IAM.

Anda menentukan tindakan di bidang `Action` kebijakan. Untuk menentukan tindakan, gunakan prefiks `codebuild:` diikuti dengan nama operasi API (misalnya `codebuild:CreateProject` dan `codebuild:StartBuild`). Untuk menetapkan beberapa tindakan dalam satu pernyataan, pisahkan dengan koma (misalnya, `"Action": [ "codebuild:CreateProject", "codebuild:StartBuild" ]`).

Menggunakan Karakter Wildcard

Anda menentukan ARN, dengan atau tanpa karakter wildcard (*), sebagai nilai sumber daya dalam bidang `Resource` kebijakan. Anda bisa menggunakan wildcard untuk menentukan beberapa tindakan atau sumber daya. Misalnya, `codebuild:*` menentukan semua CodeBuild tindakan dan

`codebuild:Batch*` menentukan semua CodeBuild tindakan yang dimulai dengan kata. `Batch` Contoh berikut memberikan akses ke semua proyek build dengan nama yang dimulai dengan`my`:

```
arn:aws:codebuild:us-east-2:123456789012:project/my*
```

CodeBuild Operasi API dan izin yang diperlukan untuk tindakan

BatchDeleteBuilds

Tindakan: `codebuild:BatchDeleteBuilds`

Diperlukan untuk menghapus build.

Sumber daya: `arn:aws:codebuild:`*`region-ID`*`:`*`account-ID`*`:project/`*`project-name`*

BatchGetBuilds

Tindakan: `codebuild:BatchGetBuilds`

Diperlukan untuk mendapatkan informasi tentang build.

Sumber daya: `arn:aws:codebuild:`*`region-ID`*`:`*`account-ID`*`:project/`*`project-name`*

BatchGetProjects

Tindakan: `codebuild:BatchGetProjects`

Diperlukan untuk mendapatkan informasi tentang membangun proyek.

Sumber daya: `arn:aws:codebuild:`*`region-ID`*`:`*`account-ID`*`:project/`*`project-name`*

BatchGetReportGroups

Tindakan: `codebuild:BatchGetReportGroups`

Diperlukan untuk mendapatkan informasi tentang grup laporan.

Sumber daya: `arn:aws:codebuild:`*`region-ID`*`:`*`account-ID`*`:report-group/`*`report-group-name`*

BatchGetReports

Tindakan: `codebuild:BatchGetReports`

Diperlukan untuk mendapatkan informasi tentang laporan.

Sumber daya: `arn:aws:codebuild:`*`region-ID`*`:`*`account-ID`*`:report-group/`*`report-group-name`*

BatchPutTestCases[1]

Tindakan: `codebuild:BatchPutTestCases`

Diperlukan untuk membuat atau memperbarui laporan pengujian.

Sumber daya: `arn:aws:codebuild:`*`region-ID`*`:`*`account-ID`*`:report-group/`*`report-group-name`*

CreateProject

Tindakan:`codebuild:CreateProject`, `iam:PassRole`

Diperlukan untuk membuat proyek pembangunan.

Sumber Daya:

- `arn:aws:codebuild:`*`region-ID`*`:`*`account-ID`*`:project/`*`project-name`*
- `arn:aws:iam::`*`account-ID`*`:role/`*`role-name`*

CreateReport[1]

Tindakan: `codebuild:CreateReport`

Diperlukan untuk membuat laporan pengujian.

Sumber daya: `arn:aws:codebuild:`*`region-ID`*`:`*`account-ID`*`:report-group/`*`report-group-name`*

CreateReportGroup

Tindakan: `codebuild:CreateReportGroup`

Diperlukan untuk membuat grup laporan.

Sumber daya: `arn:aws:codebuild:`*`region-ID`*`:`*`account-ID`*`:report-group/`*`report-group-name`*

CreateWebhook

Tindakan: `codebuild:CreateWebhook`

Diperlukan untuk membuat webhook.

Sumber daya: `arn:aws:codebuild:region-ID:account-ID:project/project-name`

DeleteProject

Tindakan: `codebuild:DeleteProject`

Diperlukan untuk menghapus CodeBuild proyek.

Sumber daya: `arn:aws:codebuild:region-ID:account-ID:project/project-name`

DeleteReport

Tindakan: `codebuild:DeleteReport`

Diperlukan untuk menghapus laporan.

Sumber daya: `arn:aws:codebuild:region-ID:account-ID:report-group/report-group-name`

DeleteReportGroup

Tindakan: `codebuild:DeleteReportGroup`

Diperlukan untuk menghapus grup laporan.

Sumber daya: `arn:aws:codebuild:region-ID:account-ID:report-group/report-group-name`

DeleteSourceCredentials

Tindakan: `codebuild:DeleteSourceCredentials`

Diperlukan untuk menghapus sekumpulan `SourceCredentialsInfo` objek yang berisi informasi tentang kredensyal untuk repositori GitHub, Server GitHub Perusahaan, atau Bitbucket.

Sumber daya: *

DeleteWebhook

Tindakan: `codebuild:DeleteWebhook`

Diperlukan untuk membuat webhook.

Sumber daya: `arn:aws:codebuild:region-ID:account-ID:project/project-name`

DescribeTestCases

Tindakan: `codebuild:DescribeTestCases`

Diperlukan untuk mengembalikan daftar kasus uji paginasi.

Sumber daya: `arn:aws:codebuild:`*`region-ID`*`:`*`account-ID`*`:report-group/`*`report-group-name`*

ImportSourceCredentials

Tindakan: `codebuild:ImportSourceCredentials`

Diperlukan untuk mengimpor sekumpulan `SourceCredentialsInfo` objek yang berisi informasi tentang kredensyal untuk repositori GitHub, Server GitHub Perusahaan, atau Bitbucket.

Sumber daya: *

InvalidateProjectCache

Tindakan: `codebuild:InvalidateProjectCache`

Diperlukan untuk mengatur ulang cache untuk sebuah proyek.

Sumber daya: `arn:aws:codebuild:`*`region-ID`*`:`*`account-ID`*`:project/`*`project-name`*

ListBuildBatches

Tindakan: `codebuild:ListBuildBatches`

Diperlukan untuk mendapatkan daftar ID batch build.

Sumber daya: *

ListBuildBatchesForProject

Tindakan: `codebuild:ListBuildBatchesForProject`

Diperlukan untuk mendapatkan daftar ID batch build untuk proyek tertentu.

Sumber daya: `arn:aws:codebuild:`*`region-ID`*`:`*`account-ID`*`:project/`*`project-name`*

ListBuilds

Tindakan: `codebuild:ListBuilds`

Diperlukan untuk mendapatkan daftar ID build.

Sumber daya: *

ListBuildsForProject

Tindakan: `codebuild:ListBuildsForProject`

Diperlukan untuk mendapatkan daftar ID build untuk proyek build.

Sumber daya: `arn:aws:codebuild:`*`region-ID`*`:`*`account-ID`*`:project/`*`project-name`*

ListCuratedEnvironmentImages

Tindakan: `codebuild:ListCuratedEnvironmentImages`

Diperlukan untuk mendapatkan informasi tentang semua gambar Docker yang dikelola oleh AWS CodeBuild.

Sumber daya: * (diperlukan, tetapi tidak mengacu pada sumber daya yang dapat dialamatkan AWS)

ListProjects

Tindakan: `codebuild:ListProjects`

Diperlukan untuk mendapatkan daftar nama proyek build.

Sumber daya: *

ListReportGroups

Tindakan: `codebuild:ListReportGroups`

Diperlukan untuk mendapatkan daftar grup laporan.

Sumber daya: *

ListReports

Tindakan: `codebuild:ListReports`

Diperlukan untuk mendapatkan daftar laporan.

Sumber daya: *

ListReportsForReportGroup

Tindakan: `codebuild:ListReportsForReportGroup`

Diperlukan untuk mendapatkan daftar laporan untuk grup laporan.

Sumber daya: `arn:aws:codebuild:`*`region-ID`*`:`*`account-ID`*`:report-group/`*`report-group-name`*

RetryBuild

Tindakan: `codebuild:RetryBuild`

Diperlukan untuk mencoba lagi build.

Sumber daya: `arn:aws:codebuild:`*`region-ID`*`:`*`account-ID`*`:project/`*`project-name`*

StartBuild

Tindakan: `codebuild:StartBuild`

Diperlukan untuk mulai menjalankan build.

Sumber daya: `arn:aws:codebuild:`*`region-ID`*`:`*`account-ID`*`:project/`*`project-name`*

StopBuild

Tindakan: `codebuild:StopBuild`

Diperlukan untuk mencoba berhenti menjalankan build.

Sumber daya: `arn:aws:codebuild:`*`region-ID`*`:`*`account-ID`*`:project/`*`project-name`*

UpdateProject

Tindakan:`codebuild:UpdateProject`, `iam:PassRole`

Diperlukan untuk mengubah informasi tentang build.

Sumber Daya:

- `arn:aws:codebuild:`*`region-ID`*`:`*`account-ID`*`:project/`*`project-name`*
- `arn:aws:iam::`*`account-ID`*`:role/`*`role-name`*

UpdateProjectVisibility

Tindakan:`codebuild:UpdateProjectVisibility`, `iam:PassRole`

Diperlukan untuk mengubah visibilitas publik dari bangunan proyek.

Sumber Daya:

- `arn:aws:codebuild:`*`region-ID`*`:`*`account-ID`*`:project/`*`project-name`*
- `arn:aws:iam::`*`account-ID`*`:role/`*`role-name`*

UpdateReport[1]

Aksi: `codebuild:UpdateReport`

Diperlukan untuk membuat atau memperbarui laporan pengujian.

Sumber daya: `arn:aws:codebuild:`*`region-ID`*`:`*`account-ID`*`:report-group/`*`report-group-name`*

UpdateReportGroup

Aksi: `codebuild:UpdateReportGroup`

Diperlukan untuk memperbarui grup laporan.

Sumber daya: `arn:aws:codebuild:`*`region-ID`*`:`*`account-ID`*`:report-group/`*`report-group-name`*

UpdateWebhook

Aksi: `codebuild:UpdateWebhook`

Diperlukan untuk memperbarui webhook.

Sumber daya: `arn:aws:codebuild:`*`region-ID`*`:`*`account-ID`*`:project/`*`project-name`*

[1] Digunakan untuk izin saja. Tidak ada API untuk tindakan ini.

Menggunakan tag untuk mengontrol akses AWS CodeBuild sumber daya

Kondisi dalam pernyataan kebijakan IAM adalah bagian dari sintaks yang dapat Anda gunakan untuk menentukan izin untuk CodeBuild tindakan berbasis proyek. Anda dapat membuat kebijakan yang mengizinkan atau menolak tindakan pada proyek berdasarkan tag yang terkait dengan proyek tersebut, lalu menerapkan kebijakan tersebut ke grup IAM yang Anda konfigurasikan untuk mengelola

pengguna. Untuk informasi tentang menerapkan tag ke proyek menggunakan konsol atau AWS CLI, lihatMembuat proyek pembangunan di AWS CodeBuild. Untuk informasi tentang menerapkan tag menggunakan CodeBuild SDK, lihat CreateProject dan Tag di Referensi CodeBuild API. Untuk informasi tentang penggunaan tag untuk mengontrol akses ke AWS sumber daya, lihat Mengontrol Akses ke AWS Sumber Daya Menggunakan Tag Sumber Daya di Panduan Pengguna IAM.

> **Important**
>
> Saat menggunakan fitur kapasitas cadangan, data yang di-cache pada instance armada, termasuk file sumber, lapisan Docker, dan direktori cache yang ditentukan dalam buildspec, dapat diakses oleh proyek lain dalam akun yang sama. Ini dirancang dan memungkinkan proyek dalam akun yang sama untuk berbagi instance armada.

Example Contoh 1: Batasi tindakan CodeBuild proyek berdasarkan tag sumber daya

Contoh berikut menyangkal semua `BatchGetProjects` tindakan pada proyek yang ditandai dengan kunci `Environment` dengan nilai kunci dari. `Production` Administrator pengguna harus melampirkan kebijakan IAM ini selain kebijakan pengguna terkelola untuk pengguna yang tidak sah. Kunci `aws:ResourceTag` kondisi digunakan untuk mengontrol akses ke sumber daya berdasarkan tag mereka.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "codebuild:BatchGetProjects"
      ],
      "Resource": "*",
      "Condition": {
        "ForAnyValue:StringEquals": {
          "aws:ResourceTag/Environment": "Production"
        }
      }
    }
  ]
```

```
}
```

Example Contoh 2: Batasi tindakan CodeBuild proyek berdasarkan tag permintaan

Kebijakan berikut menolak izin pengguna untuk `CreateProject` tindakan jika permintaan berisi tag dengan kunci `Environment` dan nilai `Production` kunci. Selain itu, kebijakan mencegah pengguna yang tidak sah ini memodifikasi proyek dengan menggunakan kunci `aws:TagKeys` kondisi untuk tidak mengizinkan `UpdateProject` jika permintaan berisi tag dengan kunci. `Environment` Administrator harus melampirkan kebijakan IAM ini selain kebijakan pengguna terkelola kepada pengguna yang tidak berwenang untuk melakukan tindakan ini. Kunci `aws:RequestTag` kondisi digunakan untuk mengontrol tag mana yang dapat diteruskan dalam permintaan IAM

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "codebuild:CreateProject"
      ],
      "Resource": "*",
      "Condition": {
        "ForAnyValue:StringEquals": {
          "aws:RequestTag/Environment": "Production"
        }
      }
    },
    {
      "Effect": "Deny",
      "Action": [
        "codebuild:UpdateProject"
      ],
      "Resource": "*",
      "Condition": {
        "ForAnyValue:StringEqualsIgnoreCase": {
          "aws:TagKeys": ["Environment"]
        }
      }
    }
```

```
  ]
}
```

Example Contoh 3: Tolak atau izinkan tindakan pada grup laporan berdasarkan tag sumber daya

Anda dapat membuat kebijakan yang mengizinkan atau menolak tindakan pada CodeBuild sumber daya (proyek dan grup laporan) berdasarkan AWS tag yang terkait dengan sumber daya tersebut, lalu menerapkan kebijakan tersebut ke grup IAM yang Anda konfigurasikan untuk mengelola pengguna. Misalnya, Anda dapat membuat kebijakan yang menolak semua CodeBuild tindakan pada grup laporan apa pun dengan kunci AWS tag `Status` dan nilai kunci`Secret`, lalu menerapkan kebijakan tersebut ke grup IAM yang Anda buat untuk pengembang umum ()*`Developers`*. Anda kemudian perlu memastikan bahwa pengembang yang mengerjakan grup laporan yang diberi tag tersebut bukan anggota *`Developers`* grup umum tersebut, melainkan milik grup IAM berbeda yang tidak memiliki kebijakan restriktif yang diterapkan (). `SecretDevelopers`

Contoh berikut menyangkal semua CodeBuild tindakan pada grup laporan yang ditandai dengan kunci `Status` dan nilai kunci: `Secret`

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "codebuild:BatchGetReportGroups",
        "codebuild:CreateReportGroup",
        "codebuild:DeleteReportGroup",
        "codebuild:ListReportGroups",
        "codebuild:ListReportsForReportGroup",
        "codebuild:UpdateReportGroup"
      ],
      "Resource": "*",
      "Condition": {
        "StringLike": {
          "aws:RequestedRegion": "us-east-1"
        }
      }
```

```
    }
  ]
}
```

Example Contoh 4: Batasi CodeBuild tindakan AWSCodeBuildDeveloperAccess berdasarkan tag sumber daya

Anda dapat membuat kebijakan yang mengizinkan CodeBuild tindakan pada semua grup laporan dan proyek yang tidak ditandai dengan tag tertentu. Misalnya, kebijakan berikut memungkinkan AWSCodeBuildDeveloperAccess izin yang setara untuk semua grup laporan dan proyek kecuali yang ditandai dengan tag yang ditentukan:

JSON

```
{
   "Version":"2012-10-17",
   "Statement": [
      {
         "Effect": "Allow",
         "Action": [
            "codebuild:StartBuild",
            "codebuild:StopBuild",
            "codebuild:BatchGet*",
            "codebuild:GetResourcePolicy",
            "codebuild:DescribeTestCases",
            "codebuild:List*",
            "codecommit:GetBranch",
            "codecommit:GetCommit",
            "codecommit:GetRepository",
            "codecommit:ListBranches",
            "cloudwatch:GetMetricStatistics",
            "events:DescribeRule",
            "events:ListTargetsByRule",
            "events:ListRuleNamesByTarget",
            "logs:GetLogEvents",
            "s3:GetBucketLocation",
            "s3:ListAllMyBuckets"
         ],
         "Resource": "*",
         "Condition": {
            "StringNotEquals": {
```

```
                "aws:ResourceTag/Status": "Secret",
                "aws:ResourceTag/Team": "Saanvi"
            }
        }
    }
  ]
}
```

Menampilkan sumber daya di konsol

AWS CodeBuild Konsol memerlukan `ListRepositories` izin untuk menampilkan daftar repositori untuk AWS akun Anda di AWS Wilayah tempat Anda masuk. Konsol juga termasuk fungsi Pergi ke sumber daya untuk secara cepat melakukan pencarian sensitif huruf besar/kecil untuk sumber daya. Pencarian ini dilakukan di AWS akun Anda di AWS Wilayah tempat Anda masuk. Sumber daya berikut ditampilkan di seluruh layanan berikut:

- AWS CodeBuild: Bangun proyek
- AWS CodeCommit: Repositori
- AWS CodeDeploy: Aplikasi
- AWS CodePipeline: Alur

Untuk melakukan pencarian ini di sumber daya di semua layanan, Anda harus memiliki izin berikut:

- CodeBuild: `ListProjects`
- CodeCommit: `ListRepositories`
- CodeDeploy: `ListApplications`
- CodePipeline: `ListPipelines`

Hasil tidak dikembalikan untuk sumber daya layanan jika Anda tidak memiliki izin untuk layanan tersebut. Bahkan jika Anda memiliki izin untuk melihat sumber daya, beberapa sumber daya tidak dikembalikan jika ada eksplisit `Deny` untuk melihat sumber daya tersebut.

Validasi kepatuhan untuk AWS CodeBuild

Pengalaman AWS pendaftaran baru kami tidak dirancang untuk beban kerja yang diatur. Jika Anda menggunakan pengalaman AWS pendaftaran baru kami, tetapi Anda ingin menggunakannya AWS

untuk beban kerja yang diatur, Anda dapat mendaftar untuk AWS (lanjutan) atau mengaktifkan fitur lanjutan untuk lingkungan Anda AWS .

Untuk mempelajari apakah an Layanan AWS berada dalam lingkup program kepatuhan tertentu, lihat Layanan AWS di Lingkup oleh Program Kepatuhan Layanan AWS dan pilih program kepatuhan yang Anda minati. Untuk informasi umum, lihat Program AWS Kepatuhan Program AWS .

Anda dapat mengunduh laporan audit pihak ketiga menggunakan AWS Artifact. Untuk informasi selengkapnya, lihat Mengunduh Laporan di AWS Artifact .

Tanggung jawab kepatuhan Anda saat menggunakan Layanan AWS ditentukan oleh sensitivitas data Anda, tujuan kepatuhan perusahaan Anda, dan hukum dan peraturan yang berlaku. Untuk informasi selengkapnya tentang tanggung jawab kepatuhan Anda saat menggunakan Layanan AWS, lihat Dokumentasi AWS Keamanan.

Ketahanan di AWS CodeBuild

Infrastruktur AWS global dibangun di sekitar AWS Wilayah dan Zona Ketersediaan. AWS Wilayah menyediakan beberapa Availability Zone yang terpisah secara fisik dan terisolasi, yang terhubung dengan latensi rendah, throughput tinggi, dan jaringan yang sangat redundan. Dengan Zona Ketersediaan, Anda dapat merancang dan mengoperasikan aplikasi dan basis data yang secara otomatis melakukan failover di antara Zona Ketersediaan tanpa gangguan. Zona Ketersediaan memiliki ketersediaan dan toleransi kesalahan yang lebih baik, dan dapat diskalakan dibandingkan infrastruktur biasa yang terdiri dari satu atau beberapa pusat data.

Untuk informasi selengkapnya tentang AWS Wilayah dan Availability Zone, lihat infrastruktur AWS global.

Keamanan Infrastruktur di AWS CodeBuild

Sebagai layanan terkelola, AWS CodeBuild dilindungi oleh keamanan jaringan AWS global. Untuk informasi tentang layanan AWS keamanan dan cara AWS melindungi infrastruktur, lihat Keamanan AWS Cloud. Untuk mendesain AWS lingkungan Anda menggunakan praktik terbaik untuk keamanan infrastruktur, lihat Perlindungan Infrastruktur dalam Kerangka Kerja yang AWS Diarsiteksikan dengan Baik Pilar Keamanan.

Anda menggunakan panggilan API yang AWS dipublikasikan untuk mengakses CodeBuild melalui jaringan. Klien harus mendukung hal-hal berikut:

- Keamanan Lapisan Pengangkutan (TLS). Kami mensyaratkan TLS 1.2 dan menganjurkan TLS 1.3.
- Cipher suite dengan perfect forward secrecy (PFS) seperti DHE (Ephemeral) atau ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Diffie-Hellman Sebagian besar sistem modern seperti Java 7 dan versi lebih baru mendukung mode-mode ini.

Akses penyedia sumber Anda di CodeBuild

Untuk GitHub atau Server GitHub Perusahaan, Anda menggunakan token akses pribadi, rahasia Secrets Manager, koneksi, atau aplikasi OAuth untuk mengakses penyedia sumber. Untuk Bitbucket, Anda menggunakan token akses, kata sandi aplikasi, token API, rahasia Secrets Manager, koneksi, atau aplikasi OAuth untuk mengakses penyedia sumber.

Topik

- Buat dan simpan token dalam rahasia Secrets Manager
- GitHub dan akses Server GitHub Perusahaan di CodeBuild
- Akses Bitbucket di CodeBuild
- GitLab akses di CodeBuild

Buat dan simpan token dalam rahasia Secrets Manager

Jika Anda memilih untuk menyimpan token akses menggunakan Secrets Manager, Anda dapat menggunakan koneksi rahasia yang ada atau membuat rahasia baru. Untuk membuat rahasia baru, lakukan hal berikut:

Konsol Manajemen AWS

Untuk membuat rahasia Secrets Manager di Konsol Manajemen AWS

1. Untuk penyedia Sumber, pilih Bitbucket, GitHub, atau GitHub Enterprise.
2. Untuk Credential, lakukan salah satu hal berikut:
   - Pilih Kredensi sumber default untuk menggunakan kredensi sumber default akun Anda untuk diterapkan ke semua proyek.
     a. Jika Anda tidak tersambung ke penyedia sumber, pilih Kelola kredensi sumber default.
     b. Untuk tipe Credential, pilih jenis kredensi selain. AWS CodeConnections

c. Untuk Layanan, pilih Secrets Manager dan untuk Rahasia pilih Rahasia baru.

d. Dalam nama Rahasia, masukkan nama rahasia Anda.

e. Dalam Deskripsi rahasia - opsional, masukkan deskripsi untuk rahasia Anda.

f. Bergantung pada penyedia sumber yang Anda pilih, masukkan token atau nama pengguna dan kata sandi aplikasi Anda, lalu pilih Simpan. Untuk token API Bitbucket, masukkan alamat email akun Atlassian Anda dan token API sebagai gantinya.

- Pilih Kredensi sumber kustom untuk menggunakan kredensi sumber kustom untuk mengganti setelan default akun Anda.

a. Untuk tipe Credential, pilih jenis kredensi selain. AWS CodeConnections

b. Dalam Koneksi, pilih Buat rahasia.

c. Dalam nama Rahasia, masukkan nama rahasia Anda.

d. Dalam Deskripsi rahasia - opsional, masukkan deskripsi untuk rahasia Anda.

e. Bergantung pada penyedia sumber yang Anda pilih, masukkan token atau nama pengguna dan kata sandi aplikasi Anda, lalu pilih Buat. Untuk token API Bitbucket, masukkan alamat email akun Atlassian Anda dan token API sebagai gantinya.

AWS CLI

Untuk membuat rahasia Secrets Manager di AWS CLI

- Buka terminal (Linux, macOS, atau Unix) atau command prompt (Windows). Gunakan create-secret perintah AWS CLI untuk menjalankan Secrets Manager.

```
aws secretsmanager create-secret --region <aws-region> \
            --name '<secret-name>' \
            --description '<secret-description>' \
            --secret-string '{
                "ServerType":"<server-type>",
                "AuthType":"<auth-type>",
                "Token":"<token>"
                }' \
            --tags Key=codebuild:source,Value='' \
                Key=codebuild:source:type,Value=<type> \
                Key=codebuild:source:provider,Value=<provider>
```

Rahasia CodeBuild Secrets Manager yang menerima harus berada di akun dan AWS Wilayah yang sama dengan CodeBuild proyek dan harus dalam format JSON berikut:

```
{
            "ServerType": ServerType,
            "AuthType: AuthType,
            "Token": string,
            "Username": string // Optional and is only used for Bitbucket app
 password or API token
        }
```

Bidang	Nilai yang valid	Deskripsi
ServerType	GITHUB GITHUB_ENTERPRISE BITBUCKET	Penyedia sumber pihak ketiga untuk rahasia Secrets Manager Anda.
AuthType	PERSONAL_ACCESS_TOKEN DASAR_AUTH	Jenis token akses yang digunakan oleh kredensil. Untuk GitHub, hanya PERSONAL_ACCESS_TOKEN yang valid. BASIC_AUTH hanya berlaku untuk kata sandi aplikasi Bitbucket atau token API.
Token	*string*	Untuk GitHub atau GitHub Enterprise, ini adalah token akses pribadi. Untuk Bitbucket, ini adalah token akses, kata sandi aplikasi Bitbucket, atau token API Bitbucket.

Bidang	Nilai yang valid	Deskripsi
nama pengguna	*string*	Nama pengguna Bitbucket saat AuthType adalah BASIC_AUTH. Jika Anda menggunakan token API Bitbucket, ini adalah alamat email yang terkait dengan akun Atlassian Anda. Parameter ini tidak berlaku untuk jenis penyedia sumber lainnya.

Selain itu, CodeBuild gunakan tag sumber daya berikut pada rahasia untuk memastikan rahasia mudah dipilih saat membuat atau mengedit proyek.

Kunci tanda	Nilai tanda	Deskripsi
codebuild: sumber: penyedia	github github_enterprise bitbucket	Memberitahu penyedia CodeBuild mana rahasia ini dimaksudkan.
codebuild: sumber: jenis	personal_access_token dasar_auth	CodeBuild Memberitahu jenis token akses dalam rahasia ini.

GitHub dan akses Server GitHub Perusahaan di CodeBuild

Untuk GitHub, Anda dapat menggunakan token akses pribadi, aplikasi OAuth, rahasia Secrets Manager, atau koneksi GitHub Aplikasi untuk mengakses penyedia sumber. Untuk GitHub Enterprise Server, Anda dapat menggunakan token akses pribadi, rahasia Secrets Manager, atau koneksi GitHub Aplikasi untuk mengakses penyedia sumber.

Topik

- GitHub Koneksi aplikasi untuk GitHub dan Server GitHub Perusahaan
- GitHub dan token akses Server GitHub Perusahaan
- GitHub Aplikasi OAuth

GitHub Koneksi aplikasi untuk GitHub dan Server GitHub Perusahaan

Anda dapat menggunakan GitHub Aplikasi untuk terhubung CodeBuild. GitHub Koneksi aplikasi didukung melalui AWS CodeConnections.

Akses penyedia sumber memungkinkan Anda memicu build dengan berlangganan GitHub acara webhook menggunakan CreateWebhook, atau menggunakannyaTutorial: Konfigurasikan pelari CodeBuild-hosted GitHub Tindakan. CodeBuild

> **Note**
>
> AWS CodeConnections tersedia di wilayah yang lebih sedikit dari CodeBuild. Anda dapat menggunakan koneksi lintas wilayah di CodeBuild. Koneksi yang dibuat di Wilayah keikutsertaan tidak dapat digunakan di Wilayah lain. Untuk informasi lebih lanjut, lihat AWS CodeConnections kuota dan titik akhir.

Topik

- Langkah 1: Buat koneksi ke GitHub App (konsol)
- Langkah 2: Berikan akses peran IAM CodeBuild proyek untuk menggunakan koneksi
- Langkah 3: Konfigurasikan CodeBuild untuk menggunakan koneksi baru
- Memecahkan masalah dengan Aplikasi GitHub

Langkah 1: Buat koneksi ke GitHub App (konsol)

Gunakan langkah-langkah ini untuk menggunakan CodeBuild konsol guna menambahkan koneksi untuk proyek Anda GitHub.

Untuk membuat koneksi ke GitHub

- Ikuti petunjuk di Panduan Pengguna Alat Pengembang untuk Membuat koneksi ke GitHub.

> **Note**
>
> Alih-alih membuat atau menggunakan koneksi yang ada di akun Anda, Anda dapat menggunakan koneksi yang dibagikan dari AWS akun lain. Untuk informasi selengkapnya, lihat Berbagi koneksi dengan AWS akun.

Langkah 2: Berikan akses peran IAM CodeBuild proyek untuk menggunakan koneksi

Anda dapat memberikan akses peran IAM CodeBuild proyek untuk menggunakan GitHub token yang dijual oleh koneksi Anda.

Untuk memberikan CodeBuild akses peran IAM proyek

1. Buat peran IAM untuk CodeBuild proyek Anda dengan mengikuti petunjuk Memungkinkan CodeBuild untuk berinteraksi dengan orang lain AWS layanan untuk CodeBuild proyek Anda.
2. Saat mengikuti petunjuk, tambahkan kebijakan IAM berikut ke peran CodeBuild proyek Anda untuk memberikan akses ke koneksi.

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "codeconnections:GetConnectionToken",
                "codeconnections:GetConnection"
            ],
            "Resource": [
                "arn:aws:iam::*:role/Service*"
            ]
        }
    ]
}
```

Langkah 3: Konfigurasikan CodeBuild untuk menggunakan koneksi baru

Anda dapat mengonfigurasi koneksi sebagai kredensi tingkat akun dan menggunakannya dalam proyek.

Konsol Manajemen AWS

Untuk mengonfigurasi koneksi sebagai kredensi tingkat akun di Konsol Manajemen AWS

1. Untuk penyedia Sumber, pilih GitHub.
2. Untuk Credential, lakukan salah satu hal berikut:
   - Pilih Kredensi sumber default untuk menggunakan kredensi sumber default akun Anda untuk diterapkan ke semua proyek.
     a. Jika Anda tidak tersambung GitHub, pilih Kelola kredensi sumber default.
     b. Untuk jenis Kredenal, pilih GitHub Aplikasi.
     c. Di Koneksi, pilih untuk menggunakan koneksi yang ada atau buat koneksi baru.
   - Pilih Kredensi sumber kustom untuk menggunakan kredensi sumber kustom untuk mengganti setelan default akun Anda.
     a. Untuk jenis Kredenal, pilih GitHub Aplikasi.
     b. Di Koneksi, pilih untuk menggunakan koneksi yang ada atau buat koneksi baru.

AWS CLI

Untuk mengonfigurasi koneksi sebagai kredensi tingkat akun di AWS CLI

- Buka terminal (Linux, macOS, atau Unix) atau command prompt (Windows). Gunakan AWS CLI untuk menjalankan import-source-credentials perintah, menentukan, `--auth-type--server-type`, dan `--token` untuk koneksi Anda.

  Gunakan perintah berikut ini.

```
aws codebuild import-source-credentials --auth-type CODECONNECTIONS --server-type GITHUB --token <connection-arn>
```

Anda juga dapat mengatur beberapa token untuk CodeBuild proyek Anda. Untuk informasi selengkapnya, lihat Konfigurasikan beberapa token sebagai kredenal tingkat sumber.

Memecahkan masalah dengan Aplikasi GitHub

Informasi berikut dapat membantu Anda memecahkan masalah umum dengan Aplikasi. GitHub

Topik

- Instal AWS Konektor untuk GitHub aplikasi di wilayah yang tidak diinginkan
- Koneksi GitHub Aplikasi tidak memiliki akses ke repositori
- Bagian AWS Peran IAM layanan tidak memiliki izin IAM yang diperlukan.

Instal AWS Konektor untuk GitHub aplikasi di wilayah yang tidak diinginkan

Masalah: Anda menginstal AWS Konektor untuk GitHub dari GitHub Marketplace, tetapi koneksi dibuat di wilayah yang tidak diinginkan. Jika Anda mencoba mengonfigurasi ulang aplikasi di GitHub situs web, itu tidak akan berfungsi karena aplikasi sudah diinstal di GitHub akun Anda.

Kemungkinan penyebabnya: Aplikasi sudah diinstal di GitHub akun Anda, jadi Anda hanya dapat mengonfigurasi ulang izin aplikasi.

Solusi yang disarankan: Anda dapat membuat koneksi baru dengan ID instalasi di wilayah yang diinginkan.

1. Buka AWS CodeConnections konsol di https://console.aws.amazon.com/codesuite/settings/connectionsdan arahkan ke wilayah yang diinginkan menggunakan pemilih wilayah di bilah navigasi AWS konsol.
2. Ikuti petunjuk di Panduan Pengguna Alat Pengembang untuk Membuat koneksi ke GitHub.

> **Note**
>
> Karena Anda telah menginstal AWS Connector for GitHub app, Anda dapat memilihnya daripada menginstal aplikasi baru.

Koneksi GitHub Aplikasi tidak memiliki akses ke repositori

Masalah: AWS Layanan yang menggunakan koneksi, seperti CodeBuild atau CodePipeline, melaporkan bahwa layanan tersebut tidak memiliki akses ke repositori atau repositori tidak ada. Beberapa pesan kesalahan yang mungkin termasuk:

- `Authentication required for primary source.`

- `Unable to create webhook at this time. Please try again later.`
- `Failed to create webhook. GitHub API limit reached. Please try again later.`

Kemungkinan penyebabnya: Anda mungkin telah menggunakan GitHub aplikasi dan belum memberikan cakupan izin webhook.

Solusi yang disarankan: Untuk memberikan cakupan izin yang diperlukan, ikuti petunjuk dalam Menavigasi ke GitHub Aplikasi yang ingin Anda tinjau atau ubah untuk mengonfigurasi aplikasi yang diinstal. Di bawah bagian izin, Anda akan melihat aplikasi tidak memiliki izin webhooks, dan ada opsi bagi Anda untuk meninjau izin yang baru diminta. Tinjau dan terima izin baru. Untuk informasi selengkapnya, lihat Menyetujui izin yang diperbarui untuk Aplikasi. GitHub

Kemungkinan penyebabnya: Koneksi berfungsi seperti yang diharapkan, tetapi tiba-tiba tidak memiliki akses ke repositori.

Solusi yang memungkinkan: Mulailah dengan meninjau otorisasi dan instalasi Anda, lalu verifikasi GitHub Aplikasi diotorisasi dan diinstal. Jika penginstalan GitHub Aplikasi ditangguhkan, maka Anda harus membatalkan penangguhannya. Jika GitHub Aplikasi tidak diizinkan untuk koneksi UAT (User Access Token), atau tidak diinstal untuk koneksi IAT (Installation Access Token), koneksi yang ada tidak dapat digunakan lagi, dan Anda perlu membuat koneksi baru. Perhatikan bahwa menginstal ulang GitHub Aplikasi tidak akan menghidupkan kembali koneksi sebelumnya yang terkait dengan instalasi lama.

Solusi yang mungkin: Jika koneksi adalah koneksi UAT, pastikan koneksi tidak digunakan secara bersamaan, seperti digunakan dalam beberapa proses build CodeBuild secara bersamaan. Ini karena GitHub segera membatalkan UAT yang dikeluarkan sebelumnya jika token yang kedaluwarsa disegarkan oleh koneksi. Jika Anda perlu menggunakan koneksi UAT untuk beberapa CodeBuild build bersamaan, Anda dapat membuat beberapa koneksi dan menggunakan setiap koneksi secara independen.

Solusi yang mungkin: Jika koneksi UAT belum digunakan dalam 6 bulan terakhir, koneksi akan dibatalkan oleh. GitHub Untuk memperbaikinya, buat koneksi baru.

Kemungkinan penyebabnya: Anda mungkin telah menggunakan koneksi UAT tanpa menginstal aplikasi.

Solusi yang disarankan: Meskipun membuat koneksi UAT tidak memerlukan pengaitkan koneksi dengan instalasi GitHub Aplikasi, penginstalan diperlukan agar repositori dapat diakses. Ikuti petunjuk untuk meninjau instalasi untuk memastikan GitHub Aplikasi diinstal. Jika tidak diinstal,

navigasikan ke halaman GitHub Aplikasi untuk menginstal aplikasi. Untuk informasi selengkapnya tentang akses UAT, lihat Tentang token akses pengguna.

Bagian AWS Peran IAM layanan tidak memiliki izin IAM yang diperlukan.

Masalah: Anda melihat salah satu pesan galat berikut:

- `Access denied to connection <connection-arn>`
- `Failed to get access token from <connection-arn>`

Solusi yang disarankan: Biasanya Anda menggunakan koneksi dengan AWS layanan, seperti CodePipeline atau CodeBuild. Ketika Anda memberikan AWS layanan peran IAM, AWS layanan dapat menggunakan izin peran untuk bertindak atas nama Anda. Pastikan peran IAM memiliki izin yang diperlukan. Untuk informasi selengkapnya tentang izin IAM yang diperlukan, lihat Memberikan akses peran IAM CodeBuild proyek untuk menggunakan koneksi dan manajemen Identitas dan akses untuk AWS CodeStar Pemberitahuan dan AWS CodeConnections di Panduan Pengguna konsol Alat Pengembang.

GitHub dan token akses Server GitHub Perusahaan

Prasyarat token akses

Sebelum memulai, Anda harus menambahkan cakupan izin yang tepat ke token GitHub akses Anda.

Untuk GitHub, token akses pribadi Anda harus memiliki cakupan berikut.

- repo: Memberikan kontrol penuh atas repositori pribadi.
- repo:status: Memberikan read/write akses ke status komit repositori publik dan pribadi.
- admin:repo_hook: Memberikan kontrol penuh atas kait repositori. Cakupan ini tidak diperlukan jika token Anda memiliki `repo` ruang lingkup.
- admin:org_hook: Memberikan kontrol penuh atas kait organisasi. Cakupan ini hanya diperlukan jika Anda menggunakan fitur webhook organisasi.

Untuk informasi selengkapnya, lihat Memahami cakupan untuk aplikasi OAuth di situs web. GitHub

Jika Anda menggunakan token akses pribadi berbutir halus, tergantung pada kasus penggunaan Anda, token akses pribadi Anda mungkin memerlukan izin berikut:

- Isi: Read-only: Memberikan akses ke repositori pribadi. Izin ini diperlukan jika Anda menggunakan repositori pribadi sebagai sumber.
- Status komit: Baca dan tulis: Memberikan izin untuk membuat status komit. Izin ini diperlukan jika proyek Anda telah menyiapkan webhook, atau fitur status build laporan diaktifkan.
- Webhooks: Baca dan tulis: Memberikan izin untuk mengelola webhook. Izin ini diperlukan jika proyek Anda telah menyiapkan webhook.
- Permintaan tarik Read-only:: Memberikan izin untuk mengakses permintaan tarik. Izin ini diperlukan jika webhook Anda memiliki `FILE_PATH` filter pada peristiwa permintaan tarik.
- Administrasi: Baca dan tulis: Izin ini diperlukan jika Anda menggunakan fitur pelari GitHub Tindakan yang dihosting sendiri. CodeBuild Untuk detail selengkapnya, lihat Membuat token pendaftaran untuk repositori dan. Tutorial: Konfigurasikan pelari CodeBuild-hosted GitHub Tindakan

> **Note**
>
> Jika Anda ingin mengakses repositori organisasi, pastikan Anda menentukan organisasi sebagai pemilik sumber daya token akses.

Untuk informasi selengkapnya, lihat Izin yang diperlukan untuk token akses pribadi berbutir halus di situs web. GitHub

Connect GitHub dengan token akses (konsol)

Untuk menggunakan konsol untuk menghubungkan proyek Anda GitHub dengan menggunakan token akses, lakukan hal berikut saat Anda membuat proyek. Untuk informasi, lihat Buat proyek build (konsol).

1. Untuk penyedia Sumber, pilih GitHub.
2. Untuk Credential, lakukan salah satu hal berikut:
   - Pilih untuk menggunakan kredensil akun untuk menerapkan kredensi sumber default akun Anda ke semua proyek.
     a. Jika Anda tidak tersambung GitHub, pilih Kelola kredensi akun.
     b. Untuk tipe Credential, pilih Token akses pribadi.
   - Jika Anda memilih untuk menggunakan kredensi tingkat akun untuk Layanan, pilih layanan mana yang ingin Anda gunakan untuk menyimpan token Anda dan lakukan hal berikut:

a. Jika Anda memilih untuk menggunakan Secrets Manager, Anda dapat memilih untuk menggunakan koneksi rahasia yang ada atau membuat rahasia baru, lalu pilih Simpan. Untuk informasi selengkapnya tentang cara membuat rahasia baru, lihatBuat dan simpan token dalam rahasia Secrets Manager.

b. Jika Anda memilih untuk menggunakan CodeBuild, masukkan token akses GitHub pribadi Anda, lalu pilih Simpan.

- Pilih Use override credentials untuk project ini hanya untuk menggunakan kredensi sumber kustom untuk mengganti setelan kredensi akun Anda.

a. Dari daftar kredensi yang terisi, pilih salah satu opsi di bawah Token akses pribadi.

b. Anda juga dapat membuat token akses pribadi baru dengan memilih buat koneksi token akses pribadi baru dalam deskripsi.

Connect GitHub dengan token akses (CLI)

Ikuti langkah-langkah ini untuk menggunakan AWS CLI untuk menghubungkan proyek Anda GitHub dengan menggunakan token akses. Untuk informasi tentang menggunakan AWS CLI with AWS CodeBuild, lihatReferensi baris perintah.

1. Jalankan perintah import-source-credentials:

```
aws codebuild import-source-credentials --generate-cli-skeleton
```

JSON-formatted data muncul di output. Salin data ke file (misalnya,*import-source-credentials.json*) di lokasi di komputer lokal atau contoh di AWS CLI mana diinstal. Ubah data yang disalin sebagai berikut, dan simpan hasil Anda.

```
{
    "serverType": "server-type",
    "authType": "auth-type",
    "shouldOverwrite": "should-overwrite",
    "token": "token",
    "username": "username"
    }
```

Ganti yang berikut ini:

- *server-type*: Nilai yang dibutuhkan. Penyedia sumber yang digunakan untuk kredensi ini. Nilai yang valid adalah GITHUB, BITBUCKET, GITHUB_ENTERPRISE, GITLAB, dan GITLAB_SELF_MANAGED.
- *auth-type*: Nilai yang dibutuhkan. Jenis otentikasi yang digunakan untuk terhubung ke repositori. Nilai yang valid adalah OAUTH, BASIC_AUTH, PERSONAL_ACCESS_TOKEN, CODECONNECTIONS, dan SECRETS_MANAGER. Untuk GitHub, hanya PERSONAL_ACCESS_TOKEN yang diizinkan. BASIC_AUTH hanya diizinkan dengan kata sandi aplikasi Bitbucket atau token API.
- *should-overwrite*: Nilai opsional. Setel `false` untuk mencegah penimpaan kredenal sumber repositori. Setel `true` untuk menimpa kredensyal sumber repositori. Nilai default-nya adalah `true`.
- *token*: Nilai yang dibutuhkan. Untuk GitHub atau Server GitHub Perusahaan, ini adalah token akses pribadi. Untuk Bitbucket, ini adalah token akses pribadi, kata sandi aplikasi, atau token API. Untuk CODECONNECTIONS tipe auth-, ini adalah koneksi ARN. Untuk tipe auth-SECRETS_MANAGER, ini adalah ARN rahasia.
- *username*: Nilai opsional. Parameter ini diabaikan untuk GitHub dan penyedia sumber Server GitHub Perusahaan. Untuk kata sandi aplikasi Bitbucket, ini adalah nama pengguna Bitbucket Anda. Untuk token API Bitbucket, ini adalah alamat email yang terkait dengan akun Atlassian Anda.

2. Untuk menghubungkan akun Anda dengan token akses, beralih ke direktori yang berisi `import-source-credentials.json` file yang Anda simpan di langkah 1 dan jalankan import-source-credentials perintah lagi.

```
aws codebuild import-source-credentials --cli-input-json file://import-source-
credentials.json
```

JSON-formatted data muncul di output dengan Amazon Resource Name (ARN).

```
{
    "arn": "arn:aws:codebuild:region:account-id:token/server-type"
    }
```

> **Note**
>
> Jika Anda menjalankan import-source-credentials perintah dengan jenis server dan jenis autentikasi yang sama untuk kedua kalinya, token akses yang disimpan diperbarui.

Setelah akun Anda terhubung dengan token akses, Anda dapat menggunakannya `create-project` untuk membuat CodeBuild proyek Anda. Untuk informasi selengkapnya, lihat Buat proyek build (AWS CLI).

3. Untuk melihat token akses yang terhubung, jalankan list-source-credentials perintah.

```
aws codebuild list-source-credentials
```

Sebuah JSON-formatted `sourceCredentialsInfos` objek muncul di output:

```
{
        "sourceCredentialsInfos": [
            {
                "authType": "auth-type",
                "serverType": "server-type",
                "arn": "arn"
            }
        ]
    }
```

`sourceCredentialsObject`Berisi daftar informasi kredensial sumber yang terhubung:

- `authType`Ini adalah jenis otentikasi yang digunakan oleh kredensyal. Ini bisa`OAUTH`,`BASIC_AUTH`,`PERSONAL_ACCESS_TOKEN`,`CODECONNECTIONS`, atau`SECRETS_MANAGER`.
- `serverType`Ini adalah jenis penyedia sumber. Ini bisa`GITHUB`,`GITHUB_ENTERPRISE`,`BITBUCKET`,`GITLAB`, atau`GITLAB_SELF_MANAGED`.
- `arn`Itu adalah ARN dari token.

4. Untuk memutuskan sambungan dari penyedia sumber dan menghapus token aksesnya, jalankan delete-source-credentials perintah dengan ARN-nya.

```
aws codebuild delete-source-credentials --arn arn-of-your-credentials
```

JSON-formatted data dikembalikan dengan ARN dari kredensyal yang dihapus.

```
{
    "arn": "arn:aws:codebuild:region:account-id:token/server-type"
    }
```

GitHub Aplikasi OAuth

Connect GitHub menggunakan OAuth (konsol)

Untuk menggunakan konsol untuk menghubungkan proyek Anda GitHub menggunakan aplikasi OAuth, lakukan hal berikut saat Anda membuat proyek. Untuk informasi, lihat Buat proyek build (konsol).

1. Untuk penyedia Sumber, pilih GitHub.
2. Untuk Credential, lakukan salah satu hal berikut:
   - Pilih untuk menggunakan kredensil akun untuk menerapkan kredensi sumber default akun Anda ke semua proyek.
     a. Jika Anda tidak tersambung GitHub, pilih Kelola kredensi akun.
     b. Untuk jenis Kredensyal, pilih aplikasi OAuth.
   - Jika Anda memilih untuk menggunakan kredensi tingkat akun untuk Layanan, pilih layanan mana yang ingin Anda gunakan untuk menyimpan token Anda dan lakukan hal berikut:
     a. Jika Anda memilih untuk menggunakan Secrets Manager, Anda dapat memilih untuk menggunakan koneksi rahasia yang ada atau membuat rahasia baru, lalu pilih Simpan. Untuk informasi selengkapnya tentang cara membuat rahasia baru, lihatBuat dan simpan token dalam rahasia Secrets Manager.
     b. Jika Anda memilih untuk menggunakan CodeBuilddan kemudian pilih Simpan.
   - Pilih Use override credentials untuk project ini hanya untuk menggunakan kredensi sumber kustom untuk mengganti setelan kredensi akun Anda.
     a. Dari daftar kredensi yang terisi, pilih salah satu opsi di bawah aplikasi OAuth.

b. Anda juga dapat membuat token aplikasi OAuth baru dengan memilih buat koneksi token aplikasi Oauth baru dalam deskripsi.

Untuk meninjau aplikasi OAuth resmi Anda, navigasikan ke Aplikasi aktif GitHub, dan verifikasi bahwa aplikasi bernama yang `AWS CodeBuild (region)` dimiliki oleh aws-codesuite terdaftar.

Akses Bitbucket di CodeBuild

Untuk Bitbucket, Anda menggunakan token akses, kata sandi aplikasi, token API, aplikasi OAuth, atau koneksi Bitbucket untuk mengakses penyedia sumber.

Topik

- Koneksi Aplikasi Bitbucket
- Kata sandi aplikasi Bitbucket, token API, atau token akses
- Aplikasi Bitbucket OAuth

Koneksi Aplikasi Bitbucket

Anda dapat menggunakan Bitbucket untuk terhubung CodeBuild. Koneksi Bitbucket App didukung melalui AWS CodeConnections.

> **Note**
>
> AWS CodeConnections tersedia di wilayah yang lebih sedikit dari CodeBuild. Anda dapat menggunakan koneksi lintas wilayah di CodeBuild. Koneksi yang dibuat di Wilayah keikutsertaan tidak dapat digunakan di Wilayah lain. Untuk informasi lebih lanjut, lihat AWS CodeConnections kuota dan titik akhir.

Topik

- Langkah 1: Buat koneksi ke Bitbucket (konsol)
- Langkah 2: Berikan akses peran IAM CodeBuild proyek untuk menggunakan koneksi
- Langkah 3: Konfigurasikan CodeBuild untuk menggunakan koneksi baru
- Bitbucket OAuth memutar token penyegaran

Langkah 1: Buat koneksi ke Bitbucket (konsol)

Gunakan langkah-langkah ini untuk menggunakan CodeBuild konsol untuk menambahkan koneksi untuk proyek Anda di Bitbucket.

Untuk membuat koneksi ke Bitbucket

- Ikuti petunjuk di Panduan Pengguna Alat Pengembang untuk Membuat koneksi ke Bitbucket.

> **Note**
>
> Alih-alih membuat atau menggunakan koneksi yang ada di akun Anda, Anda dapat menggunakan koneksi yang dibagikan dari AWS akun lain. Untuk informasi selengkapnya, lihat Berbagi koneksi dengan AWS akun.

Langkah 2: Berikan akses peran IAM CodeBuild proyek untuk menggunakan koneksi

Anda dapat memberikan akses peran IAM CodeBuild proyek untuk menggunakan token Bitbucket yang dijual oleh koneksi Anda.

Untuk memberikan CodeBuild akses peran IAM proyek

1. Buat peran IAM untuk CodeBuild proyek Anda dengan mengikuti petunjuk Memungkinkan CodeBuild untuk berinteraksi dengan orang lain AWS layanan untuk CodeBuild proyek Anda.
2. Saat mengikuti petunjuk, tambahkan kebijakan IAM berikut ke peran CodeBuild proyek Anda untuk memberikan akses ke koneksi.

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "codeconnections:GetConnectionToken",
                "codeconnections:GetConnection"
            ],
            "Resource": [
```

```
                "arn:aws:iam::*:role/Service*"
            ]
        }
    ]
}
```

Langkah 3: Konfigurasikan CodeBuild untuk menggunakan koneksi baru

Anda dapat mengonfigurasi koneksi sebagai kredensi tingkat akun dan menggunakannya dalam proyek.

Konsol Manajemen AWS

Untuk mengonfigurasi koneksi sebagai kredensi tingkat akun di Konsol Manajemen AWS

1. Untuk penyedia Sumber, pilih Bitbucket.
2. Untuk Credential, lakukan salah satu hal berikut:
   - Pilih Kredensi sumber default untuk menggunakan kredensi sumber default akun Anda untuk diterapkan ke semua proyek.
     a. Jika Anda tidak terhubung ke Bitbucket, pilih Kelola kredensi sumber default.
     b. Untuk tipe Credential, pilih AWS CodeConnections.
     c. Di Koneksi, pilih untuk menggunakan koneksi yang ada atau buat koneksi baru.
   - Pilih Kredensi sumber kustom untuk menggunakan kredensi sumber kustom untuk mengganti setelan default akun Anda.
     a. Untuk tipe Credential, pilih AWS CodeConnections.
     b. Di Koneksi, pilih untuk menggunakan koneksi yang ada atau buat koneksi baru.

AWS CLI

Untuk mengonfigurasi koneksi sebagai kredensi tingkat akun di AWS CLI

- Buka terminal (Linux, macOS, atau Unix) atau command prompt (Windows). Gunakan AWS CLI untuk menjalankan import-source-credentials perintah, menentukan, `--auth-type--server-type`, dan `--token` untuk koneksi Anda.

  Gunakan perintah berikut ini.

```
aws codebuild import-source-credentials --auth-type CODECONNECTIONS --server-
type BITBUCKET --token <connection-arn>
```

Untuk informasi selengkapnya tentang menyiapkan beberapa token dalam CodeBuild proyek Anda, lihatKonfigurasikan beberapa token sebagai kredenal tingkat sumber.

Bitbucket OAuth memutar token penyegaran

Atlassian sekarang memberlakukan token penyegaran berputar sekali pakai untuk Bitbucket OAuth. Setiap kali Anda menggunakan token penyegaran, Bitbucket membatalkannya dan mengembalikan yang baru. Untuk informasi selengkapnya, lihat Bitbucket OAuth sekali pakai token refresh (CHANGE-3052) di situs web pengembang Atlassian.

Tindakan yang diperlukan untuk Manager-stored kredensil Rahasia

Jika Anda menyimpan kredensi Bitbucket OAuth di Secrets Manager, tambahkan `secretsmanager:PutSecretValue` izin ke peran layanan Anda. CodeBuild Dengan izin ini, CodeBuild dapat menulis token penyegaran yang diperbarui kembali ke rahasia Anda setelah setiap penyegaran token. Tanpa izin ini, build berikutnya setelah penyegaran token gagal.

Contoh berikut pernyataan kebijakan IAM memberikan izin yang diperlukan:

```
{
    "Effect": "Allow",
    "Action": "secretsmanager:PutSecretValue",
    "Resource": "arn:aws:secretsmanager:*:*:secret:codebuild/*"
}
```

Jika Anda menggunakan CodeBuild-managed opsi ini, tidak diperlukan perubahan. CodeBuild menangani rotasi token secara otomatis.

Migrasi ke AWS CodeConnections (Direkomendasikan)

Kami merekomendasikan migrasi ke AWS CodeConnections untuk integrasi Bitbucket. AWS CodeConnections menggunakan mekanisme otorisasi terpisah yang tidak memengaruhi token penyegaran yang berputar. Anda tidak perlu mengelola token atau menambahkan izin.

Untuk bermigrasi ke AWS CodeConnections, selesaikan langkah-langkah berikut:

1. Buka konsol Alat Pengembang. Pilih Pengaturan, Koneksi, lalu Buat koneksi. Pilih Bitbucket sebagai penyedia.
2. Otorisasi AWS CodeConnections aplikasi di ruang kerja Bitbucket Anda. Ini memerlukan izin Mengelola ruang kerja di Bitbucket.
3. Tambahkan izin berikut ke peran CodeBuild layanan Anda: `codeconnections:GetConnectionToken` dan`codeconnections:GetConnection`.
4. Setel koneksi sebagai kredensi default Anda:

```
aws codebuild import-source-credentials --auth-type CODECONNECTIONS --server-type
 BITBUCKET --token <connection-arn>
```

5. Semua proyek yang menggunakan sakelar kredenal default secara otomatis. Perbarui proyek yang menggunakan kredenal kustom per proyek secara individual. Gunakan konsol atau `UpdateProject` API.

Kata sandi aplikasi Bitbucket, token API, atau token akses

Kata sandi aplikasi Bitbucket tidak digunakan lagi

Atlassian telah menghentikan kata sandi aplikasi Bitbucket demi token API. Untuk terus menggunakan otentikasi dasar dengan CodeBuild, buat token API di pengaturan akun Atlassian Anda. Saat Anda menghubungkannya CodeBuild, gunakan alamat email yang terkait dengan akun Atlassian Anda sebagai nama pengguna. CodeBuild secara otomatis menggunakan kredensyal yang benar untuk operasi Git dan panggilan API Bitbucket. Untuk informasi selengkapnya, lihat Menggunakan token API di situs web Atlassian.

Rotasi token Bitbucket OAuth

Jika Anda menggunakan kredensil Bitbucket OAuth, kami sarankan untuk bermigrasi ke. Koneksi Aplikasi Bitbucket Untuk informasi tentang perubahan yang diperlukan saat Anda terus menggunakan OAuth dengan Secrets Manager, lihat. Aplikasi Bitbucket OAuth

Prasyarat

Sebelum memulai, Anda harus menambahkan cakupan izin yang tepat ke kata sandi aplikasi Bitbucket, token API, atau token akses Anda.

Untuk Bitbucket, kata sandi aplikasi, token API, atau token akses Anda harus memiliki cakupan berikut.

- repository:read: Memberikan akses baca ke semua repositori yang dapat diakses oleh pengguna otorisasi.
- pullrequest:read: Memberikan akses baca ke permintaan tarik. Jika project Anda memiliki webhook Bitbucket, kata sandi aplikasi atau token akses Anda harus memiliki cakupan ini.
- webhook: Memberikan akses ke webhook. Jika project Anda memiliki operasi webhook, kata sandi aplikasi atau token akses Anda harus memiliki cakupan ini.
- akun: Memberikan akses baca ke informasi akun pengguna.

Untuk informasi selengkapnya, lihat Cakupan untuk Bitbucket Cloud REST API dan OAuth di Bitbucket Cloud di situs web Bitbucket.

Hubungkan Bitbucket dengan kata sandi aplikasi atau token API (konsol)

Untuk menggunakan konsol untuk menghubungkan proyek Anda ke Bitbucket menggunakan kata sandi aplikasi atau token API, lakukan hal berikut saat Anda membuat proyek. Untuk informasi, lihat Buat proyek build (konsol).

> **Note**
>
> Jika Anda menggunakan token API, masukkan alamat email yang terkait dengan akun Atlassian Anda sebagai nama pengguna.

1. Untuk penyedia Sumber, pilih Bitbucket.
2. Untuk Credential, lakukan salah satu hal berikut:
   - Pilih untuk menggunakan kredensil akun untuk menerapkan kredensi sumber default akun Anda ke semua proyek.
     a. Jika Anda tidak terhubung ke Bitbucket, pilih Kelola kredensi akun.

b. Untuk jenis Credential, pilih token API (sebelumnya ditampilkan sebagai kata sandi Aplikasi).

- Jika Anda memilih untuk menggunakan kredensyal tingkat akun untuk Layanan, pilih layanan mana yang ingin Anda gunakan untuk menyimpan token Anda dan lakukan hal berikut:

  a. Jika Anda memilih untuk menggunakan Secrets Manager, Anda dapat memilih untuk menggunakan koneksi rahasia yang ada atau membuat rahasia baru, lalu pilih Simpan. Untuk informasi selengkapnya tentang cara membuat rahasia baru, lihatBuat dan simpan token dalam rahasia Secrets Manager.

  b. Jika Anda memilih untuk menggunakan CodeBuild, masukkan nama pengguna dan kata sandi aplikasi Bitbucket Anda, lalu pilih Simpan. Untuk token API Bitbucket, masukkan alamat email akun Atlassian Anda dan token API sebagai gantinya.

- Pilih Use override credentials untuk project ini hanya untuk menggunakan kredensi sumber kustom untuk mengganti setelan kredensi akun Anda.

  a. Dari daftar kredensi terisi, pilih salah satu opsi di bawah Kata sandi aplikasi atau token API.

  b. Anda juga dapat membuat kata sandi aplikasi baru atau koneksi token API dengan memilih tautan buat koneksi dalam deskripsi.

Connect Bitbucket dengan token akses (konsol)

Untuk menggunakan konsol untuk menghubungkan proyek Anda ke Bitbucket menggunakan token akses, lakukan hal berikut saat Anda membuat proyek. Untuk informasi, lihat Buat proyek build (konsol).

1. Untuk penyedia Sumber, pilih Bitbucket.
2. Untuk Credential, lakukan salah satu hal berikut:

   - Pilih untuk menggunakan kredensil akun untuk menerapkan kredensi sumber default akun Anda ke semua proyek.

     a. Jika Anda tidak terhubung ke Bitbucket, pilih Kelola kredensi akun.

     b. Untuk tipe Credential, pilih Token akses pribadi.

- Jika Anda memilih untuk menggunakan kredensyal tingkat akun untuk Layanan, pilih layanan mana yang ingin Anda gunakan untuk menyimpan token Anda dan lakukan hal berikut:

  a. Jika Anda memilih untuk menggunakan Secrets Manager, Anda dapat memilih untuk menggunakan koneksi rahasia yang ada atau membuat rahasia baru, lalu pilih Simpan. Untuk informasi selengkapnya tentang cara membuat rahasia baru, lihatBuat dan simpan token dalam rahasia Secrets Manager.

  b. Jika Anda memilih untuk menggunakan CodeBuild, masukkan token akses pribadi Bitbucket Anda, lalu pilih Simpan.

- Pilih Use override credentials untuk project ini hanya untuk menggunakan kredensi sumber kustom untuk mengganti setelan kredensi akun Anda.

  a. Dari daftar kredensi yang terisi, pilih salah satu opsi di bawah Token akses pribadi.

  b. Anda juga dapat membuat token akses pribadi baru dengan memilih buat koneksi token akses pribadi baru dalam deskripsi.

Hubungkan Bitbucket dengan kata sandi aplikasi atau token akses (CLI)

Ikuti langkah-langkah berikut untuk menggunakan AWS CLI untuk menghubungkan proyek Anda ke Bitbucket menggunakan kata sandi aplikasi atau token akses. Untuk informasi tentang menggunakan AWS CLI with AWS CodeBuild, lihatReferensi baris perintah.

1. Jalankan perintah import-source-credentials:

   ```
   aws codebuild import-source-credentials --generate-cli-skeleton
   ```

   JSON-formatted data muncul di output. Salin data ke file (misalnya,*import-source-credentials.json*) di lokasi di komputer lokal atau contoh di AWS CLI mana diinstal. Ubah data yang disalin sebagai berikut, dan simpan hasil Anda.

   ```
   {
       "serverType": "BITBUCKET",
       "authType": "auth-type",
       "shouldOverwrite": "should-overwrite",
       "token": "token",
       "username": "username"
       }
   ```

Ganti yang berikut ini:

- *server-type*: Nilai yang dibutuhkan. Penyedia sumber yang digunakan untuk kredensi ini. Nilai yang valid adalah GITHUB, BITBUCKET, GITHUB_ENTERPRISE, GITLAB, dan GITLAB_SELF_MANAGED.
- *auth-type*: Nilai yang dibutuhkan. Jenis otentikasi yang digunakan untuk terhubung ke repositori. Nilai yang valid adalah OAUTH, BASIC_AUTH, PERSONAL_ACCESS_TOKEN, CODECONNECTIONS, dan SECRETS_MANAGER. Untuk GitHub, hanya PERSONAL_ACCESS_TOKEN yang diizinkan. BASIC_AUTH hanya diizinkan dengan kata sandi aplikasi Bitbucket atau token API.
- *should-overwrite*: Nilai opsional. Setel `false` untuk mencegah penimpaan kredenal sumber repositori. Setel `true` untuk menimpa kredensyal sumber repositori. Nilai default-nya adalah `true`.
- *token*: Nilai yang dibutuhkan. Untuk GitHub atau Server GitHub Perusahaan, ini adalah token akses pribadi. Untuk Bitbucket, ini adalah token akses pribadi, kata sandi aplikasi, atau token API. Untuk CODECONNECTIONS tipe auth-, ini adalah koneksi ARN. Untuk tipe auth-SECRETS_MANAGER, ini adalah ARN rahasia.
- *username*: Nilai opsional. Parameter ini diabaikan untuk GitHub dan penyedia sumber Server GitHub Perusahaan. Untuk kata sandi aplikasi Bitbucket, ini adalah nama pengguna Bitbucket Anda. Untuk token API Bitbucket, ini adalah alamat email yang terkait dengan akun Atlassian Anda.

2. Untuk menghubungkan akun Anda dengan kata sandi aplikasi, token API, atau token akses, alihkan ke direktori yang berisi `import-source-credentials.json` file yang Anda simpan di langkah 1 dan jalankan import-source-credentials perintah lagi.

```
aws codebuild import-source-credentials --cli-input-json file://import-source-
credentials.json
```

JSON-formatted data muncul di output dengan Amazon Resource Name (ARN).

```
{
    "arn": "arn:aws:codebuild:region:account-id:token/server-type"
    }
```

> **Note**
>
> Jika Anda menjalankan import-source-credentials perintah dengan jenis server dan jenis autentikasi yang sama untuk kedua kalinya, token akses yang disimpan diperbarui.

Setelah akun Anda terhubung dengan kata sandi aplikasi atau token API, Anda dapat menggunakannya `create-project` untuk membuat CodeBuild proyek Anda. Untuk informasi selengkapnya, lihat Buat proyek build (AWS CLI).

3. Untuk melihat kata sandi aplikasi yang terhubung atau token akses, jalankan list-source-credentials perintah.

```
aws codebuild list-source-credentials
```

Sebuah JSON-formatted `sourceCredentialsInfos` objek muncul di output:

```
{
        "sourceCredentialsInfos": [
            {
                "authType": "auth-type",
                "serverType": "BITBUCKET",
                "arn": "arn"
            }
        ]
    }
```

`sourceCredentialsObject`Berisi daftar informasi kredensial sumber yang terhubung:

- `authType`Ini adalah jenis otentikasi yang digunakan oleh kredensyal. Ini bisa`OAUTH`,`BASIC_AUTH`,`PERSONAL_ACCESS_TOKEN`,`CODECONNECTIONS`, atau`SECRETS_MANAGER`.
- `serverType`Ini adalah jenis penyedia sumber. Ini bisa`GITHUB`,`GITHUB_ENTERPRISE`,`BITBUCKET`,`GITLAB`, atau`GITLAB_SELF_MANAGED`.
- `arn`Itu adalah ARN dari token.

4. Untuk memutuskan sambungan dari penyedia sumber dan menghapus kata sandi aplikasi atau token aksesnya, jalankan delete-source-credentials perintah dengan ARN-nya.

```
aws codebuild delete-source-credentials --arn arn-of-your-credentials
```

JSON-formatted data dikembalikan dengan ARN dari kredensyal yang dihapus.

```
{
    "arn": "arn:aws:codebuild:region:account-id:token/server-type"
    }
```

Aplikasi Bitbucket OAuth

⚠ Tindakan yang diperlukan untuk rotasi token Bitbucket OAuth

Atlassian sekarang memberlakukan token penyegaran berputar sekali pakai untuk Bitbucket OAuth. Kami merekomendasikan migrasi keKoneksi Aplikasi Bitbucket, yang menghilangkan manajemen token sepenuhnya. Untuk petunjuk tentang migrasi, lihatMigrasi ke AWS CodeConnections (Direkomendasikan).
Jika Anda lebih suka terus menggunakan OAuth dengan Secrets Manager, tambahkan `secretsmanager:PutSecretValue` izin ke peran CodeBuild layanan Anda. Untuk informasi selengkapnya, lihat Tindakan yang diperlukan untuk Manager-stored kredensil Rahasia.
Jika Anda menggunakan CodeBuild-managed opsi ini, tidak diperlukan perubahan. CodeBuild menangani rotasi token secara otomatis.

Connect Bitbucket menggunakan OAuth (konsol)

Untuk menggunakan konsol untuk menghubungkan proyek Anda ke Bitbucket menggunakan aplikasi OAuth, lakukan hal berikut saat Anda membuat proyek. Untuk informasi, lihat Buat proyek build (konsol).

1. Untuk penyedia Sumber, pilih Bitbucket.
2. Untuk Credential, lakukan salah satu hal berikut:
   - Pilih untuk menggunakan kredensil akun untuk menerapkan kredensi sumber default akun Anda ke semua proyek.

     a. Jika Anda tidak terhubung ke Bitbucket, pilih Kelola kredensi akun.

b. Untuk jenis Kredensyal, pilih aplikasi OAuth.

- Jika Anda memilih untuk menggunakan kredensyal tingkat akun untuk Layanan, pilih layanan mana yang ingin Anda gunakan untuk menyimpan token Anda dan lakukan hal berikut:

  a. Jika Anda memilih untuk menggunakan Secrets Manager, Anda dapat memilih untuk menggunakan koneksi rahasia yang ada atau membuat rahasia baru, lalu pilih Simpan. Untuk informasi selengkapnya tentang membuat rahasia baru, lihatBuat dan simpan token dalam rahasia Secrets Manager.

  b. Jika Anda memilih untuk menggunakan CodeBuilddan kemudian pilih Simpan.

- Pilih Use override credentials untuk project ini hanya untuk menggunakan kredensi sumber kustom untuk mengganti setelan kredensi akun Anda.

  a. Dari daftar kredensi yang terisi, pilih salah satu opsi di bawah aplikasi OAuth.

  b. Anda juga dapat membuat token aplikasi OAuth baru dengan memilih buat koneksi token aplikasi Oauth baru dalam deskripsi.

Untuk meninjau aplikasi OAuth resmi Anda, navigasikan ke Otorisasi aplikasi di Bitbucket, dan verifikasi bahwa aplikasi bernama terdaftar. `AWS CodeBuild (region)`

GitLab akses di CodeBuild

Untuk GitLab, Anda menggunakan GitLab koneksi untuk mengakses penyedia sumber.

Topik

- Connect CodeBuild ke GitLab

Connect CodeBuild ke GitLab

Koneksi memungkinkan Anda untuk mengotorisasi dan membuat konfigurasi yang mengaitkan penyedia pihak ketiga Anda dengan AWS sumber daya Anda menggunakan. AWS CodeConnections Untuk mengaitkan repositori pihak ketiga Anda sebagai sumber untuk proyek build Anda, Anda menggunakan koneksi.

Untuk menambahkan GitLab atau penyedia sumber GitLab Mandiri yang Dikelola CodeBuild, Anda dapat memilih salah satu untuk:

- Gunakan wizard Buat proyek build CodeBuild konsol atau halaman Edit Sumber untuk memilih opsi GitLabatau GitLab Self Managed provider. Lihat Buat koneksi ke GitLab (konsol) untuk menambahkan penyedia sumber. Konsol membantu Anda membuat sumber daya koneksi.
- Gunakan CLI untuk membuat sumber daya koneksi Anda, lihat Buat koneksi ke GitLab (CLI) untuk membuat sumber daya koneksi dengan CLI.

> **Note**
>
> Anda juga dapat membuat koneksi menggunakan konsol Alat Pengembang di bawah Pengaturan. Lihat Membuat Koneksi.

> **Note**
>
> Dengan mengotorisasi instalasi koneksi ini di GitLab, Anda memberikan izin layanan kami untuk memproses data Anda dengan mengakses akun Anda, dan Anda dapat mencabut izin kapan saja dengan menghapus instalasi aplikasi.

Buat koneksi ke GitLab

Bagian ini menjelaskan cara terhubung GitLab ke CodeBuild. Untuk informasi selengkapnya tentang koneksi GitLab , lihat Connect CodeBuild ke GitLab.

Sebelum Anda memulai:

- Anda harus sudah membuat akun dengan GitLab.

> **Note**
>
> Koneksi hanya menyediakan akses ke repositori yang dimiliki oleh akun yang digunakan untuk membuat dan mengotorisasi koneksi.

> **Note**
>
> Anda dapat membuat koneksi ke repositori tempat Anda memiliki peran Pemilik GitLab, dan kemudian koneksi dapat digunakan dengan repositori dengan sumber daya seperti. CodeBuild Untuk repositori dalam grup, Anda tidak perlu menjadi pemilik grup.

- Untuk menentukan sumber untuk proyek build Anda, Anda harus sudah membuat repositori. GitLab

Topik

- Buat koneksi ke GitLab (konsol)
- Buat koneksi ke GitLab (CLI)

Buat koneksi ke GitLab (konsol)

Gunakan langkah-langkah ini untuk menggunakan CodeBuild konsol untuk menambahkan koneksi untuk proyek Anda (repositori) di. GitLab

> **Note**
>
> Alih-alih membuat atau menggunakan koneksi yang ada di akun Anda, Anda dapat menggunakan koneksi yang dibagikan dari AWS akun lain. Untuk informasi selengkapnya, lihat Berbagi koneksi dengan AWS akun.

Untuk membuat atau mengedit proyek build

1. Masuk ke CodeBuild konsol.
2. Pilih salah satu dari berikut ini.
   - Pilih untuk membuat proyek build. Ikuti langkah-langkah Buat proyek build (konsol) untuk menyelesaikan layar pertama dan di bagian Sumber, di bawah Penyedia Sumber, pilih GitLab.
   - Pilih untuk mengedit proyek build yang sudah ada. Pilih Edit, lalu pilih Sumber. Di halaman Edit Sumber, di bawah Penyedia sumber, pilih GitLab.
3. Pilih salah satu cara berikut:

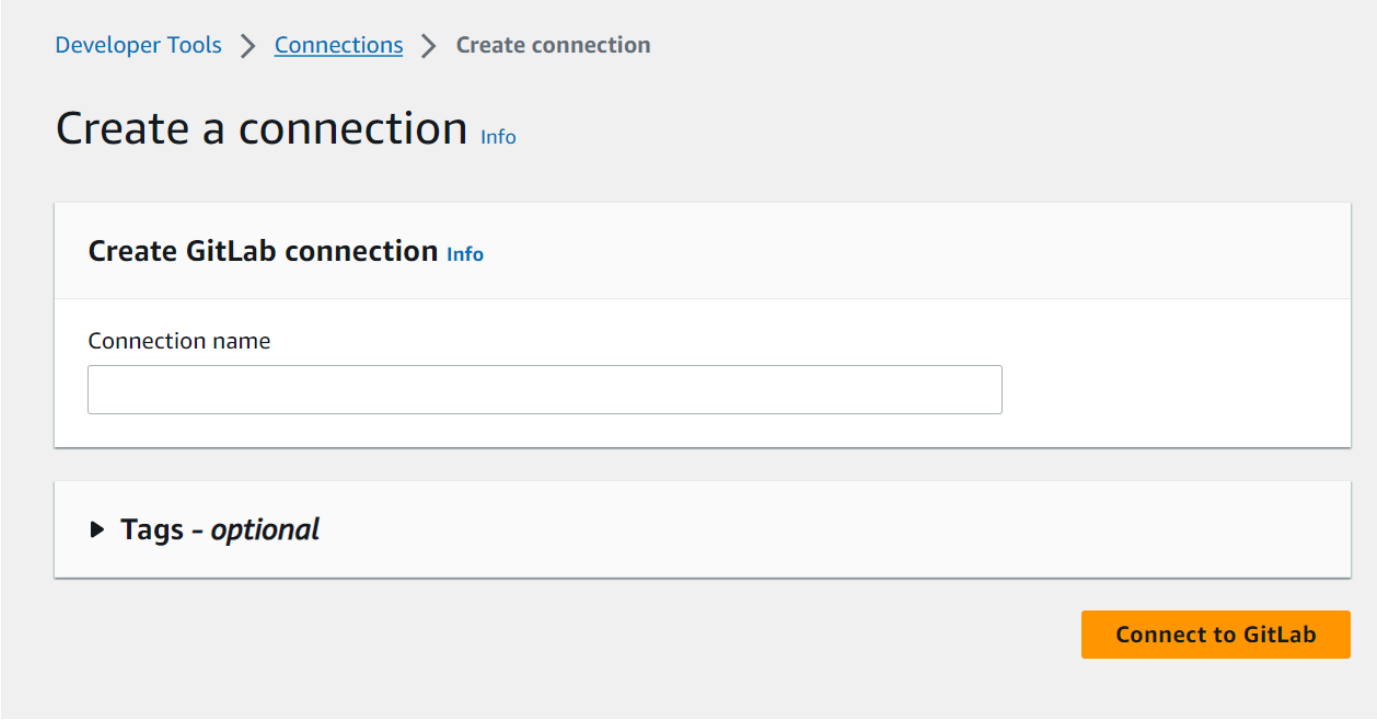
- Di bawah Koneksi, pilih Koneksi default. Koneksi default menerapkan GitLab koneksi default di semua proyek.
- Di bawah Koneksi, pilih Koneksi khusus. Koneksi kustom menerapkan GitLab koneksi kustom yang mengesampingkan pengaturan default akun Anda.

4. Lakukan salah satu tindakan berikut:

- Di bawah Koneksi default atau Koneksi khusus, jika Anda belum membuat sambungan ke penyedia Anda, pilih Buat GitLab koneksi baru. Lanjutkan ke langkah 5 untuk membuat koneksi.
- Di bawah Koneksi, jika Anda telah membuat koneksi ke penyedia Anda, pilih koneksi. Lanjutkan ke langkah 10.

> **Note**
>
> Jika Anda menutup jendela pop-up sebelum GitLab koneksi dibuat, Anda perlu menyegarkan halaman.

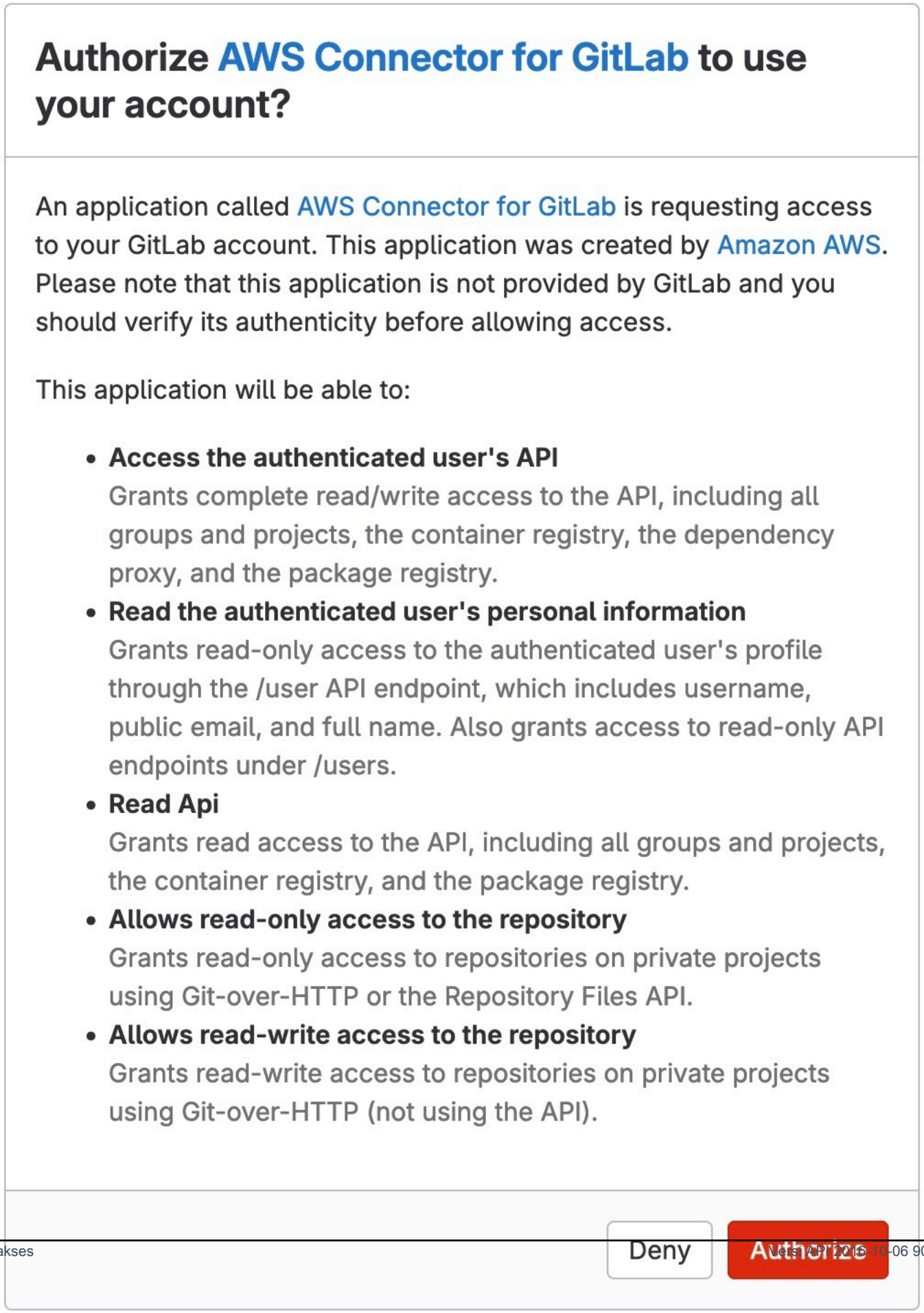
5. Untuk membuat sambungan ke GitLab repositori, di bawah Pilih penyedia, pilih. GitLab Di Nama koneksi, masukkan nama untuk koneksi yang ingin Anda buat. Pilih Connect to GitLab.



6. Saat halaman login untuk GitLab ditampilkan, masuk dengan kredensialnya, lalu pilih Masuk.

7. Jika ini adalah pertama kalinya Anda mengotorisasi koneksi, halaman otorisasi ditampilkan dengan pesan yang meminta otorisasi untuk koneksi untuk mengakses akun Anda. GitLab

   Pilih Izinkan.


Authorize AWS Connector for GitLab to use your account?
An application called AWS Connector for GitLab is requesting access to your GitLab account. This application was created by Amazon AWS. Please note that this application is not provided by GitLab and you should verify its authenticity before allowing access.
This application will be able to:
Access the authenticated user's API
Grants complete read/write access to the API, including all groups and projects, the container registry, the dependency proxy, and the package registry.
Read the authenticated user's personal information
Grants read-only access to the authenticated user's profile through the /user API endpoint, which includes username, public email, and full name. Also grants access to read-only API endpoints under /users.
Read Api
Grants read access to the API, including all groups and projects, the container registry, and the package registry.
Allows read-only access to the repository
Grants read-only access to repositories on private projects using Git-over-HTTP or the Repository Files API.
Allows read-write access to the repository
Grants read-write access to repositories on private projects using Git-over-HTTP (not using the API).
Deny
Authorize

8. Browser kembali ke halaman konsol koneksi. Di bawah pengaturan GitLab koneksi, koneksi baru ditampilkan di Nama koneksi.
9. Pilih Hubungkan.

   Setelah GitLab koneksi berhasil dibuat, spanduk sukses akan ditampilkan di bagian atas.
10. Pada halaman Buat proyek build, di daftar drop-down Koneksi default atau Koneksi kustom, pastikan ARN koneksi Anda terdaftar. Jika tidak, pilih tombol refresh agar muncul.
11. Di Repository, pilih nama proyek Anda GitLab dengan menentukan jalur proyek dengan namespace. Misalnya, untuk repositori tingkat grup, masukkan nama repositori dalam format berikut:. `group-name/repository-name` Untuk informasi selengkapnya tentang path dan namespace, lihat `path_with_namespace` bidang di https://docs.gitlab.com/ee/api/projects.html#get -single-project. Untuk informasi selengkapnya tentang namespace di GitLab, lihat. https://docs.gitlab.com/ee/user/namespace/

    > **Note**
    >
    > Untuk grup di GitLab, Anda harus secara manual menentukan jalur proyek dengan namespace. Misalnya, untuk repositori bernama `myrepo` dalam grup`mygroup`, masukkan yang berikut ini:. `mygroup/myrepo` Anda dapat menemukan jalur proyek dengan namespace di URL di. GitLab

12. Dalam versi Sumber - opsional, masukkan ID permintaan tarik, cabang, ID komit, tag, atau referensi, dan ID komit. Untuk informasi selengkapnya, lihat Contoh versi sumber dengan AWS CodeBuild.

    > **Note**
    >
    > Kami menyarankan Anda memilih nama cabang Git yang tidak terlihat seperti ID komit, seperti `811dd1ba1aba14473856cee38308caed7190c0d` atau`5392f7`. Ini membantu Anda menghindari tabrakan checkout Git dengan commit yang sebenarnya.

13. Di kedalaman klon Git - opsional, Anda dapat membuat klon dangkal dengan riwayat terpotong ke jumlah komit yang ditentukan. Jika Anda ingin klon penuh, pilih Penuh.

14. Di Status Build - opsional, pilih Laporkan status build ke penyedia sumber saat build dimulai dan selesai jika ingin status awal dan penyelesaian build dilaporkan ke penyedia sumber.

    Untuk dapat melaporkan status build ke penyedia sumber, pengguna yang terkait dengan penyedia sumber harus memiliki akses tulis ke repo. Jika pengguna tidak memiliki akses tulis, status build tidak dapat diperbarui. Untuk informasi selengkapnya, lihat Akses penyedia sumber.

Buat koneksi ke GitLab (CLI)

Anda dapat menggunakan AWS Command Line Interface (AWS CLI) untuk membuat koneksi.

Untuk melakukannya, gunakan perintah create-connection.

> **Important**
>
> Koneksi yang dibuat melalui AWS CLI atau AWS CloudFormation dalam `PENDING` status secara default. Setelah Anda membuat koneksi dengan CLI atau CloudFormation, gunakan konsol untuk mengedit koneksi untuk membuat statusnya. `AVAILABLE`

Untuk membuat koneksi

- Ikuti petunjuk di Panduan Pengguna konsol Alat Pengembang untuk Membuat koneksi ke GitLab (CLI).

Cross-service pencegahan wakil bingung

Masalah "confused deputy" adalah masalah keamanan di mana entitas yang tidak memiliki izin untuk melakukan tindakan dapat memengaruhi entitas yang memiliki hak akses lebih tinggi untuk melakukan tindakan. Pada tahun AWS, peniruan lintas layanan dapat mengakibatkan masalah wakil yang membingungkan. Cross-service peniruan identitas dapat terjadi ketika satu layanan (layanan panggilan) memanggil layanan lain (layanan yang disebut). Layanan pemanggilan dapat dimanipulasi menggunakan izinnya untuk bertindak pada sumber daya pelanggan lain dengan cara yang seharusnya tidak dilakukannya kecuali bila memiliki izin untuk mengakses. Untuk mencegah hal ini, AWS menyediakan alat yang membantu Anda melindungi data untuk semua layanan dengan principal layanan yang telah diberi akses ke sumber daya di akun Anda.

Sebaiknya gunakan kunci konteks kondisi `aws:SourceAccount`global `aws:SourceArn`dan dalam kebijakan sumber daya untuk membatasi izin yang AWS CodeBuild memberikan layanan lain ke sumber daya. Gunakan `aws:SourceArn` jika Anda ingin hanya satu sumber daya yang akan dikaitkan dengan akses lintas layanan. Gunakan `aws:SourceAccount` jika Anda ingin mengizinkan sumber daya apa pun di akun tersebut dikaitkan dengan penggunaan lintas layanan.

Cara paling efektif untuk melindungi dari masalah "confused deputy" adalah dengan menggunakan kunci konteks kondisi global `aws:SourceArn` dengan ARN lengkap sumber daya. Jika Anda tidak mengetahui ARN lengkap sumber daya atau jika Anda menentukan beberapa sumber daya, gunakan kunci kondisi konteks global `aws:SourceArn` dengan karakter wildcard (*) untuk bagian ARN yang tidak diketahui. Misalnya, `arn:aws:codebuild:*:`*`123456789012`*`:*`.

Jika nilai `aws:SourceArn` tidak berisi ID akun, seperti ARN bucket Amazon S3, Anda harus menggunakan kedua kunci konteks kondisi global tersebut untuk membatasi izin.

Nilai `aws:SourceArn` harus menjadi CodeBuild proyek ARN.

Contoh berikut menunjukkan bagaimana Anda dapat menggunakan kunci konteks kondisi `aws:SourceAccount` global `aws:SourceArn` dan CodeBuild untuk mencegah masalah wakil yang membingungkan.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "codebuild.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceArn": "arn:aws:codebuild:us-
east-1:111122223333:project/MyProject"
                }
            }
        }
    ]
}
```

Topik lanjutan

Bagian ini mencakup beberapa topik lanjutan yang berguna bagi AWS CodeBuild pengguna yang lebih berpengalaman.

Topik

- Izinkan pengguna berinteraksi dengan CodeBuild
- Memungkinkan CodeBuild untuk berinteraksi dengan orang lain AWS layanan
- Enkripsi output build menggunakan kunci yang dikelola pelanggan
- Berinteraksi dengan CodeBuild menggunakan AWS CLI
- Referensi baris perintah untuk AWS CodeBuild
- AWS Referensi SDK dan alat untuk AWS CodeBuild
- Menggunakan layanan ini dengan AWS SDK
- Tentukan AWS CodeBuild titik akhir
- Gunakan AWS CodeBuild dengan AWS CodePipeline untuk menguji kode dan menjalankan build
- Gunakan AWS CodeBuild dengan Codecov
- Gunakan AWS CodeBuild dengan Jenkins
- Gunakan AWS CodeBuild dengan aplikasi tanpa server
- Pemberitahuan pihak ketiga untuk AWS CodeBuild untuk Windows
- Gunakan kunci CodeBuild kondisi sebagai variabel peran layanan IAM untuk mengontrol akses build
- AWS CodeBuild kunci kondisi

Izinkan pengguna berinteraksi dengan CodeBuild

Jika Anda mengikuti langkah-langkah Memulai menggunakan konsol untuk mengakses AWS CodeBuild untuk pertama kalinya, kemungkinan besar Anda tidak memerlukan informasi dalam topik ini. Namun, saat Anda terus menggunakan CodeBuild, Anda mungkin ingin melakukan hal-hal seperti memberi pengguna dan grup lain dalam organisasi Anda kemampuan untuk berinteraksi dengannya CodeBuild.

Untuk mengizinkan pengguna atau grup IAM berinteraksi dengan mereka AWS CodeBuild, Anda harus memberi mereka izin akses ke CodeBuild. Bagian ini menjelaskan cara melakukan ini dengan konsol IAM atau. AWS CLI

Jika Anda akan mengakses CodeBuild dengan akun AWS root Anda (tidak disarankan) atau pengguna administrator di AWS akun Anda, maka Anda tidak perlu mengikuti instruksi ini.

Untuk informasi tentang akun AWS root dan pengguna administrator, lihat Akun AWS Pengguna root dan Membuat pengguna dan Grup Akun AWS root pertama Anda di Panduan pengguna.

Untuk menambahkan izin CodeBuild akses ke grup atau pengguna IAM (konsol)

1. Buka konsol IAM di https://console.aws.amazon.com/iam/.

   Anda seharusnya sudah masuk Konsol Manajemen AWS dengan menggunakan salah satu dari berikut ini:

   - Akun AWS root Anda. Ini tidak disarankan. Untuk informasi selengkapnya, lihat Pengguna Akun AWS root di Panduan pengguna.
   - Pengguna administrator di AWS akun Anda. Untuk informasi selengkapnya, lihat Membuat pengguna dan Grup Akun AWS root pertama Anda di Panduan pengguna.
   - Pengguna di AWS akun Anda dengan izin untuk melakukan serangkaian tindakan minimum berikut:

     ```
     iam:AttachGroupPolicy
     iam:AttachUserPolicy
     iam:CreatePolicy
     iam:ListAttachedGroupPolicies
     iam:ListAttachedUserPolicies
     iam:ListGroups
     iam:ListPolicies
     iam:ListUsers
     ```

     Untuk informasi selengkapnya, lihat Ikh tisar Kebijakan IAM di Panduan pengguna.
2. Di panel navigasi, pilih Kebijakan.
3. Untuk menambahkan kumpulan izin AWS CodeBuild akses khusus ke grup IAM atau pengguna IAM, lanjutkan ke langkah 4 dalam prosedur ini.

Untuk menambahkan kumpulan izin CodeBuild akses default ke grup IAM atau pengguna IAM, pilih Jenis Kebijakan, Dikel AWS ola , lalu lakukan hal berikut:

- Untuk menambahkan izin akses penuh ke CodeBuild, pilih kotak bernama AWSCodeBuildAdminAccess, pilih Tind akan Kebijakan, lalu pilih Lamp irkan. Pilih kotak di sebelah grup atau pengguna IAM target, lalu pilih Lamp irkan Kebijakan. Ulangi ini untuk kebijakan bernama AmazonS3ReadOnlyAccess dan IAMFullAccess.
- Untuk menambahkan izin akses ke semua CodeBuild hal kecuali administrasi proyek build, pilih kotak bernama AWSCodeBuildDeveloperAccess, pilih T indakan Kebijakan, lalu pilih Lamp irkan. Pilih kotak di sebelah grup atau pengguna IAM target, lalu pilih Lamp irkan Kebijakan. Ulangi ini untuk kebijakan yang disebutkan AmazonS3ReadOnlyAccess.
- Untuk menambahkan izin akses baca-saja CodeBuild, pilih kotak bernama. AWSCodeBuildReadOnlyAccess Pilih kotak di sebelah grup atau pengguna IAM target, lalu pilih Lamp irkan Kebijakan. Ulangi ini untuk kebijakan yang disebutkan AmazonS3ReadOnlyAccess.

Anda sekarang telah menambahkan kumpulan izin CodeBuild akses default ke grup atau pengguna IAM. Lewati sisa langkah dalam prosedur ini.

4. Pilih Buat Kebijakan.
5. Pada halaman Buat Kebijakan, di samping Buat Kebijakan Sendiri, pilih Pilih.
6. Pada halaman Kebijakan Tinjauan, untuk Nama Kebijakan, masukkan nama kebijakan (misalnya,**`CodeBuildAccessPolicy`**). Jika Anda menggunakan nama yang berbeda, pastikan untuk menggunakannya selama prosedur ini.
7. Untuk Dokumen Kebijakan, masukkan yang berikut ini, lalu pilih Buat Kebijakan.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "CodeBuildAccessPolicy",
            "Effect": "Allow",
            "Action": [
                "codebuild:*"
            ],
```

```
            "Resource": "*"
        },
        {
            "Sid": "CodeBuildRolePolicy",
            "Effect": "Allow",
            "Action": [
                "iam:PassRole"
            ],
            "Resource": "arn:aws:iam::111122223333:role/role-name"
        },
        {
            "Sid": "CloudWatchLogsAccessPolicy",
            "Effect": "Allow",
            "Action": [
                "logs:FilterLogEvents",
                "logs:GetLogEvents"
            ],
            "Resource": "*"
        },
        {
            "Sid": "S3AccessPolicy",
            "Effect": "Allow",
            "Action": [
                "s3:CreateBucket",
                "s3:GetObject",
                "s3:List*",
                "s3:PutObject"
            ],
            "Resource": "*"
        },
        {
            "Sid": "S3BucketIdentity",
            "Effect": "Allow",
            "Action": [
                "s3:GetBucketAcl",
                "s3:GetBucketLocation"
            ],
            "Resource": "*"
        }
    ]
}
```

> **Note**
>
> Kebijakan ini memungkinkan akses ke semua CodeBuild tindakan dan AWS sumber daya yang berpotensi besar. Untuk membatasi izin untuk CodeBuild tindakan tertentu, ubah nilai `codebuild:*` dalam pernyataan CodeBuild kebijakan. Untuk informasi selengkapnya, lihat Manajemen Identitas dan Akses. Untuk membatasi akses ke AWS sumber daya tertentu, ubah nilai `Resource` objek. Untuk informasi selengkapnya, lihat Manajemen Identitas dan Akses.

8. Di panel navigasi, pilih Grup atau Pengguna.
9. Dalam daftar grup atau pengguna, pilih nama grup IAM atau pengguna IAM yang ingin Anda tambahkan izin CodeBuild akses.
10. Untuk grup, pada halaman pengaturan grup, pada tab Izin, perluas Kebijakan Ter kelola, lalu pilih Lamp irkan Kebijakan.

    Untuk pengguna, pada halaman pengaturan pengguna, pada tab Izin, pilih Tam bah izin.
11. Untuk grup, pada halaman Lamp irkan Kebijakan, pilih CodeBuildAccessPolicy, lalu pilih Lamp irkan Kebijakan.

    Untuk pengguna, pada halaman Tam bah izin, pilih Lamp irkan kebijakan yang ada secara langsung. Pilih CodeBuildAccessPolicy, pilih Berikutnya: Review, lalu pilih Tambahkan izin.

Untuk menambahkan izin CodeBuild akses ke grup atau pengguna IAM (AWS CLI)

1. Pastikan Anda telah mengonfigurasi AWS CLI dengan kunci AWS akses dan kunci akses AWS rahasia yang sesuai dengan salah satu entitas IAM, seperti yang dijelaskan dalam prosedur sebelumnya. Untuk informasi selengkapnya, lihat Menyiapkan dengan AWS Command Line Interface di Panduan AWS Command Line Interface Pengguna.
2. Untuk menambahkan kumpulan izin AWS CodeBuild akses khusus ke grup IAM atau pengguna IAM, lewati ke langkah 3 dalam prosedur ini.

   Untuk menambahkan kumpulan izin CodeBuild akses default ke grup IAM atau pengguna IAM, lakukan hal berikut:

   Jalankan salah satu perintah berikut, tergantung apakah Anda ingin menambahkan izin ke grup atau pengguna IAM:

```
aws iam attach-group-policy --group-name group-name --policy-arn policy-arn

aws iam attach-user-policy --user-name user-name --policy-arn policy-arn
```

Anda harus menjalankan perintah tiga kali, mengganti *group-name* atau *user-name* dengan nama grup IAM atau nama pengguna, dan mengganti satu *policy-arn* kali untuk setiap kebijakan Amazon Resource Names (ARN) berikut:

- Untuk menambahkan izin akses penuh ke CodeBuild, gunakan ARN kebijakan berikut:
  - `arn:aws:iam::aws:policy/AWSCodeBuildAdminAccess`
  - `arn:aws:iam::aws:policy/AmazonS3ReadOnlyAccess`
  - `arn:aws:iam::aws:policy/IAMFullAccess`
- Untuk menambahkan izin akses ke CodeBuild semua hal kecuali administrasi proyek build, gunakan ARN kebijakan berikut:
  - `arn:aws:iam::aws:policy/AWSCodeBuildDeveloperAccess`
  - `arn:aws:iam::aws:policy/AmazonS3ReadOnlyAccess`
- Untuk menambahkan izin akses baca-saja ke CodeBuild, gunakan ARN kebijakan berikut:
  - `arn:aws:iam::aws:policy/AWSCodeBuildReadOnlyAccess`
  - `arn:aws:iam::aws:policy/AmazonS3ReadOnlyAccess`

Anda sekarang telah menambahkan kumpulan izin CodeBuild akses default ke grup atau pengguna IAM. Lewati sisa langkah dalam prosedur ini.

3. Di direktori kosong di workstation lokal atau instance tempat AWS CLI diinstal, buat file bernama `put-group-policy.json` atau`put-user-policy.json`. Jika Anda menggunakan nama file yang berbeda, pastikan untuk menggunakannya selama prosedur ini.

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "CodeBuildAccessPolicy",
            "Effect": "Allow",
            "Action": [
```

```
                "codebuild:*"
            ],
            "Resource": "*"
        },
        {
            "Sid": "CodeBuildRolePolicy",
            "Effect": "Allow",
            "Action": [
                "iam:PassRole"
            ],
            "Resource": "arn:aws:iam::111122223333:role/role-name"
        },
        {
            "Sid": "CloudWatchLogsAccessPolicy",
            "Effect": "Allow",
            "Action": [
                "logs:FilterLogEvents",
                "logs:GetLogEvents"
            ],
            "Resource": "*"
        },
        {
            "Sid": "S3AccessPolicy",
            "Effect": "Allow",
            "Action": [
                "s3:CreateBucket",
                "s3:GetObject",
                "s3:List*",
                "s3:PutObject"
            ],
            "Resource": "*"
        },
        {
            "Sid": "S3BucketIdentity",
            "Effect": "Allow",
            "Action": [
                "s3:GetBucketAcl",
                "s3:GetBucketLocation"
            ],
            "Resource": "*"
        }
    ]
}
```

> **Note**
>
> Kebijakan ini memungkinkan akses ke semua CodeBuild tindakan dan AWS sumber daya yang berpotensi besar. Untuk membatasi izin untuk CodeBuild tindakan tertentu, ubah nilai `codebuild:*` dalam pernyataan CodeBuild kebijakan. Untuk informasi selengkapnya, lihat Manajemen Identitas dan Akses. Untuk membatasi akses ke AWS sumber daya tertentu, ubah nilai `Resource` objek terkait. Untuk informasi selengkapnya, lihat Manajemen Identitas dan Akses atau dokumentasi keamanan AWS layanan tertentu.

4. Beralih ke direktori tempat Anda menyimpan file, lalu jalankan salah satu perintah berikut. Anda dapat menggunakan nilai yang berbeda untuk `CodeBuildGroupAccessPolicy` dan`CodeBuildUserAccessPolicy`. Jika Anda menggunakan nilai yang berbeda, pastikan untuk menggunakannya di sini.

   Untuk grup IAM:

   ```
   aws iam put-group-policy --group-name group-name --policy-name
    CodeBuildGroupAccessPolicy --policy-document file://put-group-policy.json
   ```

   Untuk pengguna:

   ```
   aws iam put-user-policy --user-name user-name --policy-name
    CodeBuildUserAccessPolicy --policy-document file://put-user-policy.json
   ```

   Dalam perintah sebelumnya, ganti *`group-name`* atau *`user-name`* dengan nama grup IAM target atau pengguna.

Memungkinkan CodeBuild untuk berinteraksi dengan orang lain AWS layanan

Jika Anda mengikuti langkah-langkah Memulai menggunakan konsol untuk mengakses AWS CodeBuild untuk pertama kalinya, kemungkinan besar Anda tidak memerlukan informasi dalam topik ini. Namun, saat Anda terus menggunakan CodeBuild, Anda mungkin ingin melakukan hal-hal seperti memungkinkan CodeBuild untuk berinteraksi dengan AWS layanan lain.

CodeBuild Untuk memungkinkan berinteraksi dengan AWS layanan dependen atas nama Anda, Anda memerlukan peran AWS CodeBuild layanan. Anda dapat membuat peran CodeBuild layanan dengan menggunakan AWS CodePipeline konsol CodeBuild atau. Untuk informasi, lihat:

- Buat proyek build (konsol)
- Buat pipeline yang menggunakan CodeBuild (CodePipeline konsol)
- Menambahkan tindakan CodeBuild build ke pipeline (CodePipeline konsol)
- Mengubah setelan proyek build (konsol)

Jika Anda tidak berencana untuk menggunakan konsol ini, bagian ini menjelaskan cara membuat peran CodeBuild layanan dengan konsol IAM atau. AWS CLI

> **Important**
>
> CodeBuild menggunakan peran layanan untuk semua operasi yang dilakukan atas nama Anda. Jika peran tersebut mencakup izin yang seharusnya tidak dimiliki pengguna, Anda dapat secara tidak sengaja meningkatkan izin pengguna. Pastikan peran memberikan keistimewaan paling rendah.
> Peran layanan yang dijelaskan di halaman ini berisi kebijakan yang memberikan izin minimum yang diperlukan untuk digunakan CodeBuild. Anda mungkin perlu menambahkan izin tambahan, tergantung pada kasus penggunaan Anda.

Untuk membuat peran CodeBuild layanan (konsol)

1. Buka konsol IAM di https://console.aws.amazon.com/iam/.

   Anda seharusnya sudah masuk ke konsol dengan menggunakan salah satu dari berikut ini:

   - Akun AWS root Anda. Ini tidak disarankan. Untuk informasi selengkapnya, lihat Pengguna Akun AWS root di Panduan pengguna.
   - Pengguna administrator di AWS akun Anda. Untuk informasi selengkapnya, lihat Membuat pengguna dan Grup Akun AWS root pertama Anda di Panduan pengguna.
   - Pengguna di AWS akun Anda dengan izin untuk melakukan serangkaian tindakan minimum berikut:

     ```
     iam:AddRoleToInstanceProfile
     ```

```
iam:AttachRolePolicy
iam:CreateInstanceProfile
iam:CreatePolicy
iam:CreateRole
iam:GetRole
iam:ListAttachedRolePolicies
iam:ListPolicies
iam:ListRoles
iam:PassRole
iam:PutRolePolicy
iam:UpdateAssumeRolePolicy
```

Untuk informasi selengkapnya, lihat Ikh tisar Kebijakan IAM di Panduan pengguna.

2. Di panel navigasi, pilih Kebijakan.
3. Pilih Buat Kebijakan.
4. Pada halaman Buat Kebijakan, pilih JSON.
5. Untuk kebijakan JSON, masukkan yang berikut ini, lalu pilih Kebijakan Tinjau:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "CloudWatchLogsPolicy",
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ],
      "Resource": "*"
    },
    {
      "Sid": "CodeCommitPolicy",
      "Effect": "Allow",
      "Action": [
        "codecommit:GitPull"
      ],
      "Resource": "*"
    },
```

```
    {
      "Sid": "S3GetObjectPolicy",
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:GetObjectVersion"
      ],
      "Resource": "*"
    },
    {
      "Sid": "S3PutObjectPolicy",
      "Effect": "Allow",
      "Action": [
        "s3:PutObject"
      ],
      "Resource": "*"
    },
    {
      "Sid": "ECRPullPolicy",
      "Effect": "Allow",
      "Action": [
        "ecr:BatchCheckLayerAvailability",
        "ecr:GetDownloadUrlForLayer",
        "ecr:BatchGetImage"
      ],
      "Resource": "*"
    },
    {
      "Sid": "ECRAuthPolicy",
      "Effect": "Allow",
      "Action": [
        "ecr:GetAuthorizationToken"
      ],
      "Resource": "*"
    },
    {
      "Sid": "S3BucketIdentity",
      "Effect": "Allow",
      "Action": [
        "s3:GetBucketAcl",
        "s3:GetBucketLocation"
      ],
      "Resource": "*"
    }
```

```
    ]
}
```

> **Note**
>
> Kebijakan ini berisi pernyataan yang memungkinkan akses ke sejumlah besar sumber AWS daya. AWS CodeBuild Untuk membatasi akses AWS sumber daya tertentu, ubah nilai `Resource` array. Untuk informasi selengkapnya, lihat dokumentasi keamanan untuk AWS layanan.

6. Pada halaman Kebijakan Tinjauan, untuk Nama Kebijakan, masukkan nama kebijakan (misalnya,**`CodeBuildServiceRolePolicy`**), lalu pilih Buat kebijakan.

> **Note**
>
> Jika Anda menggunakan nama yang berbeda, pastikan untuk menggunakannya selama prosedur ini.

7. Di panel navigasi, pilih Peran.
8. Pilih Buat peran.
9. Pada halaman Buat peran, dengan AWS Layanan sudah dipilih, pilih CodeBuild, lalu pilih Next:Permissions.
10. Pada halaman Lampirkan kebijakan izin, pilih CodeBuildServiceRolePolicy, lalu pilih Ber ikutnya: Tinjau.
11. Pada halaman Buat peran dan ulasan, untuk Nama peran, masukkan nama peran (misalnya,**`CodeBuildServiceRole`**), lalu pilih Buat peran.

Untuk membuat peran CodeBuild layanan (AWS CLI)

1. Pastikan Anda telah mengonfigurasi AWS CLI dengan kunci AWS akses dan kunci akses AWS rahasia yang sesuai dengan salah satu entitas IAM, seperti yang dijelaskan dalam prosedur sebelumnya. Untuk informasi selengkapnya, lihat Menyiapkan dengan AWS Command Line Interface di Panduan AWS Command Line Interface Pengguna.
2. Dalam direktori kosong pada workstation lokal atau contoh di mana AWS CLI diinstal, buat dua file bernama `create-role.json` dan`put-role-policy.json`. Jika Anda memilih nama file yang berbeda, pastikan untuk menggunakannya selama prosedur ini.

`create-role.json`:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "codebuild.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

> **Note**
>
> Kami menyarankan Anda menggunakan kunci syarat `aws:SourceAccount` dan `aws:SourceArn` untuk melindungi diri Anda dari masalah wakil yang membingungkan. Misalnya, Anda dapat mengedit kebijakan kepercayaan sebelumnya dengan blok kondisi berikut. Itu `aws:SourceAccount` adalah pemilik CodeBuild proyek dan `aws:SourceArn` adalah CodeBuild proyek ARN.

Jika Anda ingin membatasi peran layanan Anda ke AWS akun, `create-role.json` mungkin terlihat mirip dengan ini:

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
```

```
                "Service": "codebuild.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": [
                        "111122223333"
                    ]
                }
            }
        }
    ]
}
```

Jika Anda ingin membatasi peran layanan Anda ke CodeBuild proyek tertentu, `create-role.json` mungkin terlihat mirip dengan ini:

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "codebuild.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceArn": "arn:aws:codebuild:us-
east-1:111122223333:project/MyProject"
                }
            }
        }
    ]
}
```

> **Note**
>
> Jika Anda tidak tahu atau belum memutuskan nama untuk CodeBuild proyek Anda dan menginginkan pembatasan kebijakan kepercayaan pada pola ARN tertentu, Anda dapat mengganti bagian ARN tersebut dengan wildcard (*). Setelah Anda membuat proyek Anda, Anda kemudian dapat memperbarui kebijakan kepercayaan.

`put-role-policy.json`:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "CloudWatchLogsPolicy",
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ],
      "Resource": "*"
    },
    {
      "Sid": "CodeCommitPolicy",
      "Effect": "Allow",
      "Action": [
        "codecommit:GitPull"
      ],
      "Resource": "*"
    },
    {
      "Sid": "S3GetObjectPolicy",
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:GetObjectVersion"
```

```
      ],
      "Resource": "*"
    },
    {
      "Sid": "S3PutObjectPolicy",
      "Effect": "Allow",
      "Action": [
        "s3:PutObject"
      ],
      "Resource": "*"
    },
    {
      "Sid": "S3BucketIdentity",
      "Effect": "Allow",
      "Action": [
        "s3:GetBucketAcl",
        "s3:GetBucketLocation"
      ],
      "Resource": "*"
    }
  ]
}
```

> **Note**
>
> Kebijakan ini berisi pernyataan yang memungkinkan akses ke sejumlah besar sumber AWS daya. AWS CodeBuild Untuk membatasi akses AWS sumber daya tertentu, ubah nilai `Resource` array. Untuk informasi selengkapnya, lihat dokumentasi keamanan untuk AWS layanan.

3. Beralih ke direktori tempat Anda menyimpan file sebelumnya, dan kemudian jalankan dua perintah berikut, satu per satu, dalam urutan ini. Anda dapat menggunakan nilai yang berbeda untuk `CodeBuildServiceRole` dan`CodeBuildServiceRolePolicy`, tetapi pastikan untuk menggunakannya di sini.

```
aws iam create-role --role-name CodeBuildServiceRole --assume-role-policy-document
 file://create-role.json
```

```
aws iam put-role-policy --role-name CodeBuildServiceRole --policy-name
 CodeBuildServiceRolePolicy --policy-document file://put-role-policy.json
```

Enkripsi output build menggunakan kunci yang dikelola pelanggan

Jika Anda mengikuti langkah-langkah Memulai menggunakan konsol untuk mengakses AWS CodeBuild untuk pertama kalinya, kemungkinan besar Anda tidak memerlukan informasi dalam topik ini. Namun, saat Anda terus menggunakan CodeBuild, Anda mungkin ingin melakukan hal-hal seperti mengenkripsi artefak build.

AWS CodeBuild Untuk mengenkripsi artefak keluaran build-nya, perlu akses ke kunci KMS. Secara default, CodeBuild gunakan Kunci yang dikelola AWS untuk Amazon S3 di AWS akun Anda.

Jika Anda tidak ingin menggunakan Kunci yang dikelola AWS, Anda harus membuat dan mengonfigurasi kunci yang dikelola pelanggan sendiri. Bagian ini menjelaskan cara melakukan ini dengan konsol IAM.

Untuk informasi tentang kunci yang dikelola pelanggan, lihat AWS Key Management Service Konsep dan Membuat Kunci di Panduan AWS KMS Pengembang.

Untuk mengonfigurasi kunci yang dikelola pelanggan untuk digunakan oleh CodeBuild, ikuti petunjuk di bagian “Cara Memodifikasi Kebijakan Kunci” dari M emodifikasi Kebijakan Utama di Panduan Peng AWS KMS embang. Kemudian tambahkan pernyataan berikut (antara *### BEGIN ADDING STATEMENTS HERE ###* dan*### END ADDING STATEMENTS HERE ###*) ke kebijakan utama. Ellipses (`...`) digunakan untuk singkatnya dan untuk membantu Anda menemukan tempat menambahkan pernyataan. Jangan menghapus pernyataan apa pun, dan jangan mengetikkan elips ini ke dalam kebijakan kunci.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kms:Encrypt",
```

```
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
        "kms:DescribeKey"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "kms:ViaService": "s3.us-east-1.amazonaws.com",
          "kms:CallerAccount": "111122223333"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
        "kms:DescribeKey"
      ],
      "Resource": "*"
    }
  ]
}
```

- *region-ID*mewakili ID AWS wilayah tempat bucket Amazon S3 yang terkait CodeBuild berada (misalnya,`us-east-1`).
- *account-ID*mewakili ID AWS akun yang memiliki kunci yang dikelola pelanggan.
- *CodeBuild-service-role*mewakili nama peran CodeBuild layanan yang Anda buat atau identifikasi sebelumnya dalam topik ini.

> **Note**
>
> Untuk membuat atau mengonfigurasi kunci yang dikelola pelanggan melalui konsol IAM, Anda harus terlebih dahulu masuk Konsol Manajemen AWS dengan menggunakan salah satu dari berikut ini:

- Akun AWS root Anda. Ini tidak disarankan. Untuk informasi selengkapnya, lihat Pengguna Root Akun di Panduan pengguna.
- Pengguna administrator di AWS akun Anda. Untuk informasi selengkapnya, lihat Membuat pengguna dan Grup Akun AWS root pertama Anda di Panduan pengguna.
- Pengguna di AWS akun Anda dengan izin untuk membuat atau memodifikasi kunci yang dikelola pelanggan. Untuk informasi selengkapnya, lihat Izin yang Diperlukan untuk Menggunakan Kon AWS KMS sol di Panduan AWS KMS Pengembang.

Berinteraksi dengan CodeBuild menggunakan AWS CLI

Jika Anda mengikuti langkah-langkah Memulai menggunakan konsol untuk mengakses AWS CodeBuild untuk pertama kalinya, kemungkinan besar Anda tidak memerlukan informasi dalam topik ini. Namun, saat Anda terus menggunakan CodeBuild, Anda mungkin ingin melakukan hal-hal seperti mengizinkan pengguna menggunakan AWS CLI untuk berinteraksi dengan alih- CodeBuild alih (atau sebagai tambahan) CodeBuild konsol, CodePipeline konsol, atau AWS SDK.

Untuk menginstal dan mengonfigurasi AWS CLI, lihat Mem ulai Pengaturan dengan AWS Command Line Interface di Panduan AWS Command Line Interface Pengguna.

Setelah menginstal AWS CLI, selesaikan tugas-tugas berikut:

1. Jalankan perintah berikut untuk mengonfirmasi apakah instalasi Anda AWS CLI mendukung CodeBuild:

   ```
   aws codebuild list-builds
   ```

   Jika berhasil, informasi yang mirip dengan berikut ini akan muncul di output:

   ```
   {
     "ids": []
   }
   ```

   Tanda kurung kosong menunjukkan bahwa Anda belum menjalankan build apa pun.

2. Jika kesalahan adalah output, Anda harus menghapus versi Anda saat ini AWS CLI dan kemudian menginstal versi terbaru. Untuk informasi selengkapnya, lihat Menghapus Instalasi

AWS CLI dan AWS Command Line Interface Menginstal di Panduan AWS Command Line Interface Pengguna.

Referensi baris perintah untuk AWS CodeBuild

AWS CLI Ini menyediakan perintah untuk mengotomatisasi. AWS CodeBuild Gunakan informasi dalam topik ini sebagai pelengkap Panduan AWS Command Line Interface Pengguna dan AWS CLI Referensi untuk AWS CodeBuild.

Bukan apa yang Anda cari? Jika Anda ingin menggunakan AWS SDK untuk memanggil CodeBuild, lihat. AWS Referensi SDK dan alat

Untuk menggunakan informasi dalam topik ini, Anda seharusnya sudah menginstal AWS CLI dan mengonfigurasinya untuk digunakan CodeBuild, seperti yang dijelaskan dalamBerinteraksi dengan CodeBuild menggunakan AWS CLI.

Untuk menggunakan AWS CLI untuk menentukan titik akhir untuk CodeBuild, lihatTentukan AWS CodeBuild titik akhir (AWS CLI).

Jalankan perintah ini untuk mendapatkan daftar CodeBuild perintah.

```
aws codebuild help
```

Jalankan perintah ini untuk mendapatkan informasi tentang CodeBuild perintah, di mana *command-name* nama perintah.

```
aws codebuild command-name help
```

CodeBuild perintah meliputi:

- `batch-delete-builds`: Menghapus satu atau lebih build in. CodeBuild Untuk informasi selengkapnya, lihat Hapus build ()AWS CLI.
- `batch-get-builds`: Mendapat informasi tentang beberapa build in CodeBuild. Untuk informasi selengkapnya, lihat Lihat detail build (AWS CLI).
- `batch-get-projects`: Mendapat informasi tentang satu atau lebih proyek build tertentu. Untuk informasi selengkapnya, lihat Lihat detail proyek build (AWS CLI).
- `create-project`: Membuat proyek build. Untuk informasi selengkapnya, lihat Buat proyek build (AWS CLI).

- `delete-project`: Menghapus proyek build. Untuk informasi selengkapnya, lihat Menghapus proyek build (AWS CLI).
- `list-builds`: Daftar Nama Sumber Daya Amazon (ARN) untuk build in. CodeBuild Untuk informasi selengkapnya, lihat Melihat daftar ID build (AWS CLI).
- `list-builds-for-project`: Mendapat daftar ID build yang terkait dengan proyek build tertentu. Untuk informasi selengkapnya, lihat Melihat daftar ID build untuk proyek build (AWS CLI).
- `list-curated-environment-images`: Mendapat daftar gambar Docker yang dikelola oleh CodeBuild yang dapat Anda gunakan untuk build Anda. Untuk informasi selengkapnya, lihat Gambar Docker disediakan oleh CodeBuild.
- `list-projects`: Mendapat daftar nama proyek build. Untuk informasi selengkapnya, lihat Lihat daftar nama proyek build (AWS CLI).
- `start-build`: Mulai menjalankan build. Untuk informasi selengkapnya, lihat Jalankan build (AWS CLI).
- `stop-build`: Mencoba menghentikan build yang ditentukan agar tidak berjalan. Untuk informasi selengkapnya, lihat Hentikan membangun (AWS CLI).
- `update-project`: Mengubah informasi tentang proyek build yang ditentukan. Lihat informasi yang lebih lengkap di Mengubah setelan proyek build (AWS CLI).

AWS Referensi SDK dan alat untuk AWS CodeBuild

Untuk menggunakan salah satu AWS SDK atau alat untuk mengotomatiskan AWS CodeBuild, lihat sumber daya berikut.

Jika Anda ingin menggunakan AWS CLI untuk menjalankan CodeBuild, lihatReferensi baris perintah.

Didukung AWS SDK dan alat untuk AWS CodeBuild

AWS SDK dan alat berikut mendukung: CodeBuild

- AWS SDK untuk C ++. Untuk informasi selengkapnya, lihat bagian CodeBuild namespace Aws:: dari AWS SDK untuk Refer ensi API C++.
- AWS SDK untuk Go. Untuk informasi selengkapnya, lihat bagian codebuild dari AWS SDK for Go API Reference.
- AWS SDK for Java. Untuk informasi selengkapnya, lihat `com.amazonaws.services.codebuild.model` bagian `com.amazonaws.services.codebuild` dan referensi AWS SDK untuk Java API.

- AWS SDK untuk JavaScript di browser dan AWS SDK untuk JavaScript di Node.js. Untuk informasi lebih lanjut, lihat Kelas: AWS. CodeBuildbagian AWS SDK untuk Refer JavaScript ensi API.
- AWS SDK untuk.NET. Untuk informasi selengkapnya, lihat bagian Amazon.CodeBuild dan Amazon.CodeBuild.Model namespace dari AWS SDK untuk Refer ensi API .NET.
- AWS SDK untuk PHP. Untuk informasi selengkapnya, lihat CodeBuild bagian Namespace Aws\ dari AWS SDK untuk Refer ensi API PHP.
- AWS SDK untuk Python (Boto3). Untuk informasi lebih lanjut, lihat CodeBuild bagian Dokument asi Boto 3.
- AWS SDK untuk Ruby. Untuk informasi selengkapnya, lihat CodeBuild bagian Modul: Aws:: dari AWS SDK for Ruby API Reference.
- AWS Alat untuk PowerShell. Untuk informasi selengkapnya, lihat AWS CodeBuild bagian AWS Alat untuk Referensi PowerShell Cmdlet.

Menggunakan layanan ini dengan AWS SDK

AWS perangkat pengembangan perangkat lunak (SDK) tersedia untuk banyak bahasa pemrograman populer. Setiap SDK menyediakan API, contoh kode, dan dokumentasi yang memudahkan developer untuk membangun aplikasi dalam bahasa pilihan mereka.

Dokumentasi SDK	Contoh kode
AWS SDK untuk C++	AWS SDK untuk C++ contoh kode
AWS CLI	AWS CLI contoh kode
AWS SDK untuk Go	AWS SDK untuk Go contoh kode
AWS SDK untuk Java	AWS SDK untuk Java contoh kode
AWS SDK untuk JavaScript	AWS SDK untuk JavaScript contoh kode
AWS SDK untuk Kotlin	AWS SDK untuk Kotlin contoh kode
AWS SDK untuk .NET	AWS SDK untuk .NET contoh kode
AWS SDK untuk PHP	AWS SDK untuk PHP contoh kode

Dokumentasi SDK	Contoh kode
Alat AWS untuk PowerShell	Alat AWS untuk PowerShell contoh kode
AWS SDK untuk Python v4	AWS SDK untuk contoh kode Python v4
AWS SDK untuk Python (Boto3)	AWS SDK untuk Python (Boto3) contoh kode
AWS SDK untuk Ruby	AWS SDK untuk Ruby contoh kode
AWS SDK for Rust	AWS SDK for Rust contoh kode
AWS SDK for SAP ABAP	AWS SDK for SAP ABAP contoh kode
AWS SDK for Swift	AWS SDK for Swift contoh kode

Untuk contoh khusus untuk layanan ini, lihat Contoh kode untuk CodeBuild menggunakan AWS SDK.

> **Ketersediaan contoh**
>
> Tidak dapat menemukan apa yang Anda butuhkan? Minta contoh kode menggunakan tautan Berikan umpan balik di bagian bawah halaman ini.

Tentukan AWS CodeBuild titik akhir

Anda dapat menggunakan AWS Command Line Interface (AWS CLI) atau salah satu AWS SDK untuk menentukan titik akhir yang digunakan oleh AWS CodeBuild. Ada titik akhir untuk setiap wilayah yang CodeBuild tersedia. Selain titik akhir regional, empat wilayah juga memiliki titik akhir Standar Pemrosesan Informasi Federal (FIPS). Untuk informasi selengkapnya tentang titik akhir FIPS, lihat ikhtisar FIPS 140-2.

Menentukan titik akhir adalah opsional. Jika Anda tidak secara eksplisit memberi tahu titik akhir CodeBuild mana yang akan digunakan, layanan akan menggunakan titik akhir yang terkait dengan wilayah yang AWS digunakan akun Anda. CodeBuildtidak pernah default ke titik akhir FIPS. Jika Anda ingin menggunakan titik akhir FIPS, Anda harus meng CodeBuild aitkannya menggunakan salah satu metode berikut.

> **Note**
>
> Anda dapat menggunakan nama alias atau wilayah untuk menentukan titik akhir menggunakan AWS SDK. Jika Anda menggunakan AWS CLI, maka Anda harus menggunakan nama titik akhir lengkap.

Untuk titik akhir yang dapat digunakan dengan CodeBuild, lihat CodeBuild wilayah dan titik akhir.

Topik

- Tentukan AWS CodeBuild titik akhir (AWS CLI)
- Tentukan AWS CodeBuild titik akhir (AWS SDK)

Tentukan AWS CodeBuild titik akhir (AWS CLI)

Anda dapat menggunakan AWS CLI untuk menentukan titik akhir yang AWS CodeBuild diakses dengan menggunakan `--endpoint-url` argumen dalam CodeBuild perintah apa pun. Misalnya, jalankan perintah ini untuk mendapatkan daftar nama build proyek menggunakan titik akhir Standar Pemrosesan Informasi Federal (FIPS) di Wilayah AS Timur (Virginia Utara):

```
aws codebuild list-projects --endpoint-url https://codebuild-fips.us-
east-1.amazonaws.com
```

Sertakan `https://` di awal titik akhir.

Ar `--endpoint-url` AWS CLI gumen tersedia untuk semua AWS layanan. Untuk informasi selengkapnya tentang ini dan AWS CLI argumen lainnya, lihat Refer AWS CLI ensi Perintah.

Tentukan AWS CodeBuild titik akhir (AWS SDK)

Anda dapat menggunakan AWS SDK untuk menentukan titik akhir AWS CodeBuild yang diakses. Meskipun contoh ini menggunakan AWS SDK untuk Java, Anda dapat menentukan titik akhir dengan AWS SDK lainnya.

Gunakan `withEndpointConfiguration` metode ini saat membangun AWSCodeBuild klien. Berikut adalah format yang digunakan:

```
AWSCodeBuild awsCodeBuild = AWSCodeBuildClientBuilder.standard().
```

```
    withEndpointConfiguration(new AwsClientBuilder.EndpointConfiguration("endpoint",
 "region")).
    withCredentials(new AWSStaticCredentialsProvider(sessionCredentials)).
    build();
```

Untuk informasi tentang`AWSCodeBuildClientBuilder`, lihat Kelas AWSCodeBuildClientBuilder.

KredenSIAL yang digunakan `withCredentials` harus bertipe`AWSCredentialsProvider`. Untuk informasi selengkapnya, lihat Bek erja dengan AWS kredensional.

Jangan sertakan `https://` di awal titik akhir.

Jika Anda ingin menentukan titik akhir non-FIPS, Anda dapat menggunakan wilayah alih-alih titik akhir yang sebenarnya. Misalnya, untuk menentukan titik akhir di wilayah AS Timur (Virginia Utara), Anda dapat menggunakan sebagai `us-east-1` pengganti nama titik akhir lengkap,`codebuild.us-east-1.amazonaws.com`.

Jika Anda ingin menentukan titik akhir FIPS, Anda dapat menggunakan alias untuk menyederhanakan kode Anda. Hanya titik akhir FIPS yang memiliki alias. Titik akhir lainnya harus ditentukan menggunakan wilayah atau nama lengkapnya.

Tabel berikut mencantumkan alias untuk masing-masing dari empat titik akhir FIPS yang tersedia:

Nama wilayah	Wilayah	Titik akhir	Alias
US East (Northern Virginia)	us-east-1	codebuild-fips.us-east-1.amazonaws.com	us-timur-1-fips
AS Timur (Ohio)	us-east-2	codebuild-fips.us-east-2.amazonaws.com	us-timur-2-fips
AS Barat (California Utara)	us-west-1	codebuild-fips.us-west-1.amazonaws.com	us-barat-1-fips
AS Barat (Oregon)	us-west-2	codebuild-fips.us-west-2.amazonaws.com	us-barat-2-fips

Untuk menentukan penggunaan titik akhir FIPS di wilayah AS Barat (Oregon) menggunakan alias:

```
AWSCodeBuild awsCodeBuild = AWSCodeBuildClientBuilder.standard().
    withEndpointConfiguration(new AwsClientBuilder.EndpointConfiguration("us-west-2-
fips", "us-west-2")).
    withCredentials(new AWSStaticCredentialsProvider(sessionCredentials)).
    build();
```

Untuk menentukan penggunaan titik akhir non-FIPS di wilayah AS Timur (Virginia Utara):

```
AWSCodeBuild awsCodeBuild = AWSCodeBuildClientBuilder.standard().
    withEndpointConfiguration(new AwsClientBuilder.EndpointConfiguration("us-east-1",
 "us-east-1")).
    withCredentials(new AWSStaticCredentialsProvider(sessionCredentials)).
    build();
```

Untuk menentukan penggunaan titik akhir non-FIPS di wilayah Asia Pasifik (Mumbai):

```
AWSCodeBuild awsCodeBuild = AWSCodeBuildClientBuilder.standard().
    withEndpointConfiguration(new AwsClientBuilder.EndpointConfiguration("ap-south-1",
 "ap-south-1")).
    withCredentials(new AWSStaticCredentialsProvider(sessionCredentials)).
    build();
```

Gunakan AWS CodeBuild dengan AWS CodePipeline untuk menguji kode dan menjalankan build

Anda dapat mengotomatiskan proses rilis Anda dengan menggunakan AWS CodePipeline untuk menguji kode Anda dan menjalankan build Anda dengannya AWS CodeBuild.

Tabel berikut mencantumkan tugas dan metode yang tersedia untuk melakukannya. Menggunakan AWS SDK untuk menyelesaikan tugas-tugas ini berada di luar cakupan topik ini.

Tugas	Pendekatan yang tersedia	Pendekatan yang dijelaskan dalam topik ini
Buat pipeline pengiriman berkelanjutan (CD) CodePipeline yang mengotomatiskan build dengan CodeBuild	• CodePipeline konsol • AWS CLI • AWS SDK	• Gunakan CodePipeline konsol • Gunakan AWS CLI • Anda dapat menyesuaikan informasi dalam topik ini untuk menggunakan AWS SDK. Untuk informasi selengkapnya, lihat dokumentasi `create-pipeline` tindakan untuk bahasa pemrograman Anda di bagian SDK Alat untuk Amazon Web Services atau lihat `CreatePipeline` di Referensi AWS CodePipeline API.
Tambahkan pengujian dan buat otomatisasi CodeBuild dengan saluran yang ada di CodePipeline	• CodePipeline konsol • AWS CLI • AWS SDK	• Gunakan CodePipeline konsol untuk menambahkan otomatisasi build • Gunakan CodePipeline konsol untuk menambahkan otomatisasi pengujian • Untuk itu AWS CLI, Anda dapat menyesuaikan informasi dalam topik ini untuk membuat pipeline yang berisi tindakan CodeBuild build atau tindakan pengujian. Untuk informasi selengkapnya, lihat Mengedit pipeline (AWS CLI) dan referensi struktur CodePipeline pipa di Panduan AWS CodePipeline Pengguna. • Anda dapat menyesuaikan informasi dalam topik ini untuk menggunakan AWS SDK. Untuk informasi selengkapnya, lihat dokumentasi `update-pipeline` tindakan untuk bahasa pemrograman Anda melalui bagian SDK Alat untuk Amazon Web Services atau lihat `UpdatePipeline` di Referensi AWS CodePipeline API.

Topik

- Prasyarat
- Buat pipeline yang menggunakan CodeBuild (CodePipeline konsol)

- Buat pipeline yang menggunakan CodeBuild (AWS CLI)
- Menambahkan tindakan CodeBuild build ke pipeline (CodePipeline konsol)
- Menambahkan tindakan CodeBuild pengujian ke pipeline (CodePipeline konsol)

Prasyarat

1. Jawab pertanyaan diMerencanakan build.
2. Jika Anda menggunakan pengguna untuk mengakses al CodePipeline ih-alih akun AWS root atau pengguna administrator, lampirkan kebijakan terkelola yang diberi nama `AWSCodePipelineFullAccess` ke pengguna (atau ke grup IAM tempat pengguna tersebut berada). Menggunakan akun AWS root tidak disarankan. Kebijakan ini memberikan izin pengguna untuk membuat pipeline di CodePipeline. Untuk informasi selengkapnya, lihat Melampirkan kebijakan terkelola di Panduan pengguna.

   > **Note**
   >
   > Entitas IAM yang melampirkan kebijakan ke pengguna (atau ke grup IAM tempat pengguna berada) harus memiliki izin di IAM untuk melampirkan kebijakan. Untuk informasi selengkapnya, lihat Mendelegasikan izin untuk mengelola pengguna, grup, dan kredenSIAL IAM di Panduan pengguna.

3. Buat peran CodePipeline layanan, jika Anda belum memilikinya di AWS akun Anda. CodePipeline menggunakan peran layanan ini untuk berinteraksi dengan AWS layanan lain, termasuk AWS CodeBuild, atas nama Anda. Misalnya, untuk menggunakan AWS CLI untuk membuat peran CodePipeline layanan, jalankan `create-role` perintah IAM:

   Untuk Linux, macOS, atau Unix:

   ```
   aws iam create-role --role-name AWS-CodePipeline-CodeBuild-
   Service-Role --assume-role-policy-document '{"Version":
    "2012-10-17",       "Statement":{"Effect":"Allow","Principal":
   {"Service":"codepipeline.amazonaws.com"},"Action":"sts:AssumeRole"}}'
   ```

   Untuk Windows:

   ```
   aws iam create-role --role-name AWS-CodePipeline-CodeBuild-Service-Role --assume-
   role-policy-document "{\"Version\":\"2012-10-17\",       \"Statement\":{\"Effect
   ```

```
\":\"Allow\",\"Principal\":{\"Service\":\"codepipeline.amazonaws.com\"},\"Action\":
\"sts:AssumeRole\"}}"
```

> **Note**
>
> Entitas IAM yang membuat peran CodePipeline layanan ini harus memiliki izin di IAM untuk membuat peran layanan.

4. Setelah Anda membuat peran CodePipeline layanan atau mengidentifikasi peran yang sudah ada, Anda harus menambahkan kebijakan peran CodePipeline layanan default ke peran layanan seperti yang dijelaskan di T injau kebijakan peran CodePipeline layanan default di Panduan AWS CodePipeline Pengguna, jika itu belum menjadi bagian dari kebijakan untuk peran tersebut.

> **Note**
>
> Entitas IAM yang menambahkan kebijakan peran CodePipeline layanan ini harus memiliki izin di IAM untuk menambahkan kebijakan peran layanan ke peran layanan.

5. Buat dan unggah kode sumber ke jenis repositori yang didukung oleh CodeBuild dan CodePipeline, seperti CodeCommit, Amazon S3, Bitbucket, atau GitHub. Kode sumber harus berisi file buildspec, tetapi Anda dapat mendeklarasikannya ketika Anda mendefinisikan proyek build nanti dalam topik ini. Untuk informasi selengkapnya, lihat Referensi Buildspec.

> **Important**
>
> Jika Anda berencana menggunakan pipeline untuk menerapkan kode sumber bawaan, artefak keluaran build harus kompatibel dengan sistem penerapan yang Anda gunakan.
>
> - Untuk OpsWorks, lihat Sumber aplikasi dan Menggunakan CodePipeline dengan OpsWorks di Panduan OpsWorks Pengguna.

Buat pipeline yang menggunakan CodeBuild (CodePipeline konsol)

Gunakan prosedur berikut untuk membuat pipeline yang digunakan CodeBuild untuk membangun dan menerapkan kode sumber Anda.

Untuk membuat pipeline yang hanya menguji kode sumber Anda:

- Gunakan prosedur berikut untuk membuat pipeline, lalu hapus tahap Build dan Beta dari pipeline. Kemudian gunakan Menambahkan tindakan CodeBuild pengujian ke pipeline (CodePipeline konsol) prosedur dalam topik ini untuk menambahkan tindakan pengujian yang digunakan ke pipeline CodeBuild.
- Gunakan salah satu prosedur lain dalam topik ini untuk membuat pipeline, lalu gunakan Menambahkan tindakan CodeBuild pengujian ke pipeline (CodePipeline konsol) prosedur dalam topik ini untuk menambahkan tindakan pengujian yang digunakan ke pipeline CodeBuild.

Untuk menggunakan wizard buat pipa CodePipeline untuk membuat pipeline yang menggunakan CodeBuild

1. Masuk ke Konsol Manajemen AWS dengan menggunakan:

   - Akun AWS root Anda. Ini tidak disarankan. Untuk informasi selengkapnya, lihat Pengguna root akun di Panduan pengguna.
   - Pengguna administrator di AWS akun Anda. Untuk informasi selengkapnya, lihat Membuat pengguna dan grup Akun AWS root pertama Anda di Panduan pengguna.
   - Pengguna di AWS akun Anda dengan izin untuk menggunakan serangkaian tindakan minimum berikut:

```
codepipeline:*
iam:ListRoles
iam:PassRole
s3:CreateBucket
s3:GetBucketPolicy
s3:GetObject
s3:ListAllMyBuckets
s3:ListBucket
s3:PutBucketPolicy
codecommit:ListBranches
codecommit:ListRepositories
codedeploy:GetApplication
codedeploy:GetDeploymentGroup
codedeploy:ListApplications
codedeploy:ListDeploymentGroups
elasticbeanstalk:DescribeApplications
elasticbeanstalk:DescribeEnvironments
lambda:GetFunctionConfiguration
lambda:ListFunctions
opsworks:DescribeStacks
```

```
opsworks:DescribeApps
opsworks:DescribeLayers
```

2. Buka AWS CodePipeline konsol di https://console.aws.amazon.com/codesuite/codepipeline/home.
3. Di pemilih Wil AWS ayah, pilih Wil AWS ayah tempat AWS sumber daya proyek build Anda berada. Ini harus menjadi Wil AWS ayah di CodeBuild mana didukung. Untuk informasi selengkapnya, lihat AWS CodeBuild di Referensi Umum Amazon Web Services.
4. Buat pipa. Jika halaman CodePipeline informasi ditampilkan, pilih Buat pipeline. Jika halaman Pipelines ditampilkan, pilih Buat pipeline.
5. Pada halaman Langkah 1: Pilih pengaturan pipeline, untuk nama Pipeline, masukkan nama untuk pipeline (misalnya,**`CodeBuildDemoPipeline`**). Jika Anda memilih nama yang berbeda, pastikan untuk menggunakannya selama prosedur ini.
6. Untuk Nama Per an, lakukan salah satu hal berikut:

   Pilih Peran layanan baru, dan di Nama Peran, masukkan nama untuk peran layanan baru Anda.

   Pilih Peran layanan yang ada, lalu pilih peran CodePipeline layanan yang Anda buat atau identifikasi sebagai bagian dari prasyarat topik ini.
7. Untuk penyimpanan artefak, lakukan salah satu hal berikut:

   - Pilih Lokasi default untuk menggunakan penyimpanan artefak default, seperti bucket artefak S3 yang ditetapkan sebagai default, untuk pipeline Anda di Wil AWS ayah yang telah Anda pilih untuk pipeline Anda.
   - Pilih Lokasi kustom jika Anda sudah memiliki penyimpanan artefak yang telah Anda buat, seperti bucket artefak S3, di Wilayah yang sama dengan AWS pipeline Anda.

   > **Note**
   >
   > Ini bukan bucket sumber untuk kode sumber pipeline Anda. Ini adalah penyimpanan artifact untuk alur Anda. Penyimpanan artefak terpisah, seperti bucket S3, diperlukan untuk setiap pipeline, di Wilayah yang sama AWS dengan pipeline.
8. Pilih Berikutnya.
9. Pada Langkah 2: Tambahkan halaman tahap sumber, untuk Penye dia Sumber, lakukan salah satu hal berikut:

- Jika kode sumber Anda disimpan dalam bucket S3, pilih Amazon S3. Untuk Bucket, pilih bucket S3 yang berisi kode sumber Anda. Untuk kunci objek S3, masukkan nama file yang berisi kode sumber (misalnya,*file-name*.zip). Pilih Berikutnya.
- Jika kode sumber Anda disimpan dalam AWS CodeCommit repositori, pilih CodeCommit. Untuk Nama Repositori, pilih nama repositori yang berisi kode sumber. Untuk Nama cabang, pilih nama cabang yang berisi versi kode sumber yang ingin Anda buat. Pilih Berikutnya.
- Jika kode sumber Anda disimpan dalam GitHub repositori, pilih GitHub. Pilih Connect to GitHub, dan ikuti petunjuk untuk mengotentikasi. GitHub Untuk Repositori, pilih nama repositori yang berisi kode sumber. Untuk Branch, pilih nama cabang yang berisi versi kode sumber yang ingin Anda buat.

Pilih Berikutnya.

10. Pada Langkah 3: Tambahkan halaman tahap build, untuk Build provider, pilih CodeBuild.
11. Jika Anda sudah memiliki proyek build yang ingin Anda gunakan, untuk nama Proyek, pilih nama proyek build dan lewati ke langkah berikutnya dalam prosedur ini.

    Jika Anda perlu membuat proyek CodeBuild build baru, ikuti instruksi Buat proyek build (konsol) dan kembali ke prosedur ini.

    Jika Anda memilih proyek build yang ada, itu harus memiliki pengaturan artefak keluaran build yang sudah ditentukan (meskipun CodePipeline mengesampingkannya). Untuk informasi selengkapnya, lihat Mengubah setelan proyek build (konsol).

    > **Important**
    >
    > Jika Anda mengaktifkan webhook untuk CodeBuild proyek, dan proyek digunakan sebagai langkah membangun CodePipeline, maka dua build identik dibuat untuk setiap komit. Satu build dipicu melalui webhook, dan satu melalui CodePipeline. Karena penagihan didasarkan per build, Anda akan ditagih untuk kedua build tersebut. Oleh karena itu, jika Anda menggunakan CodePipeline, kami sarankan Anda menonaktifkan webhook di CodeBuild. Di AWS CodeBuild konsol, kosongkan kotak Webhook. Untuk informasi selengkapnya, lihat Mengubah setelan proyek build (konsol).

12. Pada halaman Langkah 4: Tambahkan tahap penerapan, lakukan salah satu hal berikut:

- Jika Anda tidak ingin menerapkan artefak keluaran build, pilih Le wati, dan konfirmasikan pilihan ini saat diminta.
- Jika Anda ingin menerapkan artefak keluaran build, untuk Penyedia Deplo y, pilih penyedia penerapan, lalu tentukan pengaturan saat diminta.

Pilih Berikutnya.

13. Pada halaman Review, tinjau pilihan Anda, lalu pilih Buat pipeline.
14. Setelah pipeline berjalan dengan sukses, Anda bisa mendapatkan artefak keluaran build. Dengan pipeline yang ditampilkan di CodePipeline konsol, dalam tindakan Build, pilih tooltip. Catat nilai untuk artefak Output (misalnya, MyAppBuild).

> **Note**
>
> Anda juga bisa mendapatkan artefak keluaran build dengan memilih tautan Bangun artefak di halaman detail build di CodeBuild konsol. Untuk sampai ke halaman ini, lewati langkah-langkah lainnya dalam prosedur ini, dan lihatLihat detail build (konsol).

15. Buka konsol Amazon S3 di https://console.aws.amazon.com/s3/.
16. Dalam daftar ember, buka ember yang digunakan oleh pipa. Nama bucket harus mengikuti format`codepipeline-`*`region-ID`*`-`*`random-number`*. Anda dapat menggunakan AWS CLI untuk menjalankan CodePipeline get-pipeline perintah untuk mendapatkan nama bucket, di *`my-pipeline-name`* mana nama tampilan pipeline Anda:

```
aws codepipeline get-pipeline --name my-pipeline-name
```

Dalam output, `pipeline` objek berisi `artifactStore` objek, yang berisi `location` nilai dengan nama bucket.

17. Buka folder yang cocok dengan nama pipeline Anda (tergantung pada panjang nama pipeline, nama folder mungkin terpotong), lalu buka folder yang cocok dengan nilai artefak Output yang Anda catat sebelumnya.
18. Ekstrak isi file tersebut. Jika ada beberapa file di folder itu, ekstrak isi file dengan stempel waktu Modifikasi Terakhir terbaru. (Anda mungkin perlu memberikan file `.zip` ekstensi sehingga Anda dapat bekerja dengannya di utilitas ZIP sistem Anda.) Artefak keluaran build ada di konten file yang diekstraksi.

19. Jika Anda diinstruksikan CodePipeline untuk menerapkan artefak keluaran build, gunakan instruksi penyedia penerapan untuk mendapatkan artefak keluaran build pada target penerapan.

Buat pipeline yang menggunakan CodeBuild (AWS CLI)

Gunakan prosedur berikut untuk membuat pipeline yang digunakan CodeBuild untuk membangun kode sumber Anda.

Untuk menggunakan AWS CLI untuk membuat pipeline yang menyebarkan kode sumber bawaan Anda atau yang hanya menguji kode sumber Anda, Anda dapat menyesuaikan instruksi di Edit pipeline (AWS CLI) dan referensi struktur CodePipeline pipeline di Panduan AWS CodePipeline Pengguna.

1. Buat atau identifikasi proyek pembangunan di CodeBuild. Untuk informasi selengkapnya, lihat Membuat proyek build.

   > **Important**
   >
   > Proyek build harus menentukan pengaturan artefak keluaran build (meskipun CodePipeline mengesampingkannya). Untuk informasi lebih lanjut, lihat deskripsi `artifacts` diBuat proyek build (AWS CLI).

2. Pastikan Anda telah mengonfigurasi kunci AWS akses dan kunci akses AWS rahasia yang sesuai dengan salah satu entitas IAM yang dijelaskan dalam topik ini. AWS CLI Untuk informasi selengkapnya, lihat Menyiapkan AWS Command Line Interface di Panduan AWS Command Line Interface Pengguna.
3. Buat JSON-formatted file yang mewakili struktur pipa. Beri nama file `create-pipeline.json` atau yang serupa. Misalnya, JSON-formatted struktur ini membuat pipeline dengan aksi sumber yang mereferensikan bucket masukan S3 dan tindakan build yang menggunakan CodeBuild:

```
{
  "pipeline": {
    "roleArn": "arn:aws:iam::<account-id>:role/<AWS-CodePipeline-service-role-name>",
    "stages": [
      {
        "name": "Source",
        "actions": [
          {
```

```
            "inputArtifacts": [],
            "name": "Source",
            "actionTypeId": {
              "category": "Source",
              "owner": "AWS",
              "version": "1",
              "provider": "S3"
            },
            "outputArtifacts": [
              {
                "name": "MyApp"
              }
            ],
            "configuration": {
              "S3Bucket": "<bucket-name>",
              "S3ObjectKey": "<source-code-file-name.zip>"
            },
            "runOrder": 1
          }
        ]
      },
      {
        "name": "Build",
        "actions": [
          {
            "inputArtifacts": [
              {
                "name": "MyApp"
              }
            ],
            "name": "Build",
            "actionTypeId": {
              "category": "Build",
              "owner": "AWS",
              "version": "1",
              "provider": "CodeBuild"
            },
            "outputArtifacts": [
              {
                "name": "default"
              }
            ],
            "configuration": {
              "ProjectName": "<build-project-name>"
```

```
            },
            "runOrder": 1
          }
        ]
      }
    ],
    "artifactStore": {
      "type": "S3",
      "location": "<CodePipeline-internal-bucket-name>"
    },
    "name": "<my-pipeline-name>",
    "version": 1
  }
}
```

Dalam JSON-formatted data ini:

- Nilai `roleArn` harus sesuai dengan ARN dari peran CodePipeline layanan yang Anda buat atau identifikasi sebagai bagian dari prasyarat.
- Nilai `S3Bucket` dan `S3ObjectKey` dalam `configuration` asumsi kode sumber disimpan dalam ember S3. Untuk pengaturan untuk jenis repositori kode sumber lainnya, lihat referensi struktur CodePipeline pipa di AWS CodePipeline Panduan Pengguna.
- Nilai dari `ProjectName` adalah nama proyek CodeBuild build yang Anda buat sebelumnya dalam prosedur ini.
- Nilai `location` adalah nama bucket S3 yang digunakan oleh pipeline ini. Untuk informasi selengkapnya, lihat Membuat kebijakan untuk Bucket S3 untuk digunakan sebagai penyimpanan artefak CodePipeline di AWS CodePipeline Panduan Pengguna.
- Nilai dari `name` adalah nama pipeline ini. Semua nama pipeline harus unik untuk akun Anda.

Meskipun data ini hanya menjelaskan tindakan sumber dan tindakan build, Anda dapat menambahkan tindakan untuk aktivitas yang terkait dengan pengujian, menerapkan artefak keluaran build, menjalankan AWS Lambda fungsi, dan banyak lagi. Untuk informasi selengkapnya, lihat referensi struktur AWS CodePipeline pipa di Panduan AWS CodePipeline Pengguna.

4. Beralih ke folder yang berisi file JSON, dan kemudian jalankan CodePipeline create-pipeline perintah, tentukan nama file:

```
aws codepipeline create-pipeline --cli-input-json file://create-pipeline.json
```

> **Note**
>
> Anda harus membuat pipeline di AWS Wilayah yang CodeBuild didukung. Untuk informasi selengkapnya, lihat AWS CodeBuild di Referensi Umum Amazon Web Services.

JSON-formatted Data muncul di output, dan CodePipeline membuat pipa.

5. Untuk mendapatkan informasi tentang status pipeline, jalankan CodePipeline get-pipeline-state perintah, dengan menentukan nama pipeline:

```
aws codepipeline get-pipeline-state --name <my-pipeline-name>
```

Dalam output, cari informasi yang mengonfirmasi bahwa build berhasil. Ellipses (`...`) digunakan untuk menunjukkan data yang telah dihilangkan untuk singkatnya.

```
{
  ...
  "stageStates": [
    ...
    {
      "actionStates": [
        {
          "actionName": "CodeBuild",
          "latestExecution": {
            "status": "SUCCEEDED",
            ...
          },
          ...
        }
      ]
    }
  ]
}
```

Jika Anda menjalankan perintah ini terlalu dini, Anda mungkin tidak melihat informasi apa pun tentang tindakan build. Anda mungkin perlu menjalankan perintah ini beberapa kali hingga pipeline selesai menjalankan aksi build.

6. Setelah build berhasil, ikuti petunjuk ini untuk mendapatkan artefak keluaran build. Buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/

   > **Note**
   >
   > Anda juga bisa mendapatkan artefak keluaran build dengan memilih tautan artefak Build di halaman detail build terkait di CodeBuild konsol. Untuk sampai ke halaman ini, lewati langkah-langkah lainnya dalam prosedur ini, dan lihatLihat detail build (konsol).

7. Dalam daftar ember, buka ember yang digunakan oleh pipa. Nama bucket harus mengikuti formatnya`codepipeline-`*`<region-ID>`*`-`*`<random-number>`*. Anda bisa mendapatkan nama bucket dari `create-pipeline.json` file atau Anda dapat menjalankan CodePipeline get-pipeline perintah untuk mendapatkan nama bucket.

   ```
   aws codepipeline get-pipeline --name <pipeline-name>
   ```

   Dalam output, `pipeline` objek berisi `artifactStore` objek, yang berisi `location` nilai dengan nama ember.

8. Buka folder yang cocok dengan nama pipeline Anda (misalnya,*`<pipeline-name>`*).

9. Di folder itu, buka folder bernama`default`.

10. Ekstrak isi file tersebut. Jika ada beberapa file di folder itu, ekstrak konten file dengan stempel waktu Terakhir Modifikasi terbaru. (Anda mungkin perlu memberikan file `.zip` ekstensi sehingga Anda dapat bekerja dengannya di utilitas ZIP sistem Anda.) Artefak keluaran build ada dalam konten file yang diekstraksi.

Menambahkan tindakan CodeBuild build ke pipeline (CodePipeline konsol)

1. Masuk ke Konsol Manajemen AWS dengan menggunakan:

   - Akun AWS root Anda. Ini tidak disarankan. Untuk informasi selengkapnya, lihat Pengguna root akun di Panduan pengguna.

- Pengguna administrator di AWS akun Anda. Untuk informasi selengkapnya, lihat Membuat pengguna dan grup Akun AWS root pertama Anda di Panduan pengguna.
- Pengguna di AWS akun Anda dengan izin untuk melakukan serangkaian tindakan minimum berikut:

```
codepipeline:*
iam:ListRoles
iam:PassRole
s3:CreateBucket
s3:GetBucketPolicy
s3:GetObject
s3:ListAllMyBuckets
s3:ListBucket
s3:PutBucketPolicy
codecommit:ListBranches
codecommit:ListRepositories
codedeploy:GetApplication
codedeploy:GetDeploymentGroup
codedeploy:ListApplications
codedeploy:ListDeploymentGroups
elasticbeanstalk:DescribeApplications
elasticbeanstalk:DescribeEnvironments
lambda:GetFunctionConfiguration
lambda:ListFunctions
opsworks:DescribeStacks
opsworks:DescribeApps
opsworks:DescribeLayers
```

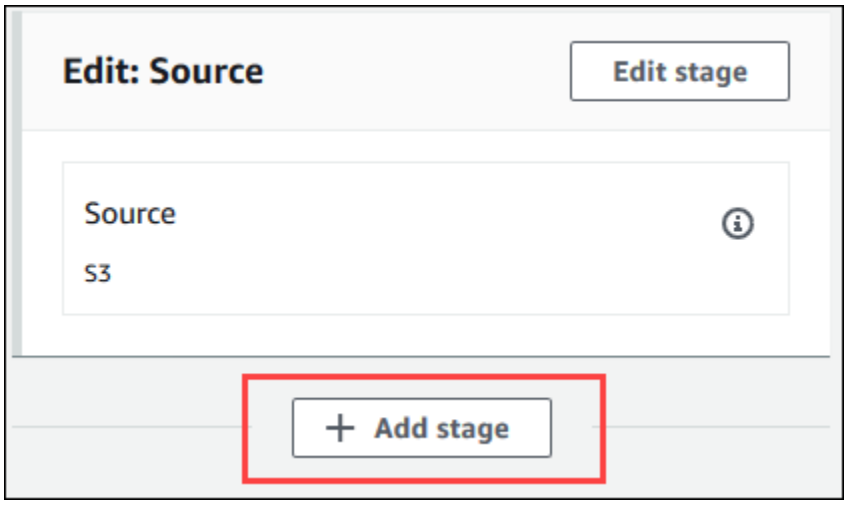
2. Buka CodePipeline konsol di https://console.aws.amazon.com/codesuite/codepipeline/home.
3. Di pemilih AWS wilayah, pilih Wil AWS ayah tempat pipa Anda berada. Ini harus menjadi Wilayah di CodeBuild mana didukung. Untuk informasi selengkapnya, lihat CodeBuild di Referensi Umum Amazon Web Services.
4. Pada halaman Pipelines, pilih nama pipeline.
5. Pada halaman rincian pipeline, dalam tindakan Sumber, pilih tooltip. Catat nilai untuk artefak Output (misalnya, MyApp).

> **Note**
>
> Prosedur ini menunjukkan cara menambahkan tindakan build di tahap build antara tahap Sumber dan Beta. Jika Anda ingin menambahkan tindakan build di tempat lain, pilih

tooltip pada tindakan tepat sebelum tempat di mana Anda ingin menambahkan tindakan build, dan catat nilai untuk artefak Output.

6. Pilih Edit.
7. Di antara tahap Sumber dan Beta, pilih Tam bah tahap.

> **Note**
>
> Prosedur ini menunjukkan cara menambahkan tahap build antara tahap Sumber dan Beta ke pipeline Anda. Untuk menambahkan tindakan build ke tahap yang ada, pilih Edit stage di stage, lalu lanjutkan ke langkah 8 dari prosedur ini. Untuk menambahkan tahap build di tempat lain, pilih Tambahkan tahap di tempat yang diinginkan.



8. Untuk Nama Pang gung, masukkan nama tahap build (misalnya,**`Build`**). Jika Anda memilih nama yang berbeda, gunakan selama prosedur ini.
9. Di dalam tahap yang dipilih, pilih Tam bah tindakan.

> **Note**
>
> Prosedur ini menunjukkan cara menambahkan tindakan build di dalam tahap build. Untuk menambahkan tindakan build di tempat lain, pilih Tambahkan tindakan di tempat yang diinginkan. Anda mungkin harus terlebih dahulu memilih tahap Edit di tahap yang ada di mana Anda ingin menambahkan tindakan build.

10. Di Edit tindakan, untuk Nama tindakan, masukkan nama untuk tindakan (misalnya,**`CodeBuild`**). Jika Anda memilih nama yang berbeda, gunakan selama prosedur ini.
11. Untuk Penyedia tindakan, pilih CodeBuild.
12. Jika Anda sudah memiliki proyek build yang ingin Anda gunakan, untuk nama Proyek, pilih nama proyek build dan lewati ke langkah berikutnya dalam prosedur ini.

    Jika Anda perlu membuat proyek CodeBuild build baru, ikuti instruksi Buat proyek build (konsol) dan kembali ke prosedur ini.

    Jika Anda memilih proyek build yang ada, itu harus memiliki pengaturan artefak keluaran build yang sudah ditentukan (meskipun CodePipeline mengesampingkannya). Untuk informasi lebih lanjut, lihat deskripsi Artefak di Buat proyek build (konsol) atauMengubah setelan proyek build (konsol).

    > **Important**
    >
    > Jika Anda mengaktifkan webhook untuk CodeBuild proyek, dan proyek digunakan sebagai langkah membangun CodePipeline, maka dua build identik dibuat untuk setiap komit. Satu build dipicu melalui webhook dan satu melalui CodePipeline. Karena penagihan didasarkan per build, Anda akan ditagih untuk kedua build. Oleh karena itu, jika Anda menggunakan CodePipeline, kami sarankan Anda menonaktifkan webhook di CodeBuild. Di CodeBuild konsol, kosongkan kotak Webhook. Untuk informasi selengkapnya, lihat Mengubah setelan proyek build (konsol)
13. Untuk artefak masukan, pilih artefak keluaran yang Anda catat sebelumnya dalam prosedur ini.
14. Untuk artefak keluaran, masukkan nama untuk artefak keluaran (misalnya,**`MyAppBuild`**).
15. Pilih Tambahkan tindakan.
16. Pilih Simpan, lalu pilih Simpan untuk menyimpan perubahan Anda ke pipeline.
17. Pilih Perubahan rilis.
18. Setelah pipeline berjalan dengan sukses, Anda bisa mendapatkan artefak keluaran build. Dengan pipeline yang ditampilkan di CodePipeline konsol, dalam tindakan Build, pilih tooltip. Catat nilai untuk artefak Output (misalnya, MyAppBuild).

> **Note**
>
> Anda juga bisa mendapatkan artefak keluaran build dengan memilih tautan Bangun artefak di halaman detail build di CodeBuild konsol. Untuk membuka halaman ini, lihatLihat detail build (konsol), lalu lewati ke langkah 31 dari prosedur ini.

19. Buka konsol Amazon S3 di https://console.aws.amazon.com/s3/.
20. Dalam daftar ember, buka ember yang digunakan oleh pipa. Nama bucket harus mengikuti formatcodepipeline-*region-ID*-*random-number*. Anda dapat menggunakan AWS CLI untuk menjalankan CodePipeline get-pipeline perintah untuk mendapatkan nama bucket:

    ```
    aws codepipeline get-pipeline --name my-pipeline-name
    ```

    Dalam output, `pipeline` objek berisi `artifactStore` objek, yang berisi `location` nilai dengan nama bucket.
21. Buka folder yang cocok dengan nama pipeline Anda (tergantung pada panjang nama pipeline, nama folder mungkin terpotong), lalu buka folder yang cocok dengan nilai artefak Output yang Anda catat sebelumnya dalam prosedur ini.
22. Ekstrak isi file tersebut. Jika ada beberapa file di folder itu, ekstrak isi file dengan stempel waktu Modifikasi Terakhir terbaru. (Anda mungkin perlu memberikan file `.zip` ekstensi sehingga Anda dapat bekerja dengannya di utilitas ZIP sistem Anda.) Artefak keluaran build ada di konten file yang diekstraksi.
23. Jika Anda diinstruksikan CodePipeline untuk menerapkan artefak keluaran build, gunakan instruksi penyedia penerapan untuk mendapatkan artefak keluaran build pada target penerapan.

Menambahkan tindakan CodeBuild pengujian ke pipeline (CodePipeline konsol)

1. Masuk ke Konsol Manajemen AWS dengan menggunakan:
    - Akun AWS root Anda. Ini tidak disarankan. Untuk informasi selengkapnya, lihat Pengguna root akun di Panduan pengguna.
    - Pengguna administrator di AWS akun Anda. Untuk informasi selengkapnya, lihat Membuat pengguna dan grup Akun AWS root pertama Anda di Panduan pengguna.

- Pengguna di AWS akun Anda dengan izin untuk melakukan serangkaian tindakan minimum berikut:

```
codepipeline:*
iam:ListRoles
iam:PassRole
s3:CreateBucket
s3:GetBucketPolicy
s3:GetObject
s3:ListAllMyBuckets
s3:ListBucket
s3:PutBucketPolicy
codecommit:ListBranches
codecommit:ListRepositories
codedeploy:GetApplication
codedeploy:GetDeploymentGroup
codedeploy:ListApplications
codedeploy:ListDeploymentGroups
elasticbeanstalk:DescribeApplications
elasticbeanstalk:DescribeEnvironments
lambda:GetFunctionConfiguration
lambda:ListFunctions
opsworks:DescribeStacks
opsworks:DescribeApps
opsworks:DescribeLayers
```

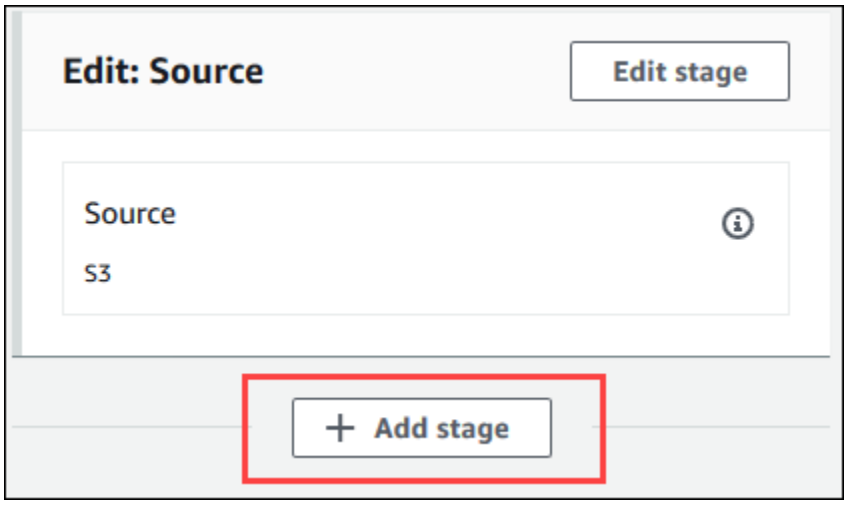
2. Buka CodePipeline konsol di https://console.aws.amazon.com/codesuite/codepipeline/home.
3. Di pemilih AWS wilayah, pilih Wil AWS ayah tempat pipa Anda berada. Ini harus menjadi Wil AWS ayah di CodeBuild mana didukung. Untuk informasi selengkapnya, lihat AWS CodeBuild di Referensi Umum Amazon Web Services.
4. Pada halaman Pipelines, pilih nama pipeline.
5. Pada halaman rincian pipeline, dalam tindakan Sumber, pilih tooltip. Catat nilai untuk artefak Output (misalnya, MyApp).

> **Note**
>
> Prosedur ini menunjukkan cara menambahkan tindakan pengujian di dalam tahap pengujian antara tahap Sumber dan Beta. Jika Anda ingin menambahkan tindakan

pengujian di tempat lain, letakkan penunjuk mouse Anda pada tindakan tepat sebelumnya, dan catat nilai untuk artefak Output.

6. Pilih Edit.
7. Segera setelah tahap Sumber, pilih Tam bah tahap.

> **Note**
>
> Prosedur ini menunjukkan cara menambahkan tahap pengujian segera setelah tahap Sumber ke pipeline Anda. Untuk menambahkan tindakan pengujian ke tahap yang ada, pilih Edit tahap di tahap, lalu lewati ke langkah 8 dari prosedur ini. Untuk menambahkan tahap pengujian di tempat lain, pilih Tambahkan tahap di tempat yang diinginkan.



8. Untuk Nama pang gung, masukkan nama tahap pengujian (misalnya,**`Test`**). Jika Anda memilih nama yang berbeda, gunakan selama prosedur ini.
9. Pada tahap yang dipilih, pilih Tam bah tindakan.

> **Note**
>
> Prosedur ini menunjukkan cara menambahkan tindakan pengujian dalam tahap pengujian. Untuk menambahkan tindakan pengujian di tempat lain, pilih Tambahkan tindakan di tempat yang diinginkan. Anda mungkin harus terlebih dahulu memilih Edit di tahap yang ada di mana Anda ingin menambahkan tindakan pengujian.

10. Di Edit tindakan, untuk Nama tindakan, masukkan nama untuk tindakan (misalnya,**`Test`**). Jika Anda memilih nama yang berbeda, gunakan selama prosedur ini.
11. Untuk Penye dia tindakan, di bawah Uji, pilih CodeBuild.
12. Jika Anda sudah memiliki proyek build yang ingin Anda gunakan, untuk nama Proyek, pilih nama proyek build dan lewati ke langkah berikutnya dalam prosedur ini.

    Jika Anda perlu membuat proyek CodeBuild build baru, ikuti instruksi Buat proyek build (konsol) dan kembali ke prosedur ini.

    > **Important**
    >
    > Jika Anda mengaktifkan webhook untuk CodeBuild proyek, dan proyek digunakan sebagai langkah membangun CodePipeline, maka dua build identik dibuat untuk setiap komit. Satu build dipicu melalui webhook dan satu melalui CodePipeline. Karena penagihan didasarkan per build, Anda akan ditagih untuk kedua build. Oleh karena itu, jika Anda menggunakan CodePipeline, kami sarankan Anda menonaktifkan webhook di CodeBuild. Di CodeBuild konsol, kosongkan kotak Webhook. Untuk informasi selengkapnya, lihat Mengubah setelan proyek build (konsol)

13. Untuk artefak masukan, pilih nilai untuk artefak keluaran yang Anda catat sebelumnya dalam prosedur ini.
14. (Opsional) Jika Anda ingin tindakan pengujian menghasilkan artefak keluaran, dan Anda mengatur buildspec sesuai, maka untuk artefak keluaran, masukkan nilai yang ingin Anda tetapkan ke artefak keluaran.
15. Pilih Simpan.
16. Pilih Perubahan rilis.
17. Setelah pipa berjalan dengan sukses, Anda bisa mendapatkan hasil tes. Pada tahap U ji pipa, pilih CodeBuild hyperlink untuk membuka halaman proyek build terkait di CodeBuild konsol.
18. Pada halaman proyek build, di Ri wayat build, pilih hyperlink Build run.
19. Pada halaman build run, di Log build, pilih hyperlink Lihat seluruh log untuk membuka log build di CloudWatch konsol Amazon.
20. Gulir melalui log build untuk melihat hasil pengujian.

Gunakan AWS CodeBuild dengan Codecov

Codecov adalah alat yang mengukur cakupan pengujian kode Anda. Codecov mengidentifikasi metode dan pernyataan mana dalam kode Anda yang tidak diuji. Gunakan hasil untuk menentukan di mana menulis tes untuk meningkatkan kualitas kode Anda. Codecov tersedia untuk tiga repositori sumber yang didukung oleh CodeBuild: GitHub, GitHub Enterprise Server, dan Bitbucket. Jika proyek build Anda menggunakan GitHub Enterprise Server, Anda harus menggunakan Codecov Enterprise.

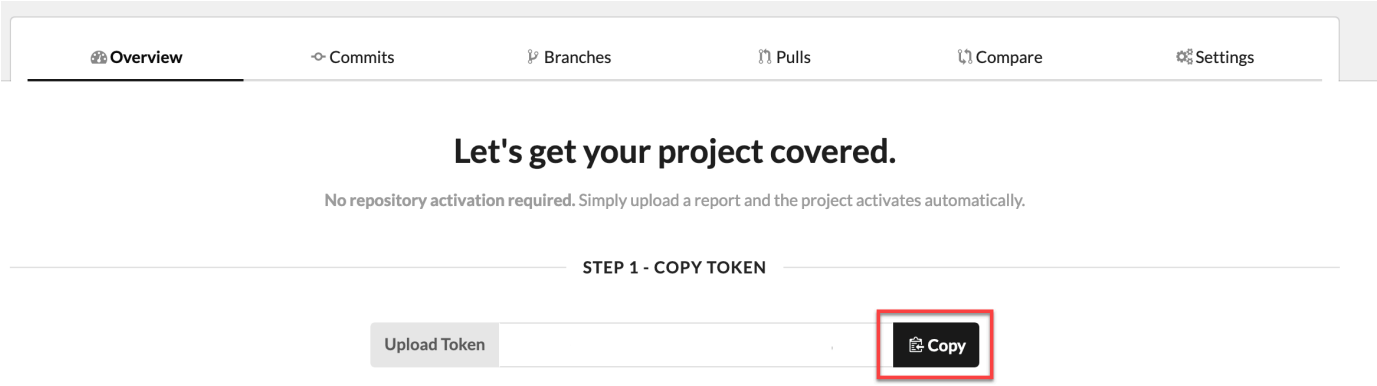
Ketika Anda menjalankan build CodeBuild proyek yang terintegrasi dengan Codecov, Codecov melaporkan bahwa menganalisis kode di repositori Anda diunggah ke Codecov. Log build menyertakan tautan ke laporan. Contoh ini menunjukkan cara mengintegrasikan Python dan proyek build Java dengan Codecov. Untuk daftar bahasa yang didukung oleh Codecov, lihat Bahasa yang didukung Codecov di situs web Codecov.

Integrasikan Codecov ke dalam proyek build

Gunakan prosedur berikut untuk mengintegrasikan Codecov ke dalam proyek build.

Untuk mengintegrasikan Codecov dengan proyek build Anda

1. Buka https://codecov.io/signup dan daftar untuk repositori sumber GitHub atau Bitbucket. Jika Anda menggunakan GitHub Enterprise, lihat Codecov Enterprise di situs web Codecov.
2. Di Codecov, tambahkan repositori yang Anda inginkan cakupannya.
3. Saat informasi token ditampilkan, pilih Sal in.



4. Tambahkan token yang disalin sebagai variabel lingkungan bernama `CODECOV_TOKEN` ke proyek build Anda. Untuk informasi selengkapnya, lihat Mengubah setelan proyek build (konsol).
5. Buat file teks bernama `my_script.sh` di repositori Anda. Masukkan yang berikut ke dalam file:

```
#/bin/bash
```

```
bash <(curl -s https://codecov.io/bash) -t $CODECOV_TOKEN
```

6. Pilih tab Python atau Java, sesuai untuk penggunaan proyek build Anda, dan ikuti langkah-langkah ini.

   Java

   1. Tambahkan JaCoCo plugin berikut ke `pom.xml` dalam repositori Anda.

```
<build>
  <plugins>
    <plugin>
      <groupId>org.jacoco</groupId>
      <artifactId>jacoco-maven-plugin</artifactId>
      <version>0.8.2</version>
      <executions>
          <execution>
              <goals>
                  <goal>prepare-agent</goal>
              </goals>
          </execution>
          <execution>
              <id>report</id>
              <phase>test</phase>
              <goals>
                  <goal>report</goal>
              </goals>
          </execution>
      </executions>
    </plugin>
  </plugins>
</build>
```

   2. Masukkan perintah berikut di file buildspec Anda. Untuk informasi selengkapnya, lihat Sintaks Buildspec.

```
build:
  - mvn test -f pom.xml -fn
postbuild:
  - echo 'Connect to CodeCov'
  - bash my_script.sh
```

Python

Masukkan perintah berikut di file buildspec Anda. Untuk informasi selengkapnya, lihat Sintaks Buildspec.

```
build:
  - pip install coverage
  - coverage run -m unittest discover
postbuild:
  - echo 'Connect to CodeCov'
  - bash my_script.sh
```

7. Jalankan build proyek build Anda. Tautan ke laporan Codecov yang dihasilkan untuk proyek Anda muncul di log build Anda. Gunakan tautan untuk melihat laporan Codecov. Untuk informasi selengkapnya, lihat Jalankan AWS CodeBuild build secara manual dan Log AWS CodeBuild Panggilan API dengan AWS CloudTrail. Informasi Codecov dalam log build terlihat seperti berikut:

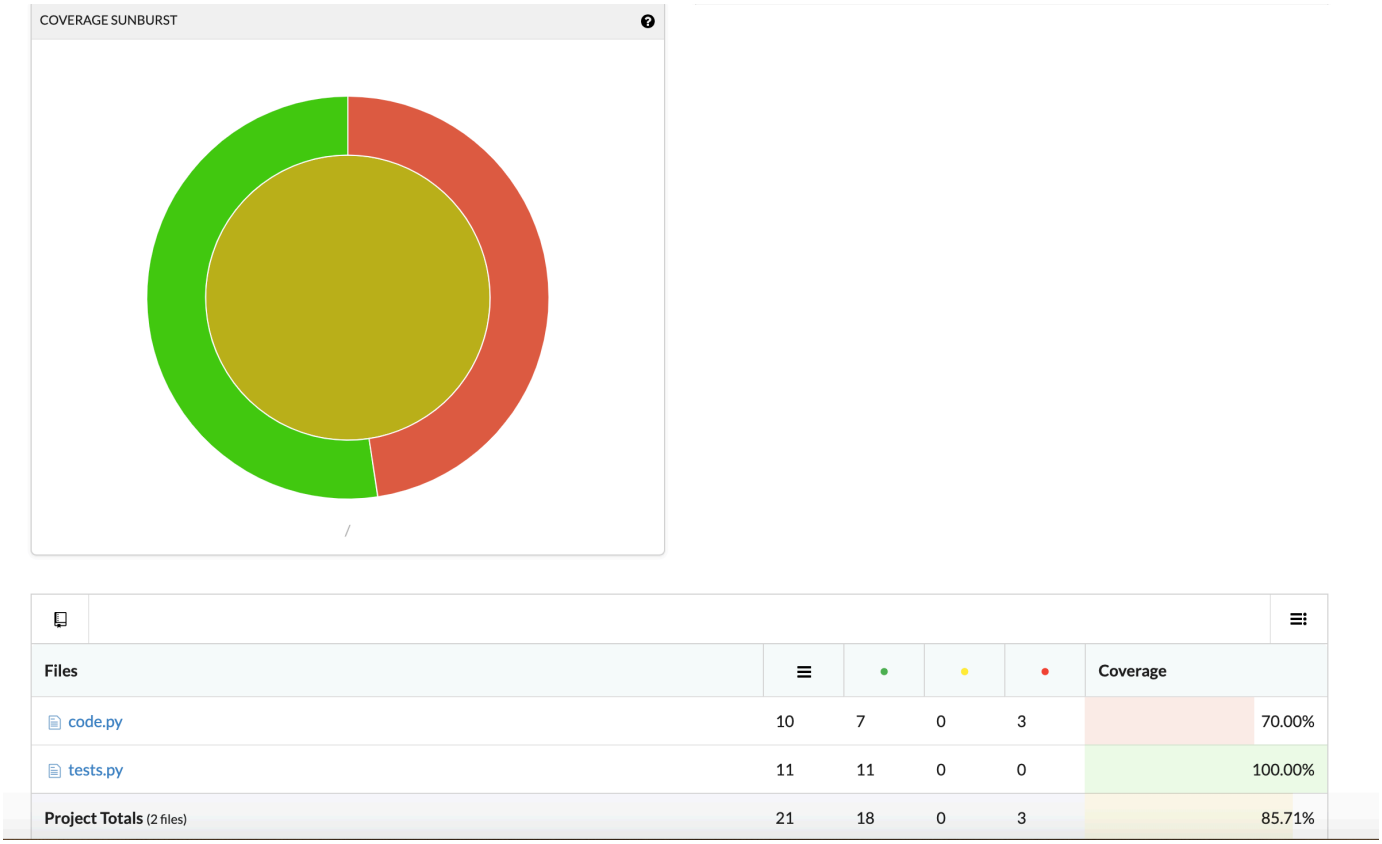
```
[Container] 2020/03/09 16:31:04 Running command bash my_script.sh


  _____          _
 / ____|        | |
| |     ___   __| | ___  ___ _____   __
| |    / _ \ / _` |/ _ \/ __/ _ \ \ / /
| |___| (_) | (_| |  __/ (_| (_) \ V /
 \_____\___/ \__,_|\___|\___\___/ \_/
                              Bash-20200303-bc4d7e6

·[0;90m==>·[0m AWS CodeBuild detected.
... The full list of Codecov log entries has been omitted for brevity ...
    .
    ·[0;32m->·[0m View reports at ·[0;36mhttps://codecov.io/github/user/test_py/
commit/commit-id·[0m

[Container] 2020/03/09 16:31:07 Phase complete: POST_BUILD State: SUCCEEDED
```

Laporan terlihat seperti berikut:



Gunakan AWS CodeBuild dengan Jenkins

Anda dapat menggunakan plugin Jenkins AWS CodeBuild untuk berintegrasi CodeBuild dengan pekerjaan membangun Jenkins Anda. Alih-alih mengirim pekerjaan build Anda ke node build Jenkins, Anda menggunakan plugin untuk mengirim pekerjaan build Anda ke CodeBuild. Ini menghilangkan kebutuhan bagi Anda untuk menyediakan, mengkonfigurasi, dan mengelola node build Jenkins.

Topik

- Siapkan Jenkins
- Pasang plugin
- Gunakan plugin

Siapkan Jenkins

Untuk informasi tentang mengatur Jenkins dengan AWS CodeBuild plugin, dan untuk mengunduh kode sumber plugin, lihat https://github.com/awslabs/aws-codebuild-jenkins-plugin.

Pasang plugin

Jika Anda sudah menyiapkan server Jenkins dan hanya ingin menginstal AWS CodeBuild plugin, pada instance Jenkins Anda, di Pengelola Plugin, cari. **`CodeBuild Plugin for Jenkins`**

Gunakan plugin

Untuk menggunakan AWS CodeBuild dengan sumber dari luar VPC

1. Buat proyek di CodeBuild konsol. Untuk informasi selengkapnya, lihat Buat proyek build (konsol).
   - Pilih Wil AWS ayah tempat Anda ingin menjalankan build.
   - (Opsional) Tetapkan konfigurasi Amazon VPC untuk mengizinkan wadah CodeBuild build mengakses sumber daya di VPC Anda.
   - Tuliskan nama proyek Anda. Anda membutuhkannya di langkah 3.
   - (Opsional) Jika repositori sumber Anda tidak didukung secara native oleh CodeBuild, Anda dapat menetapkan Amazon S3 sebagai jenis sumber masukan untuk proyek Anda.
2. Di konsol IAM, buat pengguna untuk digunakan oleh plugin Jenkins.
   - Saat Anda membuat kredenSIAL untuk pengguna, pilih Ak ses Terprogram.
   - Buat kebijakan yang mirip dengan berikut ini, lalu lampirkan kebijakan tersebut ke pengguna Anda.

     JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Resource": [
                "arn:aws:logs:us-east-1:111122223333:log-group:/aws/
codebuild/{{projectName}}:*"
            ],
            "Action": [
                "logs:GetLogEvents"
            ]
        },
        {
            "Effect": "Allow",
```

```
            "Resource": [
                "arn:aws:s3:::{{inputBucket}}"
            ],
            "Action": [
                "s3:GetBucketVersioning"
            ]
        },
        {
            "Effect": "Allow",
            "Resource": [
                "arn:aws:s3:::{{inputBucket}}/{{inputObject}}"
            ],
            "Action": [
                "s3:PutObject"
            ]
        },
        {
            "Effect": "Allow",
            "Resource": [
                "arn:aws:s3:::{{outputBucket}}/*"
            ],
            "Action": [
                "s3:GetObject"
            ]
        },
        {
            "Effect": "Allow",
            "Resource": [
                "arn:aws:codebuild:us-east-1:111122223333:project/
{{projectName}}"
            ],
            "Action": [
                "codebuild:StartBuild",
                "codebuild:BatchGetBuilds",
                "codebuild:BatchGetProjects"
            ]
        }
    ]
}
```

3. Buat proyek gaya bebas di Jenkins.

   - Pada halaman Konfigurasi, pilih Tambahkan langkah build, lalu pilih J alankan build pada CodeBuild.

- Konfigurasikan langkah pembuatan Anda.
  - Berikan nilai untuk Wilayah, Kreden si, dan Nama Proyek.
  - Pilih Gunakan sumber proyek.
  - Simpan konfigurasi dan jalankan build dari Jenkins.

4. Untuk Manajemen Kode Sumber, pilih bagaimana Anda ingin mengambil sumber Anda. Anda mungkin perlu menginstal GitHub plugin (atau plugin Jenkins untuk penyedia repositori sumber Anda) di server Jenkins Anda.

   - Pada halaman Konfigurasi, pilih Tambahkan langkah build, lalu pilih J alankan build pada AWS CodeBuild.
   - Konfigurasikan langkah pembuatan Anda.
     - Berikan nilai untuk Wilayah, Kreden si, dan Nama Proyek.
     - Pilih Gunakan sumber Jenkins.
     - Simpan konfigurasi dan jalankan build dari Jenkins.

Untuk menggunakan AWS CodeBuild plugin dengan plugin pipa Jenkins

- Pada halaman proyek pipa Jenkins Anda, gunakan generator cuplikan untuk menghasilkan skrip pipeline yang ditambahkan CodeBuild sebagai langkah dalam pipeline Anda. Ini harus menghasilkan skrip yang mirip dengan ini:

```
awsCodeBuild projectName: 'project', credentialsType: 'keys', region: 'us-west-2',
 sourceControlType: 'jenkins'
```

Gunakan AWS CodeBuild dengan aplikasi tanpa server

The AWS Serverless Application Model (AWS SAM) adalah kerangka kerja sumber terbuka untuk membangun aplikasi tanpa server. Untuk informasi selengkapnya, lihat repositori model aplikasi AWS tanpa server di. GitHub

Anda dapat AWS CodeBuild menggunakan paket dan menyebarkan aplikasi tanpa server yang mengikuti standar. AWS SAM Untuk langkah penerapan, CodeBuild bisa menggunakan AWS CloudFormation. Untuk mengotomatiskan pembangunan dan penyebaran aplikasi tanpa server dengan CodeBuild dan CloudFormation, Anda dapat menggunakannya. AWS CodePipeline

Untuk informasi selengkapnya, lihat Menerapkan Aplikasi Tanpa Server di Panduan Pengembang.AWS Serverless Application Model

Sumber daya terkait

- Untuk informasi tentang memulai AWS CodeBuild, lihatMemulai dengan AWS CodeBuild menggunakan konsol.
- Untuk informasi tentang pemecahan masalah di CodeBuild, lihat. Pemecahan Masalah AWS CodeBuild
- Untuk informasi tentang kuota di CodeBuild, lihatKuota untuk AWS CodeBuild.

Pemberitahuan pihak ketiga untuk AWS CodeBuild untuk Windows

Ketika Anda menggunakan CodeBuild untuk build Windows, Anda memiliki opsi untuk menggunakan beberapa paket dan modul pihak ketiga untuk memungkinkan aplikasi buatan Anda berjalan pada sistem operasi Microsoft Windows dan untuk berinteraksi dengan beberapa produk pihak ketiga. Daftar berikut berisi ketentuan hukum pihak ketiga yang berlaku yang mengatur penggunaan paket dan modul pihak ketiga yang ditentukan oleh Anda.

Topik

- 1) gambar dasar Docker — windowsservercore
- 2) gambar Docker berbasis windows-choco
- 3) gambar Docker berbasis windows-git - versi 2.16.2
- 4) gambar Docker berbasis windows-microsoft-build-tools - versi 15.0.26320.2
- 5) gambar Docker berbasis windows-nuget.commandline -- versi 4.5.1
- 7) gambar Docker berbasis windows-netfx-4.6.2-devpack
- 8) gambar Docker berbasis windows-visualfsharptools, v 4.0
- 9) gambar Docker berbasis windows-netfx-pcl-rakitan referensi-4.6
- 10) gambar Docker berbasis windows-visualcppbuildtools v 14.0.25420.1
- 11) gambar Docker berbasis windows-microsoft-windows-netfx3-ondemand-package.cab
- 12) gambar Docker berbasis windows-dotnet-sdk

1) gambar dasar Docker — windowsservercore

(ketentuan lisensi tersedia di: https://hub.docker.com/_/microsoft-windows-servercore)

Lisensi: Dengan meminta dan menggunakan Gambar OS Container ini untuk wadah Windows, Anda mengakui, memahami, dan menyetujui Ketentuan Lisensi Tambahan berikut:

KETENTUAN LISENSI TAMBAHAN PERANGKAT LUNAK MICROSOFT

GAMBAR KONTAINER OS

Microsoft Corporation (atau berdasarkan tempat tinggal Anda, salah satu afiliasinya) (dirujuk sebagai “kami,” “kami,” atau “Microsoft”) melisensikan suplemen Gambar OS Container ini kepada Anda (“Tambahan”). Anda dilisensikan untuk menggunakan Tambahan ini bersama dengan perangkat lunak sistem operasi host yang mendasarinya (“Perangkat Lunak Host”) semata-mata untuk membantu menjalankan fitur kontainer di Perangkat Lunak Host. Ketentuan lisensi Perangkat Lunak Host berlaku untuk penggunaan Tambahan oleh Anda. Anda tidak boleh menggunakannya jika Anda tidak memiliki lisensi untuk Perangkat Lunak Host. Anda dapat menggunakan Suplemen ini dengan setiap salinan Perangkat Lunak Host yang berlisensi sah.

PERSYARATAN LISENSI TAMBAHAN AND/OR HAK PENGGUNAAN

Penggunaan Anda atas Suplemen sebagaimana ditentukan dalam paragraf sebelumnya dapat mengakibatkan pembuatan atau modifikasi gambar wadah (“Gambar Kontainer”) yang mencakup komponen Suplemen tertentu. Untuk kejelasan, Container Image terpisah dan berbeda dari mesin virtual atau gambar alat virtual. Sesuai dengan ketentuan lisensi ini, kami memberi Anda hak terbatas untuk mendistribusikan kembali komponen Suplemen tersebut dalam kondisi berikut:

(i) Anda dapat menggunakan komponen Suplemen hanya seperti yang digunakan dalam, dan sebagai bagian dari Gambar Kontainer Anda,

(ii) Anda dapat menggunakan komponen Suplemen tersebut dalam Gambar Kontainer Anda selama Anda memiliki fungsionalitas utama yang signifikan dalam Gambar Kontainer Anda yang secara material terpisah dan berbeda dari Suplemen; dan

(iii) Anda setuju untuk memasukkan persyaratan lisensi ini (atau persyaratan serupa yang diperlukan oleh kami atau hoster) dengan Gambar Kontainer Anda untuk melisensikan kemungkinan penggunaan komponen Tambahan oleh pengguna akhir Anda dengan benar.

Kami menyimpan semua hak lain yang tidak secara tegas diberikan di sini.

Dengan menggunakan Suplemen ini, Anda menerima persyaratan ini. Jika Anda tidak menerimanya, jangan gunakan Suplemen ini.

Sebagai bagian dari Ketentuan Lisensi Tambahan untuk Container OS Image untuk wadah Windows ini, Anda juga tunduk pada persyaratan lisensi perangkat lunak host Windows Server yang mendasarinya, yang terletak di: https://www.microsoft.com/en-us/useterms.

2) gambar Docker berbasis windows-choco

(ketentuan lisensi tersedia di: https://github.com/chocolatey/choco/blob/master/LICENSE)

Hak Cipta 2011 - RealDimensions Perangkat Lunak Saat Ini, LLC

Dilisensikan di bawah Lisensi Apache, versi 2.0 (“Lisensi”); Anda tidak boleh menggunakan file-file ini kecuali sesuai dengan Lisensi. Anda dapat memperoleh salinan Lisensi di

http://www.apache.org/licenses/LICENSE-2.0

Kecuali diwajibkan oleh hukum yang berlaku atau sebagaimana disepakati secara tertulis, perangkat lunak yang didistribusikan di bawah Lisensi didistribusikan berdasarkan “SEBAGAIMANA ADANYA”, TANPA JAMINAN ATAU KETENTUAN DALAM BENTUK APA PUN, baik tersurat maupun tersirat. Lihat Lisensi untuk bahasa tertentu yang mengatur izin dan batasan di bawah Lisensi.

3) gambar Docker berbasis windows-git - versi 2.16.2

(ketentuan lisensi tersedia di: https://chocolatey.org/packages/git/2.16.2)

Dilisensikan di bawah Lisensi Publik Umum GNU, versi 2, tersedia di: https://www.gnu.org/licenses/old-licenses/gpl-2.0.html.

4) gambar Docker berbasis windows-microsoft-build-tools - versi 15.0.26320.2

(ketentuan lisensi tersedia di: https://www.visualstudio.com/license-terms/mt171552/)

EKSTENSI MICROSOFT VISUAL STUDIO 2015, VISUAL STUDIO SHELLS, dan C ++ REDISTRIBUTABLE

Ketentuan lisensi ini merupakan perjanjian antara Microsoft Corporation (atau berdasarkan tempat tinggal Anda, salah satu afiliasinya) dan Anda. Mereka berlaku untuk perangkat lunak yang disebutkan di atas. Ketentuan ini juga berlaku untuk layanan atau pembaruan Microsoft untuk perangkat lunak, kecuali sejauh yang memiliki persyaratan tambahan.

JIKA ANDA MEMATUHI PERSYARATAN LISENSI INI, ANDA MEMILIKI HAK DI BAWAH INI.

1. HAK INSTALASI DAN PENGGUNAAN. Anda dapat menginstal dan menggunakan sejumlah salinan perangkat lunak.
2. ISTILAH UNTUK KOMPONEN TERTENTU.
   a. Utilitas. Perangkat lunak mungkin berisi beberapa item pada Daftar Utilitas di https://docs.microsoft.com/en-us/visualstudio/productinfo/2015-redistribution-vs. Anda dapat menyalin dan menginstal item tersebut, jika disertakan dengan perangkat lunak, ke mesin milik Anda atau pihak ketiga lainnya, untuk men-debug dan menerapkan aplikasi dan database Anda yang Anda kembangkan dengan perangkat lunak. Harap dicatat bahwa Utilitas dirancang untuk penggunaan sementara, bahwa Microsoft mungkin tidak dapat menambal atau memperbarui Utilitas secara terpisah dari perangkat lunak lainnya, dan bahwa beberapa Utilitas menurut sifatnya memungkinkan orang lain untuk mengakses mesin tempat mereka diinstal. Akibatnya, Anda harus menghapus semua Utilitas yang telah Anda instal setelah Anda selesai men-debug atau menerapkan aplikasi dan database Anda. Microsoft tidak bertanggung jawab atas penggunaan pihak ketiga atau akses Utilitas yang Anda instal pada mesin apa pun.
   b. Platform Microsoft. Perangkat lunak ini dapat mencakup komponen dari Microsoft Windows; Microsoft Windows Server; Microsoft SQL Server; Microsoft Exchange; Microsoft Office; dan Microsoft SharePoint. Komponen-komponen ini diatur oleh perjanjian terpisah dan kebijakan dukungan produk mereka sendiri, seperti yang dijelaskan dalam ketentuan lisensi yang ditemukan di direktori instalasi untuk komponen tersebut atau di folder “Lisensi” yang menyertainya perangkat lunak.
   c. Komponen Pihak Ketiga. Perangkat lunak ini dapat mencakup komponen pihak ketiga dengan pemberitahuan hukum terpisah atau diatur oleh perjanjian lain, seperti yang dijelaskan dalam ThirdPartyNotices file yang menyertainya perangkat lunak. Bahkan jika komponen tersebut diatur oleh perjanjian lain, penafian dan batasan serta pengecualian kerusakan di bawah ini juga berlaku. Perangkat lunak ini juga dapat mencakup komponen yang dilisensikan di bawah lisensi sumber terbuka dengan kewajiban ketersediaan kode sumber. Salinan lisensi tersebut, jika berlaku, disertakan dalam ThirdPartyNotices file. Anda dapat memperoleh kode sumber ini dari kami, jika dan sebagaimana diperlukan berdasarkan lisensi sumber terbuka yang relevan,

dengan mengirimkan wesel atau cek sebesar $5,00 ke: Tim Kepatuhan Kode Sumber, Microsoft Corporation, 1 Microsoft Way, Redmond, WA 98052. Silakan tulis kode sumber untuk satu atau lebih komponen yang tercantum di bawah ini di baris memo pembayaran Anda:

- Alat Jarak Jauh untuk Visual Studio 2015;
- Profiler Mandiri untuk Visual Studio 2015;
- IntelliTraceCollector untuk Visual Studio 2015;
- Microsoft VC ++ Dapat Didistribusikan Ulang 2015;
- Perpustakaan MFC Multibyte untuk Visual Studio 2015;
- Alat Pembuatan Microsoft 2015;
- Umpan Balik Klien;
- Visual Studio 2015 Shell Terintegrasi; atau
- Visual Studio 2015 Shell Terisolasi.

Kami juga dapat membuat salinan kode sumber tersedia di http://thirdpartysource.microsoft.com.

3. DATA. Perangkat lunak dapat mengumpulkan informasi tentang Anda dan penggunaan perangkat lunak oleh Anda, dan mengirimkannya ke Microsoft. Microsoft dapat menggunakan informasi ini untuk menyediakan layanan dan meningkatkan produk dan layanan kami. Anda dapat memilih keluar dari banyak skenario ini, tetapi tidak semua, seperti yang dijelaskan dalam dokumentasi produk. Ada juga beberapa fitur dalam perangkat lunak yang memungkinkan Anda mengumpulkan data dari pengguna aplikasi Anda. Jika Anda menggunakan fitur-fitur ini untuk mengaktifkan pengumpulan data dalam aplikasi Anda, Anda harus mematuhi hukum yang berlaku, termasuk memberikan pemberitahuan yang sesuai kepada pengguna aplikasi Anda. Anda dapat mempelajari lebih lanjut tentang pengumpulan dan penggunaan data dalam dokumentasi bantuan dan pernyataan privasi di https://privacy.microsoft.com/en-us/privacystatement. Penggunaan perangkat lunak oleh Anda beroperasi sebagai persetujuan Anda untuk praktik ini.
4. RUANG LINGKUP LISENSI. Perangkat lunak ini dilisensikan, tidak dijual. Perjanjian ini hanya memberi Anda beberapa hak untuk menggunakan perangkat lunak. Microsoft memiliki semua hak lainnya. Kecuali hukum yang berlaku memberi Anda lebih banyak hak terlepas dari batasan ini, Anda dapat menggunakan perangkat lunak hanya sebagaimana diizinkan secara tegas dalam perjanjian ini. Dengan demikian, Anda harus mematuhi batasan teknis apa pun dalam perangkat lunak yang hanya memungkinkan Anda untuk menggunakannya dengan cara tertentu. Anda mungkin tidak
   - mengatasi keterbatasan teknis apa pun dalam perangkat lunak;

- merekayasa balik, mendekompilasi atau membongkar perangkat lunak, atau mencoba melakukannya, kecuali dan hanya sejauh yang diperlukan oleh ketentuan lisensi pihak ketiga yang mengatur penggunaan komponen sumber terbuka tertentu yang mungkin disertakan dengan perangkat lunak;
- menghapus, meminimalkan, memblokir, atau memodifikasi pemberitahuan apa pun dari Microsoft atau pemasoknya dalam perangkat lunak;
- menggunakan perangkat lunak dengan cara apa pun yang melanggar hukum; atau
- berbagi, menerbitkan, menyewakan atau menyewakan perangkat lunak, atau menyediakan perangkat lunak sebagai hosting yang berdiri sendiri sebagai solusi untuk digunakan orang lain.

5. PEMBATASAN EKSPOR. Anda harus mematuhi semua undang-undang dan peraturan ekspor domestik dan internasional yang berlaku untuk perangkat lunak, yang mencakup pembatasan tujuan, pengguna akhir, dan penggunaan akhir. Untuk informasi lebih lanjut tentang pembatasan ekspor, kunjungi (alias. ms/exporting).
6. LAYANAN DUKUNGAN. Karena perangkat lunak ini “apa adanya,” kami mungkin tidak menyediakan layanan dukungan untuk itu.
7. KESELURUHAN PERJANJI AN. Perjanjian ini, dan ketentuan untuk suplemen, pembaruan, Internet-based layanan, dan layanan dukungan yang Anda gunakan, adalah keseluruhan perjanjian untuk perangkat lunak dan layanan dukungan.
8. HUKUM YANG BERLAKU. Jika Anda memperoleh perangkat lunak di Amerika Serikat, hukum Washington berlaku untuk interpretasi dan klaim pelanggaran perjanjian ini, dan hukum negara bagian tempat Anda tinggal berlaku untuk semua klaim lainnya. Jika Anda memperoleh perangkat lunak di negara lain mana pun, hukumnya berlaku.
9. HAK KONSUMEN; VARIASI REGIONAL. Perjanjian ini menjelaskan hak-hak hukum tertentu. Anda mungkin memiliki hak lain, termasuk hak konsumen, berdasarkan hukum negara bagian atau negara Anda. Terpisah dan terlepas dari hubungan Anda dengan Microsoft, Anda mungkin juga memiliki hak sehubungan dengan pihak tempat Anda memperoleh perangkat lunak. Perjanjian ini tidak mengubah hak-hak lain tersebut jika hukum negara bagian atau negara Anda tidak mengizinkannya untuk melakukannya. Misalnya, jika Anda memperoleh perangkat lunak di salah satu wilayah di bawah ini, atau hukum negara wajib berlaku, maka ketentuan berikut berlaku untuk Anda:
   a. Australia. Anda memiliki jaminan hukum berdasarkan Undang-Undang Konsumen Australia dan tidak ada dalam perjanjian ini yang dimaksudkan untuk mempengaruhi hak-hak tersebut.
   b. Kanada. Jika Anda memperoleh perangkat lunak ini di Kanada, Anda dapat berhenti menerima pembaruan dengan mematikan fitur pembaruan otomatis, memutuskan sambungan perangkat

Anda dari Internet (jika dan ketika Anda terhubung kembali ke Internet, bagaimanapun, perangkat lunak akan melanjutkan memeriksa dan menginstal pembaruan), atau menghapus instalasi perangkat lunak. Dokumentasi produk, jika ada, juga dapat menentukan cara mematikan pembaruan untuk perangkat atau perangkat lunak spesifik Anda.

c. Jerman dan Austria.

i. Garansi. Perangkat lunak berlisensi dengan benar akan bekerja secara substansif seperti yang dijelaskan dalam materi Microsoft apa pun yang menyertainya perangkat lunak. Namun, Microsoft tidak memberikan jaminan kontrak sehubungan dengan perangkat lunak berlisensi.

ii. Batasan Tanggung Jawab. Dalam hal tindakan yang disengaja, kelalaian berat, klaim berdasarkan Undang-Undang Tanggung Jawab Produk, serta, dalam kasus kematian atau cedera pribadi atau fisik, Microsoft bertanggung jawab sesuai dengan undang-undang perundang-undang.Tunduk pada klausul (ii) sebelumnya, Microsoft hanya akan bertanggung jawab atas kelalaian ringan jika Microsoft melanggar kewajiban kontrak material tersebut, yang pemenuhannya memfasilitasi pelaksanaan perjanjian ini, yang pelanggarannya akan membahayakan tujuan perjanjian ini dan kepatuhan yang dapat selalu dipercaya oleh suatu pihak (jadi- disebut “kewajiban kardinal”). Dalam kasus kelalaian ringan lainnya, Microsoft tidak akan bertanggung jawab atas kelalaian ringan.

10. PENAFIAN GARANSI. PERANGKAT LUNAK INI BERLISENSI "AS-IS.” ANDA MENANGGUNG RISIKO MENGGUNAKANNYA. MICROSOFT TIDAK MEMBERIKAN JAMINAN, JAMINAN, ATAU KETENTUAN YANG JELAS. SEJAUH DIIZINKAN BERDASARKAN UNDANG-UNDANG SETEMPAT ANDA, MICROSOFT MENGECUALIKAN JAMINAN TERSIRAT TENTANG KELAYAKAN UNTUK DIPERDAGANGKAN, KESESUAIAN UNTUK TUJUAN TERTENTU, DAN. NON-INFRINGEMENT

11. PEMBATASAN DAN PENGECUALIAN KERUSAKAN. ANDA DAPAT MEMULIHKAN DARI MICROSOFT DAN PEMASOKNYA HANYA KERUSAKAN LANGSUNG HINGGA U.S. $5,00. ANDA TIDAK DAPAT MEMULIHKAN KERUSAKAN LAINNYA, TERMASUK KERUSAKAN KONSEKUENSIAL, KEHILANGAN KEUNTUNGAN, KERUSAKAN KHUSUS, TIDAK LANGSUNG ATAU INSIDENTAL. Batasan ini berlaku untuk (a) apa pun yang terkait dengan perangkat lunak, layanan, konten (termasuk kode) di situs Internet pihak ketiga, atau aplikasi pihak ketiga; dan (b) klaim atas pelanggaran kontrak, pelanggaran garansi, jaminan atau kondisi, tanggung jawab yang ketat, kelalaian, atau pelanggaran lainnya sejauh diizinkan oleh hukum yang berlaku.

Ini juga berlaku bahkan jika Microsoft tahu atau seharusnya tahu tentang kemungkinan kerusakan. Batasan atau pengecualian di atas mungkin tidak berlaku untuk Anda karena negara Anda mungkin tidak mengizinkan pengecualian atau pembatasan kerusakan insidental, konsekuensial, atau lainnya.

ID EULA: VS ShellsRedist 2015_Update3_ _ <ENU>

5) gambar Docker berbasis windows-nuget.commandline -- versi 4.5.1

(ketentuan lisensi tersedia di: https://github.com/NuGet/Home/blob/dev/LICENSE.txt)

Hak Cipta (c) .NET Foundation. Semua hak dilindungi undang-undang.

Dilisensikan di bawah Lisensi Apache, versi 2.0 (“Lisensi”); Anda tidak boleh menggunakan file-file ini kecuali sesuai dengan Lisensi. Anda dapat memperoleh salinan Lisensi di

http://www.apache.org/licenses/LICENSE-2.0

Kecuali diwajibkan oleh hukum yang berlaku atau sebagaimana disepakati secara tertulis, perangkat lunak yang didistribusikan di bawah Lisensi didistribusikan berdasarkan “SEBAGAIMANA ADANYA”, TANPA JAMINAN ATAU KETENTUAN DALAM BENTUK APA PUN, baik tersurat maupun tersirat. Lihat Lisensi untuk bahasa tertentu yang mengatur izin dan batasan di bawah Lisensi.

7) gambar Docker berbasis windows-netfx-4.6.2-devpack

KETENTUAN LISENSI TAMBAHAN PERANGKAT LUNAK MICROSOFT

.NET FRAMEWORK DAN PAKET BAHASA TERKAIT UNTUK SISTEM OPERASI MICROSOFT WINDOWS

Microsoft Corporation (atau berdasarkan tempat tinggal Anda, salah satu afiliasinya) melisensikan suplemen ini kepada Anda. Jika Anda berlisensi untuk menggunakan perangkat lunak sistem operasi Microsoft Windows (“perangkat lunak”), Anda dapat menggunakan suplemen ini. Anda tidak boleh menggunakannya jika Anda tidak memiliki lisensi untuk perangkat lunak. Anda dapat menggunakan suplemen ini dengan setiap salinan perangkat lunak berlisensi yang sah.

Ketentuan lisensi berikut menjelaskan ketentuan penggunaan tambahan untuk suplemen ini. Ketentuan ini dan ketentuan lisensi untuk perangkat lunak berlaku untuk penggunaan suplemen oleh Anda. Jika ada konflik, ketentuan lisensi tambahan ini berlaku.

DENGAN MENGGUNAKAN SUPLEMEN INI, ANDA MENERIMA KETENTUAN INI. JIKA ANDA TIDAK MENERIMANYA, JANGAN GUNAKAN SUPLEMEN INI.

Jika Anda mematuhi persyaratan lisensi ini, Anda memiliki hak di bawah ini.

1. KODE YANG DAPAT DIDISTRIBUSIKAN. Suplemen terdiri dari Kode yang Dapat Didistribusikan. “Kode yang Dapat Didistribusikan” adalah kode yang diizinkan untuk Anda distribusikan dalam program yang Anda kembangkan jika Anda mematuhi ketentuan di bawah ini.
   a. Hak untuk Menggunakan dan Mendistribusikan.
      - Anda dapat menyalin dan mendistribusikan bentuk kode objek suplemen.
      - Distribusi Pihak Ketiga. Anda dapat mengizinkan distributor program Anda untuk menyalin dan mendistribusikan Kode yang Dapat Didistribusikan sebagai bagian dari program tersebut.
   b. Persyaratan Distribusi. Untuk setiap Kode yang Dapat Didistribusikan yang Anda distribusikan, Anda harus
      - tambahkan fungsionalitas utama yang signifikan untuk itu dalam program Anda;
      - untuk setiap Kode yang Dapat Didistribusikan yang memiliki ekstensi nama file.lib, distribusikan hanya hasil menjalankan Kode yang Dapat Didistribusikan tersebut melalui penghubung dengan program Anda;
      - mendistribusikan Kode yang Dapat Didistribusikan yang disertakan dalam program penyiapan hanya sebagai bagian dari program penyiapan itu tanpa modifikasi;
      - mewajibkan distributor dan pengguna akhir eksternal untuk menyetujui persyaratan yang melindunginya setidaknya sebanyak perjanjian ini;
      - menampilkan pemberitahuan hak cipta yang valid pada program Anda; dan
      - mengganti rugi, membela, dan membebaskan Microsoft dari klaim apa pun, termasuk biaya pengacara, terkait dengan distribusi atau penggunaan program Anda.
   c. Pembatasan Distribusi. Anda mungkin tidak
      - mengubah hak cipta, merek dagang, atau pemberitahuan paten apa pun dalam Kode yang Dapat Didistribusikan;
      - menggunakan merek dagang Microsoft dalam nama program Anda atau dengan cara yang menunjukkan bahwa program Anda berasal dari atau didukung oleh Microsoft;
      - Mendistribusikan Kode yang Dapat Didistribusikan untuk dijalankan pada platform selain platform Windows;
      - menyertakan Kode yang Dapat Didistribusikan dalam program jahat, menipu, atau melanggar hukum; atau

- memodifikasi atau mendistribusikan kode sumber dari Kode yang Dapat Didistribusikan sehingga setiap bagian darinya menjadi tunduk pada Lisensi yang Dikecualikan. Lisensi yang Dikecualikan adalah lisensi yang mensyaratkan, sebagai syarat penggunaan, modifikasi atau distribusi,
  - kode diungkapkan atau didistribusikan dalam bentuk kode sumber; atau
  - orang lain memiliki hak untuk memodifikasinya.

2. LAYANAN DUKUNGAN UNTUK SUPLEMEN. Microsoft menyediakan layanan dukungan untuk perangkat lunak ini seperti yang dijelaskan di www.support.microsoft. com/common/international.aspx.

8) gambar Docker berbasis windows-visualfsharptools, v 4.0

(ketentuan lisensi tersedia di: https://github.com/dotnet/fsharp/blob/main/License.txt)

Hak Cipta (c) Microsoft Corporation. Semua hak dilindungi undang-undang.

Dilisensikan di bawah Lisensi Apache, versi 2.0 (“Lisensi”); Anda tidak boleh menggunakan file-file ini kecuali sesuai dengan Lisensi. Anda dapat memperoleh salinan Lisensi di

http://www.apache.org/licenses/LICENSE-2.0

Kecuali diwajibkan oleh hukum yang berlaku atau sebagaimana disepakati secara tertulis, perangkat lunak yang didistribusikan di bawah Lisensi didistribusikan berdasarkan “SEBAGAIMANA ADANYA”, TANPA JAMINAN ATAU KETENTUAN DALAM BENTUK APA PUN, baik tersurat maupun tersirat. Lihat Lisensi untuk bahasa tertentu yang mengatur izin dan batasan di bawah Lisensi.

9) gambar Docker berbasis windows-netfx-pcl-rakitan referensi-4.6

KETENTUAN LISENSI PERANGKAT LUNAK MICROSOFT

RAKITAN REFERENSI PERPUSTAKAAN KELAS PORTABEL MICROSOFT.NET - 4.6

Ketentuan lisensi ini merupakan perjanjian antara Microsoft Corporation (atau berdasarkan tempat tinggal Anda, salah satu afiliasinya) dan Anda. Silakan baca mereka. Mereka berlaku untuk perangkat lunak yang disebutkan di atas. Ketentuan ini juga berlaku untuk semua Microsoft

- pembaruan,

- suplemen,
- Internet-based layanan, dan
- layanan dukungan

untuk perangkat lunak ini, kecuali persyaratan lain menyertainya. Jika demikian, ketentuan tersebut berlaku.

DENGAN MENGGUNAKAN PERANGKAT LUNAK, ANDA MENERIMA KETENTUAN INI. JIKA ANDA TIDAK MENERIMANYA, JANGAN GUNAKAN PERANGKAT LUNAK.

JIKA ANDA MEMATUHI KETENTUAN LISENSI INI, ANDA MEMILIKI HAK ABADI DI BAWAH INI.

1. HAK INSTALASI DAN PENGGUNAAN. Anda dapat menginstal dan menggunakan sejumlah salinan perangkat lunak untuk merancang, mengembangkan, dan menguji program Anda.
2. PERSYARATAN LISENSI TAMBAHAN AND/OR MENGGUNAKAN HAK.
   a. Kode yang dapat didistribusikan. Anda dapat mendistribusikan perangkat lunak dalam program alat pengembang yang Anda kembangkan, untuk memungkinkan pelanggan program Anda mengembangkan perpustakaan portabel untuk digunakan dengan perangkat atau sistem operasi apa pun, jika Anda mematuhi ketentuan di bawah ini.
      i. Hak untuk Menggunakan dan Mendistribusikan. Perangkat lunak ini adalah “Kode yang Dapat Didistribusikan.”
         - Kode yang dapat didistribusikan. Anda dapat menyalin dan mendistribusikan bentuk kode objek perangkat lunak.
         - Distribusi Pihak Ketiga. Anda dapat mengizinkan distributor program Anda untuk menyalin dan mendistribusikan Kode yang Dapat Didistribusikan sebagai bagian dari program tersebut.
      ii. Persyaratan Distribusi. Untuk setiap Kode yang Dapat Didistribusikan yang Anda distribusikan, Anda harus
         - tambahkan fungsionalitas utama yang signifikan untuk itu dalam program Anda;
         - mewajibkan distributor dan pelanggan Anda untuk menyetujui persyaratan yang melindunginya setidaknya sebanyak perjanjian ini;
         - menampilkan pemberitahuan hak cipta yang valid pada program Anda; dan

- mengganti rugi, membela, dan membebaskan Microsoft dari klaim apa pun, termasuk biaya pengacara, terkait dengan distribusi atau penggunaan program Anda.

iii. Pembatasan Distribusi. Anda mungkin tidak

- mengubah hak cipta, merek dagang, atau pemberitahuan paten apa pun dalam Kode yang Dapat Didistribusikan;
- menggunakan merek dagang Microsoft dalam nama program Anda atau dengan cara yang menunjukkan bahwa program Anda berasal dari atau didukung oleh Microsoft;
- menyertakan Kode yang Dapat Didistribusikan dalam program jahat, menipu, atau melanggar hukum; atau
- memodifikasi atau mendistribusikan Kode yang Dapat Didistribusikan sehingga setiap bagian darinya menjadi tunduk pada Lisensi yang Dikecualikan. Lisensi yang Dikecualikan adalah lisensi yang mensyaratkan, sebagai syarat penggunaan, modifikasi atau distribusi, bahwa
  - kode diungkapkan atau didistribusikan dalam bentuk kode sumber; atau
  - orang lain memiliki hak untuk memodifikasinya.

3. RUANG LINGKUP LISENSI. Perangkat lunak ini dilisensikan, tidak dijual. Perjanjian ini hanya memberi Anda beberapa hak untuk menggunakan perangkat lunak. Microsoft memiliki semua hak lainnya. Kecuali hukum yang berlaku memberi Anda lebih banyak hak terlepas dari batasan ini, Anda dapat menggunakan perangkat lunak hanya sebagaimana diizinkan secara tegas dalam perjanjian ini. Dengan demikian, Anda harus mematuhi batasan teknis apa pun dalam perangkat lunak yang hanya memungkinkan Anda untuk menggunakannya dengan cara tertentu. Anda mungkin tidak
   - mengatasi keterbatasan teknis apa pun dalam perangkat lunak;
   - merekayasa balik, mendekompilasi, atau membongkar perangkat lunak, kecuali dan hanya sejauh hukum yang berlaku secara tegas mengizinkan, terlepas dari batasan ini;
   - menerbitkan perangkat lunak untuk orang lain untuk disalin; atau
   - menyewa, menyewakan atau meminjamkan perangkat lunak.
4. UMPAN BALIK. Anda dapat memberikan umpan balik tentang perangkat lunak. Jika Anda memberikan umpan balik tentang perangkat lunak kepada Microsoft, Anda memberikan kepada Microsoft, tanpa biaya, hak untuk menggunakan, berbagi, dan mengkomersialkan umpan balik Anda dengan cara apa pun dan untuk tujuan apa pun. Anda juga memberikan kepada pihak ketiga, tanpa biaya, hak paten apa pun yang diperlukan untuk produk, teknologi, dan layanan mereka untuk menggunakan atau berinteraksi dengan bagian tertentu dari perangkat lunak atau layanan Microsoft yang menyertakan umpan balik. Anda tidak akan memberikan umpan balik

yang tunduk pada lisensi yang mengharuskan Microsoft melisensikan perangkat lunak atau dokumentasinya kepada pihak ketiga karena kami menyertakan umpan balik Anda di dalamnya. Hak-hak ini bertahan dalam perjanjian ini.

5. TRANSFER KE PIHAK KETIGA. Pengguna pertama perangkat lunak dapat mentransfernya, dan perjanjian ini, langsung ke pihak ketiga. Sebelum transfer, pihak tersebut harus menyetujui bahwa perjanjian ini berlaku untuk transfer dan penggunaan perangkat lunak. Pengguna pertama harus menghapus instalasi perangkat lunak sebelum mentransfernya secara terpisah dari perangkat. Pengguna pertama tidak boleh menyimpan salinan apa pun.
6. PEMBATASAN EKSPOR. Perangkat lunak ini tunduk pada undang-undang dan peraturan ekspor Amerika Serikat. Anda harus mematuhi semua undang-undang dan peraturan ekspor domestik dan internasional yang berlaku untuk perangkat lunak. Undang-undang ini mencakup pembatasan tujuan, pengguna akhir, dan penggunaan akhir. Untuk informasi tambahan, lihat www.microsoft.com/exporting.
7. LAYANAN DUKUNGAN. Karena perangkat lunak ini “apa adanya,” kami mungkin tidak menyediakan layanan dukungan untuk itu.
8. KESELURUHAN PERJANJI AN. Perjanjian ini, dan ketentuan untuk suplemen, pembaruan, Internet-based layanan, dan layanan dukungan yang Anda gunakan, adalah seluruh perjanjian untuk perangkat lunak dan layanan dukungan apa pun yang kami sediakan.
9. HUKUM YANG BERLAKU.
   a. Amerika Serikat. Jika Anda memperoleh perangkat lunak di Amerika Serikat, hukum negara bagian Washington mengatur interpretasi perjanjian ini dan berlaku untuk klaim pelanggarannya, terlepas dari konflik prinsip hukum. Hukum negara tempat Anda tinggal mengatur semua klaim lainnya, termasuk klaim berdasarkan undang-undang perlindungan konsumen negara bagian, undang-undang persaingan tidak sehat, dan gugatan.
   b. Di luar Amerika Serikat. Jika Anda memperoleh perangkat lunak di negara lain mana pun, hukum negara itu berlaku.
10. EFEK HUKUM. Perjanjian ini menjelaskan hak-hak hukum tertentu. Anda mungkin memiliki hak lain berdasarkan hukum negara Anda. Anda juga mungkin memiliki hak sehubungan dengan pihak dari mana Anda memperoleh perangkat lunak. Perjanjian ini tidak mengubah hak Anda berdasarkan hukum negara Anda jika hukum negara Anda tidak mengizinkannya untuk melakukannya.
11. PENAFIAN GARANSI. PERANGKAT LUNAK INI BERLISENSI "AS-IS.” ANDA MENANGGUNG RISIKO MENGGUNAKANNYA. MICROSOFT TIDAK MEMBERIKAN JAMINAN, JAMINAN, ATAU KETENTUAN YANG JELAS. ANDA MUNGKIN MEMILIKI HAK KONSUMEN TAMBAHAN ATAU JAMINAN HUKUM BERDASARKAN UNDANG-UNDANG SETEMPAT YANG TIDAK

DAPAT DIUBAH OLEH PERJANJIAN INI. SEJAUH DIIZINKAN BERDASARKAN UNDANG-UNDANG SETEMPAT ANDA, MICROSOFT MENGECUALIKAN JAMINAN TERSIRAT TENTANG KELAYAKAN UNTUK DIPERDAGANGKAN, KESESUAIAN UNTUK TUJUAN TERTENTU, DAN. NON-INFRINGEMENT

UNTUK AUSTRALIA — ANDA MEMILIKI JAMINAN HUKUM BERDASARKAN UNDANG-UNDANG KONSUMEN AUSTRALIA DAN TIDAK ADA DALAM KETENTUAN INI YANG DIMAKSUDKAN UNTUK MEMENGARUHI HAK-HAK TERSEBUT.

12 PEMBATASAN DAN PENGECUALIAN SOLUSI DAN KERUSAKAN. ANDA DAPAT MEMULIHKAN DARI MICROSOFT DAN PEMASOKNYA HANYA KERUSAKAN LANGSUNG HINGGA U.S. $5,00. ANDA TIDAK DAPAT MEMULIHKAN KERUSAKAN LAINNYA, TERMASUK KERUSAKAN KONSEKUENSIAL, KEHILANGAN KEUNTUNGAN, KERUSAKAN KHUSUS, TIDAK LANGSUNG ATAU INSIDENTAL.

Batasan ini berlaku untuk

- apa pun yang terkait dengan perangkat lunak, layanan, konten (termasuk kode) di situs Internet pihak ketiga, atau program pihak ketiga; dan
- klaim atas pelanggaran kontrak, pelanggaran garansi, jaminan atau kondisi, tanggung jawab yang ketat, kelalaian, atau pelanggaran lainnya sejauh diizinkan oleh hukum yang berlaku.

Ini juga berlaku bahkan jika Microsoft tahu atau seharusnya tahu tentang kemungkinan kerusakan. Batasan atau pengecualian di atas mungkin tidak berlaku untuk Anda karena negara Anda mungkin tidak mengizinkan pengecualian atau pembatasan kerusakan insidental, konsekuensial, atau lainnya.

10) gambar Docker berbasis windows-visualcppbuildtools v 14.0.25420.1

(ketentuan lisensi tersedia di: https://www.visualstudio.com/license-terms/mt644918/)

ALAT PEMBUATAN MICROSOFT VISUAL C++

KETENTUAN LISENSI PERANGKAT LUNAK MICROSOFT

ALAT PEMBUATAN MICROSOFT VISUAL C++

Ketentuan lisensi ini merupakan perjanjian antara Microsoft Corporation (atau berdasarkan tempat tinggal Anda, salah satu afiliasinya) dan Anda. Mereka berlaku untuk perangkat lunak yang

disebutkan di atas. Ketentuan ini juga berlaku untuk layanan atau pembaruan Microsoft untuk perangkat lunak, kecuali sejauh mereka memiliki persyaratan yang berbeda.

JIKA ANDA MEMATUHI KETENTUAN LISENSI INI, ANDA MEMILIKI HAK DI BAWAH INI.

1. HAK INSTALASI DAN PENGGUNAAN.
   a. Satu pengguna dapat menggunakan salinan perangkat lunak untuk mengembangkan dan menguji aplikasi mereka.
2. DATA. Perangkat lunak dapat mengumpulkan informasi tentang Anda dan penggunaan perangkat lunak oleh Anda, dan mengirimkannya ke Microsoft. Microsoft dapat menggunakan informasi ini untuk menyediakan layanan dan meningkatkan produk dan layanan kami. Anda dapat memilih keluar dari banyak skenario ini, tetapi tidak semua, seperti yang dijelaskan dalam dokumentasi produk. Ada juga beberapa fitur dalam perangkat lunak yang memungkinkan Anda mengumpulkan data dari pengguna aplikasi Anda. Jika Anda menggunakan fitur-fitur ini untuk mengaktifkan pengumpulan data dalam aplikasi Anda, Anda harus mematuhi hukum yang berlaku, termasuk memberikan pemberitahuan yang sesuai kepada pengguna aplikasi Anda. Anda dapat mempelajari lebih lanjut tentang pengumpulan dan penggunaan data dalam dokumentasi bantuan dan pernyataan privasi di http://go.microsoft.com/fwlink/? linkID=528096. Penggunaan perangkat lunak oleh Anda beroperasi sebagai persetujuan Anda untuk praktik ini.
3. ISTILAH UNTUK KOMPONEN TERTENTU.
   a. Membangun Server. Perangkat lunak ini mungkin berisi beberapa komponen Build Server yang tercantum dalam BuildServer.TXT file, file and/or apa pun yang tercantum dalam BuildeServer daftar yang terletak mengikuti Ketentuan Lisensi Perangkat Lunak Microsoft ini. Anda dapat menyalin dan menginstal item tersebut, jika disertakan dalam perangkat lunak, ke mesin build Anda. Anda dan orang lain di organisasi Anda dapat menggunakan item ini pada mesin build Anda semata-mata untuk tujuan kompilasi, membangun, memverifikasi, dan mengarsipkan aplikasi Anda atau menjalankan pengujian kualitas atau kinerja sebagai bagian dari proses pembuatan.
   b. Platform Microsoft. Perangkat lunak ini dapat mencakup komponen dari Microsoft Windows; Microsoft Windows Server; Microsoft SQL Server; Microsoft Exchange; Microsoft Office; dan Microsoft SharePoint. Komponen-komponen ini diatur oleh perjanjian terpisah dan kebijakan dukungan produk mereka sendiri, seperti yang dijelaskan dalam ketentuan lisensi yang ditemukan di direktori instalasi untuk komponen tersebut atau di folder “Lisensi” yang menyertainya perangkat lunak.

c. Komponen Pihak Ketiga. Perangkat lunak ini dapat mencakup komponen pihak ketiga dengan pemberitahuan hukum terpisah atau diatur oleh perjanjian lain, seperti yang dijelaskan dalam ThirdPartyNotices file yang menyertainya perangkat lunak. Bahkan jika komponen tersebut diatur oleh perjanjian lain, penafian dan batasan serta pengecualian kerusakan di bawah ini juga berlaku.

d. Manajer Paket. Perangkat lunak ini mungkin termasuk manajer paket, seperti Nuget, yang memberi Anda opsi untuk mengunduh paket perangkat lunak Microsoft dan pihak ketiga lainnya untuk digunakan dengan aplikasi Anda. Paket-paket tersebut berada di bawah lisensi mereka sendiri, dan bukan perjanjian ini. Microsoft tidak mendistribusikan, melisensikan, atau memberikan jaminan apa pun untuk paket pihak ketiga mana pun.

4. RUANG LINGKUP LISENSI. Perangkat lunak ini dilisensikan, tidak dijual. Perjanjian ini hanya memberi Anda beberapa hak untuk menggunakan perangkat lunak. Microsoft memiliki semua hak lainnya. Kecuali hukum yang berlaku memberi Anda lebih banyak hak terlepas dari batasan ini, Anda dapat menggunakan perangkat lunak hanya sebagaimana diizinkan secara tegas dalam perjanjian ini. Dengan demikian, Anda harus mematuhi batasan teknis apa pun dalam perangkat lunak yang hanya memungkinkan Anda untuk menggunakannya dengan cara tertentu. Untuk informasi selengkapnya, lihat https://docs.microsoft.com/en-us/legal/information-protection/software-license-terms #1 -install ation-and-use-rights. Anda mungkin tidak

   - mengatasi keterbatasan teknis apa pun dalam perangkat lunak;
   - merekayasa balik, mendekompilasi atau membongkar perangkat lunak, atau mencoba melakukannya, kecuali dan hanya sejauh yang diperlukan oleh ketentuan lisensi pihak ketiga yang mengatur penggunaan komponen open source tertentu yang mungkin disertakan dengan perangkat lunak;
   - menghapus, meminimalkan, memblokir, atau memodifikasi pemberitahuan apa pun dari Microsoft atau pemasoknya;
   - menggunakan perangkat lunak dengan cara apa pun yang melanggar hukum; atau
   - berbagi, menerbitkan, menyewakan atau menyewakan perangkat lunak, atau menyediakan perangkat lunak sebagai hosting yang berdiri sendiri sebagai solusi untuk digunakan orang lain.

5. PEMBATASAN EKSPOR. Anda harus mematuhi semua undang-undang dan peraturan ekspor domestik dan internasional yang berlaku untuk perangkat lunak, yang mencakup pembatasan tujuan, pengguna akhir, dan penggunaan akhir. Untuk informasi lebih lanjut tentang pembatasan ekspor, kunjungi (alias. ms/exporting).

6. LAYANAN DUKUNGAN. Karena perangkat lunak ini “apa adanya,” kami mungkin tidak menyediakan layanan dukungan untuk itu.

7. KESELURUHAN PERJANJI AN. Perjanjian ini, dan ketentuan untuk suplemen, pembaruan, Internet-based layanan, dan layanan dukungan yang Anda gunakan, adalah keseluruhan perjanjian untuk perangkat lunak dan layanan dukungan.
8. HUKUM YANG BERLAKU. Jika Anda memperoleh perangkat lunak di Amerika Serikat, hukum Washington berlaku untuk interpretasi dan klaim pelanggaran perjanjian ini, dan hukum negara bagian tempat Anda tinggal berlaku untuk semua klaim lainnya. Jika Anda memperoleh perangkat lunak di negara lain mana pun, hukumnya berlaku.
9. HAK KONSUMEN; VARIASI REGIONAL. Perjanjian ini menjelaskan hak-hak hukum tertentu. Anda mungkin memiliki hak lain, termasuk hak konsumen, berdasarkan hukum negara bagian atau negara Anda. Terpisah dan terlepas dari hubungan Anda dengan Microsoft, Anda mungkin juga memiliki hak sehubungan dengan pihak tempat Anda memperoleh perangkat lunak. Perjanjian ini tidak mengubah hak-hak lain tersebut jika hukum negara bagian atau negara Anda tidak mengizinkannya untuk melakukannya. Misalnya, jika Anda memperoleh perangkat lunak di salah satu wilayah di bawah ini, atau hukum negara wajib berlaku, maka ketentuan berikut berlaku untuk Anda:

    - Australia. Anda memiliki jaminan hukum berdasarkan Undang-Undang Konsumen Australia dan tidak ada dalam perjanjian ini yang dimaksudkan untuk mempengaruhi hak-hak tersebut.
    - Kanada. Jika Anda memperoleh perangkat lunak ini di Kanada, Anda dapat berhenti menerima pembaruan dengan mematikan fitur pembaruan otomatis, memutuskan sambungan perangkat Anda dari Internet (jika dan ketika Anda terhubung kembali ke Internet, bagaimanapun, perangkat lunak akan melanjutkan memeriksa dan menginstal pembaruan), atau menghapus instalasi perangkat lunak. Dokumentasi produk, jika ada, juga dapat menentukan cara mematikan pembaruan untuk perangkat atau perangkat lunak spesifik Anda.
    - Jerman dan Austria.
        - Garansi. Perangkat lunak berlisensi dengan benar akan bekerja secara substansif seperti yang dijelaskan dalam materi Microsoft apa pun yang menyertainya perangkat lunak. Namun, Microsoft tidak memberikan jaminan kontrak sehubungan dengan perangkat lunak berlisensi.
        - Batasan Tanggung Jawab. Dalam kasus tindakan yang disengaja, kelalaian berat, klaim berdasarkan Undang-Undang Tanggung Jawab Produk, serta, dalam kasus kematian atau cedera pribadi atau fisik, Microsoft bertanggung jawab sesuai dengan undang-undang hukum.

          Tunduk pada klausul (ii) di atas, Microsoft hanya akan bertanggung jawab atas kelalaian ringan jika Microsoft melanggar kewajiban kontrak material tersebut, yang pemenuhannya memfasilitasi pelaksanaan perjanjian ini, yang pelanggarannya akan membahayakan tujuan perjanjian ini dan kepatuhan yang dapat dipercaya oleh pihak secara konstan (disebut

"kewajiban utama"). Dalam kasus kelalaian ringan lainnya, Microsoft tidak akan bertanggung jawab atas kelalaian ringan.

10. EFEK HUKUM. Perjanjian ini menjelaskan hak-hak hukum tertentu. Anda mungkin memiliki hak lain berdasarkan hukum negara bagian atau negara Anda. Perjanjian ini tidak mengubah hak Anda berdasarkan hukum negara bagian atau negara Anda jika hukum negara bagian atau negara Anda tidak mengizinkannya untuk melakukannya. Tanpa batasan dari hal tersebut di atas, untuk Australia, ANDA MEMILIKI JAMINAN HUKUM BERDASARKAN HUKUM KONSUMEN AUSTRALIA DAN TIDAK ADA DALAM KETENTUAN INI YANG DIMAKSUDKAN UNTUK MEMPENGARUHI HAK-HAK TERSEBUT

11. PENAFIAN GARANSI. PERANGKAT LUNAK INI BERLISENSI "AS-IS." ANDA MENANGGUNG RISIKO MENGGUNAKANNYA. MICROSOFT TIDAK MEMBERIKAN JAMINAN, JAMINAN, ATAU KETENTUAN YANG JELAS. SEJAUH DIIZINKAN BERDASARKAN UNDANG-UNDANG SETEMPAT ANDA, MICROSOFT MENGECUALIKAN JAMINAN TERSIRAT TENTANG KELAYAKAN UNTUK DIPERDAGANGKAN, KESESUAIAN UNTUK TUJUAN TERTENTU, DAN. NON-INFRINGEMENT

12. PEMBATASAN DAN PENGECUALIAN KERUSAKAN. ANDA DAPAT MEMULIHKAN DARI MICROSOFT DAN PEMASOKNYA HANYA KERUSAKAN LANGSUNG HINGGA U.S. $5,00. ANDA TIDAK DAPAT MEMULIHKAN KERUSAKAN LAINNYA, TERMASUK KERUSAKAN KONSEKUENSIAL, KEHILANGAN KEUNTUNGAN, KERUSAKAN KHUSUS, TIDAK LANGSUNG ATAU INSIDENTAL.

    Batasan ini berlaku untuk (a) apa pun yang terkait dengan perangkat lunak, layanan, konten (termasuk kode) di situs Internet pihak ketiga, atau aplikasi pihak ketiga; dan (b) klaim atas pelanggaran kontrak, pelanggaran garansi, jaminan atau kondisi, tanggung jawab yang ketat, kelalaian, atau pelanggaran lainnya sejauh diizinkan oleh hukum yang berlaku.

    Ini juga berlaku bahkan jika Microsoft tahu atau seharusnya tahu tentang kemungkinan kerusakan. Batasan atau pengecualian di atas mungkin tidak berlaku untuk Anda karena negara Anda mungkin tidak mengizinkan pengecualian atau pembatasan kerusakan insidental, konsekuensial, atau lainnya.

11) gambar Docker berbasis windows-microsoft-windows-netfx3-ondemand-package.cab

KETENTUAN LISENSI TAMBAHAN PERANGKAT LUNAK MICROSOFT

MICROSOFT.NET FRAMEWORK 3.5 SP1 UNTUK SISTEM OPERASI MICROSOFT WINDOWS

Microsoft Corporation (atau berdasarkan tempat tinggal Anda, salah satu afiliasinya) melisensikan suplemen ini kepada Anda. Jika Anda dilisensikan untuk menggunakan perangkat lunak sistem operasi Microsoft Windows (untuk mana suplemen ini berlaku) (“perangkat lunak”), Anda dapat menggunakan suplemen ini. Anda tidak boleh menggunakannya jika Anda tidak memiliki lisensi untuk perangkat lunak. Anda dapat menggunakan salinan suplemen ini dengan setiap salinan perangkat lunak berlisensi yang sah.

Ketentuan lisensi berikut menjelaskan ketentuan penggunaan tambahan untuk suplemen ini. Ketentuan ini dan ketentuan lisensi untuk perangkat lunak berlaku untuk penggunaan suplemen oleh Anda. Jika ada konflik, ketentuan lisensi tambahan ini berlaku.

DENGAN MENGGUNAKAN SUPLEMEN INI, ANDA MENERIMA KETENTUAN INI. JIKA ANDA TIDAK MENERIMANYA, JANGAN GUNAKAN SUPLEMEN INI.

Jika Anda mematuhi persyaratan lisensi ini, Anda memiliki hak di bawah ini.

1. LAYANAN DUKUNGAN UNTUK SUPLEMEN. Microsoft menyediakan layanan dukungan untuk perangkat lunak ini seperti yang dijelaskan di www.support.microsoft. com/common/international.aspx.
2. PENGUJIAN TOLOK UKUR MICROSOFT.NET. Perangkat lunak ini mencakup komponen.NET Framework, Windows Communication Foundation, Windows Presentation Foundation, dan Windows Workflow Foundation dari sistem operasi Windows (Komponen.NET). Anda dapat melakukan pengujian benchmark internal Komponen.NET. Anda dapat mengungkapkan hasil pengujian tolok ukur Komponen.NET, asalkan Anda mematuhi ketentuan yang ditetapkan di http://go.microsoft.com/fwlink/? linkID=66406.

   Terlepas dari perjanjian lain yang mungkin Anda miliki dengan Microsoft, jika Anda mengungkapkan hasil pengujian tolok ukur tersebut, Microsoft berhak untuk mengungkapkan hasil pengujian benchmark yang dilakukannya terhadap produk Anda yang bersaing dengan Komponen.NET yang berlaku, asalkan sesuai dengan ketentuan yang sama yang ditetapkan di http://go.microsoft.com/fwlink/? linkID=66406.

12) gambar Docker berbasis windows-dotnet-sdk

(tersedia di https://github.com/dotnet/core/blob/main/LICENSE.TXT)

Lisensi MIT (MIT)

Hak Cipta (c) Microsoft Corporation

Izin dengan ini diberikan, secara gratis, kepada siapa pun yang memperoleh salinan perangkat lunak ini dan file dokumentasi terkait ("Perangkat Lunak"), untuk menangani Perangkat Lunak tanpa batasan, termasuk tanpa batasan hak untuk menggunakan, menyalin, memodifikasi, menggabungkan, menerbitkan, mendistribusikan, mensublisensikan, and/or menjual salinan Perangkat Lunak, dan untuk mengizinkan orang-orang yang kepadanya Perangkat Lunak diberikan untuk melakukannya, tunduk pada ketentuan berikut:

Pemberitahuan hak cipta di atas dan pemberitahuan izin ini harus disertakan dalam semua salinan atau bagian substantif dari Perangkat Lunak.

PERANGKAT LUNAK DISEDIAKAN "SEBAGAIMANA ADANYA", TANPA JAMINAN DALAM BENTUK APA PUN, TERSURAT ATAU TERSIRAT, TERMASUK TETAPI TIDAK TERBATAS PADA JAMINAN KELAYAKAN UNTUK DIPERJUALBELIKAN, KESESUAIAN UNTUK TUJUAN TERTENTU DAN NON-PELANGGARAN. DALAM HAL APA PUN PENULIS ATAU PEMEGANG HAK CIPTA TIDAK AKAN BERTANGGUNG JAWAB ATAS KLAIM, KERUSAKAN ATAU TANGGUNG JAWAB LAINNYA, BAIK DALAM TINDAKAN KONTRAK, GUGATAN ATAU SEBALIKNYA, YANG TIMBUL DARI, DARI ATAU SEHUBUNGAN DENGAN PERANGKAT LUNAK ATAU PENGGUNAAN ATAU TRANSAKSI LAIN DALAM PERANGKAT LUNAK.

Gunakan kunci CodeBuild kondisi sebagai variabel peran layanan IAM untuk mengontrol akses build

Dengan ARN CodeBuild build, Anda dapat membatasi akses sumber daya build dengan menggunakan kunci konteks untuk mengurangi akses sumber daya dalam peran layanan Anda CodeBuild . Untuk CodeBuild, kunci yang dapat digunakan untuk mengontrol perilaku akses build adalah `codebuild:buildArn` dan`codebuild:projectArn`. Dengan ARN proyek build, Anda dapat memverifikasi apakah panggilan ke sumber daya Anda berasal dari proyek build tertentu. Untuk memverifikasi ini, gunakan `codebuild:buildArn` atau kunci `codebuild:projectArn` kondisi dalam kebijakan berbasis identitas IAM.

Untuk menggunakan `codebuild:buildArn` atau `codebuild:projectArn` kondisi kunci dalam polis Anda, sertakan sebagai syarat dengan salah satu operator kondisi ARN. Nilai kunci harus berupa variabel IAM yang menyelesaikan ARN yang valid. Dalam contoh kebijakan di bawah ini, satu-satunya akses yang diizinkan adalah ke proyek build dengan proyek ARN untuk variabel `${codebuild:projectArn}` IAM.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::bucket-name/${codebuild:projectArn}/*"
        }
    ]
}
```

AWS CodeBuild kunci kondisi

AWS CodeBuild menyediakan kumpulan kunci kondisi yang dapat Anda gunakan dalam kebijakan IAM untuk menegakkan kebijakan organisasi Anda pada CodeBuild sumber daya seperti proyek dan armada. Kunci kondisi mencakup sebagian besar konteks permintaan CodeBuild API, termasuk pengaturan jaringan, konfigurasi kredensia, dan batasan komputasi.

Topik

- Menerapkan pengaturan konektivitas VPC pada proyek dan armada Anda
- Mencegah modifikasi tidak sah pada spesifikasi build proyek
- Batasi jenis komputasi untuk build Anda
- Kontrol pengaturan variabel lingkungan
- Gunakan variabel dalam nama kunci kondisi
- Periksa keberadaan atribut dalam permintaan API

Menerapkan pengaturan konektivitas VPC pada proyek dan armada Anda

Kebijakan ini memungkinkan pemanggil menggunakan VPC, subnet, dan grup keamanan yang dipilih saat membuat CodeBuild proyek dan armada. Untuk informasi selengkapnya tentang kunci konteks multinilai, lihat Single-valued vs. kunci konteks multinilai.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [{
        "Effect": "Allow",
        "Action": [
            "codebuild:CreateProject",
            "codebuild:CreateFleet"
        ],
        "Resource": "*",
        "Condition": {
            "ForAnyValue:StringEquals": {
                "codebuild:vpcConfig.vpcId": [
                    "vpc-01234567890abcdef",
                    "vpc-abcdef01234567890"
                ],
                "codebuild:vpcConfig.subnets": [
                    "subnet-1234abcd",
                    "subnet-5678abcd"
                ],
                "codebuild:vpcConfig.securityGroupIds": [
                    "sg-12345678abcdefghij",
                    "sg-01234567abcdefghij"
                ]
            }
        }
    }]
}
```

Mencegah modifikasi tidak sah pada spesifikasi build proyek

Kebijakan ini tidak mengizinkan pemanggil untuk mengganti buildspec di lapangan. `buildspecOverride`

> **Note**
>
> K `codebuild:source.buildspec` unci kondisi hanya mendukung operator Null untuk memeriksa keberadaan bidang API. Itu tidak mengevaluasi konten buildspec.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [{
        "Effect": "Allow",
        "Action": "codebuild:StartBuild",
        "Resource": "*"
    }, {
        "Effect": "Deny",
        "Action": "codebuild:StartBuild",
        "Resource": "*",
        "Condition": {
            "Null": {
                "codebuild:source.buildspec": "false"
            }
        }
    }]
}
```

Batasi jenis komputasi untuk build Anda

Kebijakan ini memungkinkan pembuatan armada yang dapat dibuat hanya dengan `c5.large` atau `m5.large` menghitung jenis instans.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [{
        "Effect": "Allow",
        "Action": "codebuild:CreateFleet",
        "Resource": "*",
```

```
        "Condition": {
            "ForAnyValue:StringEquals": {
                "codebuild:computeConfiguration.instanceType": ["c5.large",
 "m5.large"]
            }
        }
    }]
}
```

Kontrol pengaturan variabel lingkungan

Kebijakan ini memungkinkan pemanggil mengganti variabel `STAGE` lingkungan menjadi salah satu atau`BETA`. `GAMMA` Itu juga secara eksplisit menyangkal menges `STAGE` ampingkan menjadi`PRODUCTION`, dan menyangkal mengesampingkan variabel `MY_APP_VERSION` lingkungan. Untuk beberapa kunci konteks nilai, silakan lihat Single-valued vs. kunci konteks multinilai.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "codebuild:StartBuild"
            ],
            "Resource": "*",
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "codebuild:environment.environmentVariables/STAGE.value": [
                        "BETA",
                        "GAMMA"
                    ]
                }
            }
        },
        {
            "Effect": "Deny",
            "Action": [
                "codebuild:StartBuild"
            ],
```

```
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "codebuild:environment.environmentVariables/STAGE.value":
 "PRODUCTION"
                },
                "ForAnyValue:StringEquals": {
                    "codebuild:environment.environmentVariables.name": [
                        "MY_APP_VERSION"
                    ]
                }
            }
        }
    ]
}
```

Gunakan variabel dalam nama kunci kondisi

Anda dapat menggunakan variabel dalam nama kunci kondisi seperti `secondarySources/${sourceIdentifier}.location` dan`secondaryArtifacts/${artifactIdentifier}.location`, di mana Anda dapat menentukan sumber sekunder atau pengidentifikasi artefak sekunder dalam kebijakan IAM. Kebijakan di bawah ini memungkinkan pemanggil membuat proyek dengan lokasi sumber tertentu untuk sumber `mySecondSource` sekunder.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "codebuild:CreateProject",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "codebuild:secondarySources/mySecondSource.location": "my-
source-location"
                }
            }
```

```
        }
    ]
}
```

Periksa keberadaan atribut dalam permintaan API

CodeBuild mendukung kunci kondisi untuk memeriksa keberadaan beberapa bidang dalam permintaan API. Kebijakan memberlakukan persyaratan VPC saat membuat atau memperbarui proyek.

```
{
    "Version": "2012-10-17"             ,
    "Statement": [{
        "Effect": "Allow",
        "Action": [
            "codebuild:CreateProject",
            "codebuild:UpdateProject"
        ],
        "Resource": "*",
        "Condition": {
            "Null": {
                "codebuild:vpcConfig.vpcId": "false"
            }
        }
    }]
}
```

Contoh kode untuk CodeBuild menggunakan AWS SDK

Contoh kode berikut menunjukkan cara menggunakan CodeBuild kit pengembangan AWS perangkat lunak (SDK).

Tindakan merupakan kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Sementara tindakan menunjukkan cara memanggil fungsi layanan individual, Anda dapat melihat tindakan dalam konteks dalam skenario terkait.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan layanan ini dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Contoh kode

- Contoh dasar untuk CodeBuild menggunakan AWS SDK
  - Tindakan untuk CodeBuild menggunakan AWS SDK
    - Gunakan CreateProject dengan AWS SDK atau CLI
    - Gunakan ListBuilds dengan AWS SDK atau CLI
    - Gunakan ListProjects dengan AWS SDK atau CLI
    - Gunakan StartBuild dengan AWS SDK atau CLI

Contoh dasar untuk CodeBuild menggunakan AWS SDK

Contoh kode berikut menunjukkan cara menggunakan dasar-dasar AWS CodeBuild dengan AWS SDK.

Contoh

- Tindakan untuk CodeBuild menggunakan AWS SDK
  - Gunakan CreateProject dengan AWS SDK atau CLI
  - Gunakan ListBuilds dengan AWS SDK atau CLI
  - Gunakan ListProjects dengan AWS SDK atau CLI
  - Gunakan StartBuild dengan AWS SDK atau CLI

Tindakan untuk CodeBuild menggunakan AWS SDK

Contoh kode berikut menunjukkan cara melakukan CodeBuild tindakan individual dengan AWS SDK. Setiap contoh menyertakan tautan ke GitHub, di mana Anda dapat menemukan instruksi untuk mengatur dan menjalankan kode.

Contoh berikut hanya mencakup tindakan yang paling umum digunakan. Untuk daftar lengkapnya, lihat Referensi AWS CodeBuild API.

Contoh

- Gunakan CreateProject dengan AWS SDK atau CLI
- Gunakan ListBuilds dengan AWS SDK atau CLI
- Gunakan ListProjects dengan AWS SDK atau CLI
- Gunakan StartBuild dengan AWS SDK atau CLI

Gunakan **`CreateProject`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`CreateProject`.

CLI

AWS CLI

Contoh 1: Untuk membuat proyek AWS CodeBuild build

`create-project`Contoh berikut membuat proyek CodeBuild build menggunakan file sumber dari bucket S3

```
aws codebuild create-project \
    --name "my-demo-project" \
    --source "{\"type\": \"S3\",\"location\": \"codebuild-us-west-2-123456789012-
input-bucket/my-source.zip\"}" \
    --artifacts {"\"type\": \"S3\",\"location\": \"codebuild-us-
west-2-123456789012-output-bucket\""} \
    --environment "{\"type\": \"LINUX_CONTAINER\",\"image\": \"aws/codebuild/
standard:1.0\",\"computeType\": \"BUILD_GENERAL1_SMALL\"}" \
    --service-role "arn:aws:iam::123456789012:role/service-role/my-codebuild-
service-role"
```

Output:

```
{
    "project": {
        "arn": "arn:aws:codebuild:us-west-2:123456789012:project/my-demo-
project",
        "name": "my-cli-demo-project",
        "encryptionKey": "arn:aws:kms:us-west-2:123456789012:alias/aws/s3",
        "serviceRole": "arn:aws:iam::123456789012:role/service-role/my-codebuild-
service-role",
        "lastModified": 1556839783.274,
        "badge": {
            "badgeEnabled": false
        },
        "queuedTimeoutInMinutes": 480,
        "environment": {
            "image": "aws/codebuild/standard:1.0",
            "computeType": "BUILD_GENERAL1_SMALL",
            "type": "LINUX_CONTAINER",
            "imagePullCredentialsType": "CODEBUILD",
            "privilegedMode": false,
            "environmentVariables": []
        },
        "artifacts": {
            "location": "codebuild-us-west-2-123456789012-output-bucket",
            "name": "my-cli-demo-project",
            "namespaceType": "NONE",
            "type": "S3",
            "packaging": "NONE",
            "encryptionDisabled": false
        },
        "source": {
            "type": "S3",
            "location": "codebuild-us-west-2-123456789012-input-bucket/my-
source.zip",
            "insecureSsl": false
        },
        "timeoutInMinutes": 60,
        "cache": {
            "type": "NO_CACHE"
        },
        "created": 1556839783.274
    }
```

```
}
```

Contoh 2: Untuk membuat proyek AWS CodeBuild build menggunakan file input JSON untuk parameter

`create-project`Contoh berikut membuat proyek CodeBuild build dengan meneruskan semua parameter yang diperlukan dalam file input JSON. Buat template file input dengan menjalankan perintah hanya dengan file`--generate-cli-skeleton parameter`.

```
aws codebuild create-project --cli-input-json file://create-project.json
```

File JSON masukan `create-project.json` berisi konten berikut:

```
{
    "name": "codebuild-demo-project",
    "source": {
        "type": "S3",
        "location": "codebuild-region-ID-account-ID-input-bucket/MessageUtil.zip"
    },
    "artifacts": {
        "type": "S3",
        "location": "codebuild-region-ID-account-ID-output-bucket"
    },
    "environment": {
        "type": "LINUX_CONTAINER",
        "image": "aws/codebuild/standard:1.0",
        "computeType": "BUILD_GENERAL1_SMALL"
    },
    "serviceRole": "serviceIAMRole"
}
```

Output:

```
{
    "project": {
        "name": "codebuild-demo-project",
        "serviceRole": "serviceIAMRole",
        "tags": [],
        "artifacts": {
            "packaging": "NONE",
            "type": "S3",
            "location": "codebuild-region-ID-account-ID-output-bucket",
```

```
            "name": "message-util.zip"
        },
        "lastModified": 1472661575.244,
        "timeoutInMinutes": 60,
        "created": 1472661575.244,
        "environment": {
            "computeType": "BUILD_GENERAL1_SMALL",
            "image": "aws/codebuild/standard:1.0",
            "type": "LINUX_CONTAINER",
            "environmentVariables": []
        },
        "source": {
            "type": "S3",
            "location": "codebuild-region-ID-account-ID-input-bucket/
MessageUtil.zip"
        },
        "encryptionKey": "arn:aws:kms:region-ID:account-ID:alias/aws/s3",
        "arn": "arn:aws:codebuild:region-ID:account-ID:project/codebuild-demo-
project"
    }
}
```

Untuk informasi selengkapnya, lihat Membuat Proyek Build (AWS CLI) di AWS CodeBuild Panduan Pengguna.

- Untuk detail API, lihat CreateProjectdi Referensi AWS CLI Perintah.

JavaScript

SDK untuk JavaScript (v3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

Buat proyek.

```
import {
  ArtifactsType,
  CodeBuildClient,
```

```
  ComputeType,
  CreateProjectCommand,
  EnvironmentType,
  SourceType,
} from "@aws-sdk/client-codebuild";

// Create the AWS CodeBuild project.
export const createProject = async (
  projectName = "MyCodeBuilder",
  roleArn = "arn:aws:iam::xxxxxxxxxxxx:role/CodeBuildAdmin",
  buildOutputBucket = "xxxx",
  githubUrl = "https://...",
) => {
  const codeBuildClient = new CodeBuildClient({});

  const response = await codeBuildClient.send(
    new CreateProjectCommand({
      artifacts: {
        // The destination of the build artifacts.
        type: ArtifactsType.S3,
        location: buildOutputBucket,
      },
      // Information about the build environment. The combination of
 "computeType" and "type" determines the
      // requirements for the environment such as CPU, memory, and disk space.
      environment: {
        // Build environment compute types.
        // https://docs.aws.amazon.com/codebuild/latest/userguide/build-env-ref-
compute-types.html
        computeType: ComputeType.BUILD_GENERAL1_SMALL,
        // Docker image identifier.
        // See https://docs.aws.amazon.com/codebuild/latest/userguide/build-env-
ref-available.html
        image: "aws/codebuild/standard:7.0",
        // Build environment type.
        type: EnvironmentType.LINUX_CONTAINER,
      },
      name: projectName,
      // A role ARN with permission to create a CodeBuild project, write to the
 artifact location, and write CloudWatch logs.
      serviceRole: roleArn,
      source: {
        // The type of repository that contains the source code to be built.
        type: SourceType.GITHUB,
```

```
       // The location of the repository that contains the source code to be
built.
       location: githubUrl,
     },
   }),
 );
 console.log(response);
 //   {
 //     '$metadata': {
 //       httpStatusCode: 200,
 //       requestId: 'b428b244-777b-49a6-a48d-5dffedced8e7',
 //       extendedRequestId: undefined,
 //       cfId: undefined,
 //       attempts: 1,
 //       totalRetryDelay: 0
 //     },
 //     project: {
 //       arn: 'arn:aws:codebuild:us-east-1:xxxxxxxxxxxx:project/MyCodeBuilder',
 //       artifacts: {
 //         encryptionDisabled: false,
 //         location: 'xxxxxx-xxxxxxx-xxxxxx',
 //         name: 'MyCodeBuilder',
 //         namespaceType: 'NONE',
 //         packaging: 'NONE',
 //         type: 'S3'
 //       },
 //       badge: { badgeEnabled: false },
 //       cache: { type: 'NO_CACHE' },
 //       created: 2023-08-18T14:46:48.979Z,
 //       encryptionKey: 'arn:aws:kms:us-east-1:xxxxxxxxxxxx:alias/aws/s3',
 //       environment: {
 //         computeType: 'BUILD_GENERAL1_SMALL',
 //         environmentVariables: [],
 //         image: 'aws/codebuild/standard:7.0',
 //         imagePullCredentialsType: 'CODEBUILD',
 //         privilegedMode: false,
 //         type: 'LINUX_CONTAINER'
 //       },
 //       lastModified: 2023-08-18T14:46:48.979Z,
 //       name: 'MyCodeBuilder',
 //       projectVisibility: 'PRIVATE',
 //       queuedTimeoutInMinutes: 480,
 //       serviceRole: 'arn:aws:iam::xxxxxxxxxxxx:role/CodeBuildAdmin',
 //       source: {
```

```
  //         insecureSsl: false,
  //         location: 'https://...',
  //         reportBuildStatus: false,
  //         type: 'GITHUB'
  //       },
  //       timeoutInMinutes: 60
  //     }
  //   }
  return response;
};
```

- Untuk informasi selengkapnya, silakan lihat Panduan Developer AWS SDK untuk JavaScript.
- Untuk detail API, lihat CreateProjectdi Referensi AWS SDK untuk JavaScript API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan layanan ini dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **`ListBuilds`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`ListBuilds`.

C++

SDK untuk C++

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
//! List the CodeBuild builds.
/*!
  \param sortType: 'SortOrderType' type.
  \param clientConfiguration: AWS client configuration.
  \return bool: Function succeeded.
 */
bool AwsDoc::CodeBuild::listBuilds(Aws::CodeBuild::Model::SortOrderType sortType,
```

```
                                    const Aws::Client::ClientConfiguration
&clientConfiguration) {
   Aws::CodeBuild::CodeBuildClient codeBuildClient(clientConfiguration);

   Aws::CodeBuild::Model::ListBuildsRequest listBuildsRequest;
   listBuildsRequest.SetSortOrder(sortType);

   Aws::String nextToken; // Used for pagination.

   do {
       if (!nextToken.empty()) {
           listBuildsRequest.SetNextToken(nextToken);
       }

       Aws::CodeBuild::Model::ListBuildsOutcome listBuildsOutcome =
codeBuildClient.ListBuilds(
               listBuildsRequest);

       if (listBuildsOutcome.IsSuccess()) {
           const Aws::Vector<Aws::String> &ids =
listBuildsOutcome.GetResult().GetIds();
           if (!ids.empty()) {

               std::cout << "Information about each build:" << std::endl;
               Aws::CodeBuild::Model::BatchGetBuildsRequest getBuildsRequest;
               getBuildsRequest.SetIds(listBuildsOutcome.GetResult().GetIds());
               Aws::CodeBuild::Model::BatchGetBuildsOutcome getBuildsOutcome =
codeBuildClient.BatchGetBuilds(
                       getBuildsRequest);

               if (getBuildsOutcome.IsSuccess()) {
                   const Aws::Vector<Aws::CodeBuild::Model::Build> &builds =
getBuildsOutcome.GetResult().GetBuilds();
                   std::cout << builds.size() << " build(s) found." <<
std::endl;
                   for (auto val: builds) {
                       std::cout << val.GetId() << std::endl;
                   }
               } else {
                   std::cerr << "Error getting builds"
                             << getBuildsOutcome.GetError().GetMessage() <<
std::endl;
                   return false;
               }
```

```
            } else {
                std::cout << "No builds found." << std::endl;
            }

            // Get the next token for pagination.

            nextToken = listBuildsOutcome.GetResult().GetNextToken();
        } else {
            std::cerr << "Error listing builds"
                      << listBuildsOutcome.GetError().GetMessage()
                      << std::endl;
            return false;
        }

    } while (!nextToken.

            empty()

            );

    return true;
}
```

- Untuk detail API, lihat ListBuildsdi Referensi AWS SDK untuk C++ API.

CLI

AWS CLI

Untuk mendapatkan daftar ID AWS CodeBuild build.

`list-builds`Contoh berikut mendapatkan daftar CodeBuild ID diurutkan dalam urutan menaik.

```
aws codebuild list-builds --sort-order ASCENDING
```

Output mencakup `nextToken` nilai yang menunjukkan bahwa ada lebih banyak output yang tersedia.

```
{
```

```
    "nextToken": "4AEA6u7J...The full token has been omitted for
 brevity...MzY2OA==",
    "ids": [
        "codebuild-demo-project:815e755f-bade-4a7e-80f0-efe51EXAMPLE"
        "codebuild-demo-project:84a7f3d1-d40e-4956-b4cf-7a9d4EXAMPLE"
            ... The full list of build IDs has been omitted for brevity ...
        "codebuild-demo-project:931d0b72-bf6f-4040-a472-5c707EXAMPLE"
    ]
}
```

Jalankan perintah ini lagi dan berikan `nextToken` nilai dalam respons sebelumnya sebagai parameter untuk mendapatkan bagian berikutnya dari output. Ulangi sampai Anda tidak menerima `nextToken` nilai dalam respons.

```
aws codebuild list-builds --sort-order ASCENDING --next-
token 4AEA6u7J...The full token has been omitted for brevity...MzY2OA==
```

Bagian selanjutnya dari output:

```
{
    "ids": [
        "codebuild-demo-project:49015049-21cf-4b50-9708-df115EXAMPLE",
        "codebuild-demo-project:543e7206-68a3-46d6-a4da-759abEXAMPLE",
            ... The full list of build IDs has been omitted for brevity ...
        "codebuild-demo-project:c282f198-4582-4b38-bdc0-26f96EXAMPLE"
    ]
}
```

Untuk informasi selengkapnya, lihat Melihat Daftar ID Build (AWS CLI) di Panduan Pengguna AWS CodeBuild

- Untuk detail API, lihat ListBuildsdi Referensi AWS CLI Perintah.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan layanan ini dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **`ListProjects`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`ListProjects`.

C++

SDK untuk C++

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
//! List the CodeBuild projects.
/*!
  \param sortType: 'SortOrderType' type.
  \param clientConfiguration: AWS client configuration.
  \return bool: Function succeeded.
 */
bool AwsDoc::CodeBuild::listProjects(Aws::CodeBuild::Model::SortOrderType
 sortType,
                                     const Aws::Client::ClientConfiguration
 &clientConfiguration) {
    Aws::CodeBuild::CodeBuildClient codeBuildClient(clientConfiguration);

    Aws::CodeBuild::Model::ListProjectsRequest listProjectsRequest;
    listProjectsRequest.SetSortOrder(sortType);

    Aws::String nextToken; // Next token for pagination.
    Aws::Vector<Aws::String> allProjects;

    do {
        if (!nextToken.empty()) {
            listProjectsRequest.SetNextToken(nextToken);
        }

        Aws::CodeBuild::Model::ListProjectsOutcome outcome =
 codeBuildClient.ListProjects(
                listProjectsRequest);

        if (outcome.IsSuccess()) {
            const Aws::Vector<Aws::String> &projects =
 outcome.GetResult().GetProjects();
            allProjects.insert(allProjects.end(), projects.begin(),
 projects.end());
```

```
            nextToken = outcome.GetResult().GetNextToken();
        }

        else {
            std::cerr << "Error listing projects" <<
 outcome.GetError().GetMessage()
                      << std::endl;
        }

    } while (!nextToken.empty());

    std::cout << allProjects.size() << " project(s) found." << std::endl;
    for (auto project: allProjects) {
        std::cout << project << std::endl;
    }

    return true;
}
```

- Untuk detail API, lihat ListProjectsdi Referensi AWS SDK untuk C++ API.

CLI

AWS CLI

Untuk mendapatkan daftar nama proyek AWS CodeBuild build.

`list-projects`Contoh berikut mendapatkan daftar proyek CodeBuild build yang diurutkan berdasarkan nama dalam urutan menaik.

```
aws codebuild list-projects --sort-by NAME --sort-order ASCENDING
```

Output mencakup `nextToken` nilai yang menunjukkan bahwa ada lebih banyak output yang tersedia.

```
{
    "nextToken": "Ci33ACF6...The full token has been omitted for brevity...U
+AkMx8=",
    "projects": [
        "codebuild-demo-project",
```

```
        "codebuild-demo-project2",
            ... The full list of build project names has been omitted for
 brevity ...
        "codebuild-demo-project99"
    ]
}
```

Jalankan perintah ini lagi dan berikan `nextToken` nilai dari respons sebelumnya sebagai parameter untuk mendapatkan bagian berikutnya dari output. Ulangi sampai Anda tidak menerima `nextToken` nilai dalam respons.

```
aws codebuild list-projects  --sort-by NAME --sort-order ASCENDING --next-
token Ci33ACF6...The full token has been omitted for brevity...U+AkMx8=

{
    "projects": [
        "codebuild-demo-project100",
        "codebuild-demo-project101",

  ... The full list of build project names has been omitted for brevity ...
        "codebuild-demo-project122"

  ]
}
```

Untuk informasi selengkapnya, lihat Melihat Daftar Nama Proyek Build (AWS CLI) di AWS CodeBuild Panduan Pengguna.

- Untuk detail API, lihat ListProjectsdi Referensi AWS CLI Perintah.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan layanan ini dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **`StartBuild`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`StartBuild`.

C++

SDK untuk C++

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
//! Start an AWS CodeBuild project build.
/*!
  \param projectName: A CodeBuild project name.
  \param clientConfiguration: AWS client configuration.
  \return bool: Function succeeded.
 */
bool AwsDoc::CodeBuild::startBuild(const Aws::String &projectName,
                                   const Aws::Client::ClientConfiguration
 &clientConfiguration) {
    Aws::CodeBuild::CodeBuildClient codeBuildClient(clientConfiguration);

    Aws::CodeBuild::Model::StartBuildRequest startBuildRequest;
    startBuildRequest.SetProjectName(projectName);

    Aws::CodeBuild::Model::StartBuildOutcome outcome =
 codeBuildClient.StartBuild(
            startBuildRequest);

    if (outcome.IsSuccess()) {
        std::cout << "Successfully started build" << std::endl;
        std::cout << "Build ID: " << outcome.GetResult().GetBuild().GetId()
                  << std::endl;
    }

    else {
        std::cerr << "Error starting build" << outcome.GetError().GetMessage()
                  << std::endl;
    }

    return outcome.IsSuccess();
}
```

- Untuk detail API, lihat StartBuilddi Referensi AWS SDK untuk C++ API.

CLI

AWS CLI

Untuk mulai menjalankan build proyek AWS CodeBuild build.

`start-build`Contoh berikut memulai build untuk CodeBuild proyek yang ditentukan. Build mengesampingkan pengaturan proyek selama beberapa menit build diizinkan untuk antri sebelum waktu habis dan pengaturan artefak proyek.

```
aws codebuild start-build \
    --project-name "my-demo-project" \
    --queued-timeout-in-minutes-override 5 \
    --artifacts-override {"\"type\": \"S3\",\"location\":
 \"arn:aws:s3:::artifacts-override\",\"overrideArtifactName\":true"}
```

Output:

```
{
    "build": {
        "serviceRole": "arn:aws:iam::123456789012:role/service-role/my-codebuild-
service-role",
        "buildStatus": "IN_PROGRESS",
        "buildComplete": false,
        "projectName": "my-demo-project",
        "timeoutInMinutes": 60,
        "source": {
            "insecureSsl": false,
            "type": "S3",
            "location": "codebuild-us-west-2-123456789012-input-bucket/my-
source.zip"
        },
        "queuedTimeoutInMinutes": 5,
        "encryptionKey": "arn:aws:kms:us-west-2:123456789012:alias/aws/s3",
        "currentPhase": "QUEUED",
        "startTime": 1556905683.568,
        "environment": {
            "computeType": "BUILD_GENERAL1_MEDIUM",
            "environmentVariables": [],
            "type": "LINUX_CONTAINER",
```

```
            "privilegedMode": false,
            "image": "aws/codebuild/standard:1.0",
            "imagePullCredentialsType": "CODEBUILD"
        },
        "phases": [
            {
                "phaseStatus": "SUCCEEDED",
                "startTime": 1556905683.568,
                "phaseType": "SUBMITTED",
                "durationInSeconds": 0,
                "endTime": 1556905684.524
            },
            {
                "startTime": 1556905684.524,
                "phaseType": "QUEUED"
            }
        ],
        "logs": {
            "deepLink": "https://console.aws.amazon.com/cloudwatch/home?
region=us-west-2#logEvent:group=null;stream=null"
        },
        "artifacts": {
            "encryptionDisabled": false,
            "location": "arn:aws:s3:::artifacts-override/my-demo-project",
            "overrideArtifactName": true
        },
        "cache": {
            "type": "NO_CACHE"
        },
        "id": "my-demo-project::12345678-a1b2-c3d4-e5f6-11111EXAMPLE",
        "initiator": "my-aws-account-name",
        "arn": "arn:aws:codebuild:us-west-2:123456789012:build/my-demo-
project::12345678-a1b2-c3d4-e5f6-11111EXAMPLE"
    }
}
```

Untuk informasi selengkapnya, lihat Menjalankan Build (AWS CLI) di AWS CodeBuild Panduan Pengguna.

- Untuk detail API, lihat StartBuilddi Referensi AWS CLI Perintah.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan layanan ini dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

API build publik

API build publik digunakan oleh AWS CodeBuild konsol dan situs web hasil build publik untuk mendapatkan informasi tentang proyek pembangunan publik. Untuk informasi selengkapnya, lihat the section called “Dapatkan URL proyek build publik”.

> **Note**
>
> API build CodeBuild publik tidak terdapat dalam AWS CLI atau AWS SDK.

Topik

- Tindakan membangun publik
- Tipe data build publik

Tindakan membangun publik

> **Note**
>
> API build CodeBuild publik tidak terdapat dalam AWS CLI atau AWS SDK.

Tindakan berikut didukung oleh API build CodeBuild publik:

Tindakan

- DescribeBuildBatchesForPublicProject
- DescribeBuildsForPublicProject
- GetCloudWatchLogsForPublicBuild
- GetPresignedUrlsForPublicBuild
- GetPublicBuild
- GetPublicBuildBatch
- GetPublicProject

DescribeBuildBatchesForPublicProject

> **Note**
>
> Elemen API ini tidak terkandung dalam AWS CLI atau AWS SDK.

Sintaksis Permintaan

```
{
   "filter": {
      "status": "string"
   },
   "maxResults": number,
   "nextToken": "string",
   "publicProjectAlias": "string",
   "sortOrder": "string"
}
```

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

Permintaan menerima data berikut dalam format JSON.

> **Note**
>
> Dalam daftar berikut, parameter yang diperlukan dijelaskan terlebih dahulu.

publik ProjectAlias

Tipe: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 1024.

Pola: `^[0-9a-zA-Z%+=]+$`

Wajib: Ya

menyaring

Menentukan filter saat mengambil build batch.

Tipe: Objek BuildBatchFilter

Wajib: Tidak

maxResults

Tipe: Bilangan Bulat

Rentang yang Valid: Nilai minimum 1. Nilai maksimum 100.

Wajib: Tidak

NextToken

Tipe: String

Wajib: Tidak

SortOrder

Tipe: String

Nilai yang Valid: `ASCENDING | DESCENDING`

Wajib: Tidak

Sintaksis Respons

```
{
   "nextToken": "string",
   "publicBuildBatches": [
      {
         "buildBatchNumber": number,
         "buildBatchStatus": "string",
         "endTime": number,
         "publicBuildBatchAlias": "string",
         "sourceVersion": "string",
         "startTime": number
      }
   ]
```

```
}
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

NextToken

Tipe: String

publik BuildBatches

Tipe: Array objek BuildBatchForDescribeBuildBatchesPublic

Kesalahan

Untuk informasi tentang kesalahan yang umum terjadi pada semua tindakan, lihat Kesalahan Umum.

InvalidInputException

Kode Status HTTP: 400

ResourceNotFoundException

Kode Status HTTP: 400

DescribeBuildsForPublicProject

> **Note**
>
> Elemen API ini tidak terkandung dalam AWS CLI atau AWS SDK.

Sintaksis Permintaan

```
{
   "maxResults": number,
   "nextToken": "string",
   "publicProjectAlias": "string",
   "sortOrder": "string"
}
```

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

Permintaan menerima data berikut dalam format JSON.

> **Note**
>
> Dalam daftar berikut, parameter yang diperlukan dijelaskan terlebih dahulu.

publik ProjectAlias

Tipe: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 1024.

Pola: `^[0-9a-zA-Z%+=]+$`

Wajib: Ya

maxResults

Tipe: Bilangan Bulat

Rentang yang Valid: Nilai minimum 1. Nilai maksimum 100.

Wajib: Tidak

NextToken

Tipe: String

Wajib: Tidak

SortOrder

Tipe: String

Nilai yang Valid: `ASCENDING | DESCENDING`

Wajib: Tidak

Sintaksis Respons

```
{
   "builds": [
      {
         "artifacts": {
            "artifactIdentifier": "string",
            "packaging": "string",
            "type": "string"
         },
         "buildComplete": boolean,
         "buildNumber": number,
         "buildStatus": "string",
         "endTime": number,
         "environment": {
            "computeType": "string",
            "environmentVariables": [
               {
                  "name": "string",
                  "type": "string",
                  "value": "string"
               }
            ],
            "image": "string",
            "type": "string"
         },
         "id": "string",
         "initiator": "string",
```

```
        "logsStatus": {
           "cloudWatchLogsStatus": "string",
           "s3LogsStatus": "string"
        },
        "phases": [
           {
              "contexts": [
                 {
                    "message": "string",
                    "statusCode": "string"
                 }
              ],
              "durationInSeconds": number,
              "endTime": number,
              "phaseStatus": "string",
              "phaseType": "string",
              "startTime": number
           }
        ],
        "projectName": "string",
        "queuedTimeoutInMinutes": number,
        "resolvedSourceVersion": "string",
        "secondaryArtifacts": [
           {
              "artifactIdentifier": "string",
              "packaging": "string",
              "type": "string"
           }
        ],
        "secondarySources": [
           {
              "buildspec": "string",
              "gitCloneDepth": number,
              "gitSubmodulesConfig": {
                 "fetchSubmodules": boolean
              },
              "location": "string",
              "sourceIdentifier": "string",
              "type": "string"
           }
        ],
        "secondarySourceVersions": [
           {
              "sourceIdentifier": "string",
```

```
            "sourceVersion": "string"
         }
      ],
      "source": {
         "buildspec": "string",
         "gitCloneDepth": number,
         "gitSubmodulesConfig": {
            "fetchSubmodules": boolean
         },
         "location": "string",
         "sourceIdentifier": "string",
         "type": "string"
      },
      "sourceVersion": "string",
      "startTime": number,
      "timeoutInMinutes": number
   }
],
"nextToken": "string"
}
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

membangun

Tipe: Array objek PublicBuild

NextToken

Tipe: String

Kesalahan

Untuk informasi tentang kesalahan yang umum terjadi pada semua tindakan, lihat Kesalahan Umum.

InvalidInputException

Kode Status HTTP: 400

ResourceNotFoundException

Kode Status HTTP: 400

GetCloudWatchLogsForPublicBuild

> **Note**
>
> Elemen API ini tidak terkandung dalam AWS CLI atau AWS SDK.

Sintaksis Permintaan

```
{
   "maxResult": number,
   "nextToken": "string",
   "publicBuildAlias": "string"
}
```

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

Permintaan menerima data berikut dalam format JSON.

> **Note**
>
> Dalam daftar berikut, parameter yang diperlukan dijelaskan terlebih dahulu.

publik BuildAlias

Tipe: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 1024.

Pola: `^[0-9a-zA-Z%+=]+:\p{XDigit}{8}(-\p{XDigit}{4}){3}-\p{XDigit}{12}$`

Wajib: Ya

MaxResult

Tipe: Bilangan Bulat

Rentang yang Valid: Nilai minimum 1. Nilai maksimum 10000.

Wajib: Tidak

NextToken

Tipe: String

Wajib: Tidak

Sintaksis Respons

```
{
   "logs": [ "string" ],
   "nextToken": "string"
}
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

log

Tipe: Array string

NextToken

Tipe: String

Batasan Panjang: Panjang minimum 1.

Kesalahan

Untuk informasi tentang kesalahan yang umum terjadi pada semua tindakan, lihat Kesalahan Umum.

InvalidInputException

Kode Status HTTP: 400

ResourceNotFoundException

Kode Status HTTP: 400

GetPresignedUrlsForPublicBuild

> **Note**
>
> Elemen API ini tidak terkandung dalam AWS CLI atau AWS SDK.

Sintaksis Permintaan

```
{
   "publicBuildAlias": "string"
}
```

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

Permintaan menerima data berikut dalam format JSON.

> **Note**
>
> Dalam daftar berikut, parameter yang diperlukan dijelaskan terlebih dahulu.

publik BuildAlias

Tipe: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 1024.

Pola: `^[0-9a-zA-Z%+=]+:\p{XDigit}{8}(-\p{XDigit}{4}){3}-\p{XDigit}{12}$`

Diperlukan: Ya

Sintaksis Respons

```
{
   "artifacts": {
```

```
      "expiredAt": number,
      "identifier": "string",
      "md5Checksum": "string",
      "presignedUrl": "string",
      "s3Arn": "string",
      "sha256Checksum": "string"
   },
   "log": {
      "expiredAt": number,
      "identifier": "string",
      "md5Checksum": "string",
      "presignedUrl": "string",
      "s3Arn": "string",
      "sha256Checksum": "string"
   },
   "secondaryArtifacts": [
      {
         "expiredAt": number,
         "identifier": "string",
         "md5Checksum": "string",
         "presignedUrl": "string",
         "s3Arn": "string",
         "sha256Checksum": "string"
      }
   ]
}
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

artefak

Tipe: Objek S3Dapat Diunduh

log

Tipe: Objek S3Dapat Diunduh

SecondaryArtifacts

Tipe: Array objek S3Dapat Diunduh

Kesalahan

Untuk informasi tentang kesalahan yang umum terjadi pada semua tindakan, lihat Kesalahan Umum.

InvalidInputException

Kode Status HTTP: 400

ResourceNotFoundException

Kode Status HTTP: 400

GetPublicBuild

> **Note**
>
> Elemen API ini tidak terkandung dalam AWS CLI atau AWS SDK.

Sintaksis Permintaan

```
{
   "publicBuildAlias": "string"
}
```

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

Permintaan menerima data berikut dalam format JSON.

> **Note**
>
> Dalam daftar berikut, parameter yang diperlukan dijelaskan terlebih dahulu.

publik BuildAlias

Tipe: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 1024.

Pola: `^[0-9a-zA-Z%+=]+:\p{XDigit}{8}(-\p{XDigit}{4}){3}-\p{XDigit}{12}$`

Diperlukan: Ya

Sintaksis Respons

```
{
   "build": {
      "artifacts": {
         "artifactIdentifier": "string",
         "packaging": "string",
```

```
      "type": "string"
   },
   "buildComplete": boolean,
   "buildNumber": number,
   "buildStatus": "string",
   "endTime": number,
   "environment": {
      "computeType": "string",
      "environmentVariables": [
         {
            "name": "string",
            "type": "string",
            "value": "string"
         }
      ],
      "image": "string",
      "type": "string"
   },
   "id": "string",
   "initiator": "string",
   "logsStatus": {
      "cloudWatchLogsStatus": "string",
      "s3LogsStatus": "string"
   },
   "phases": [
      {
         "contexts": [
            {
               "message": "string",
               "statusCode": "string"
            }
         ],
         "durationInSeconds": number,
         "endTime": number,
         "phaseStatus": "string",
         "phaseType": "string",
         "startTime": number
      }
   ],
   "projectName": "string",
   "queuedTimeoutInMinutes": number,
   "resolvedSourceVersion": "string",
   "secondaryArtifacts": [
      {
```

```
            "artifactIdentifier": "string",
            "packaging": "string",
            "type": "string"
         }
      ],
      "secondarySources": [
         {
            "buildspec": "string",
            "gitCloneDepth": number,
            "gitSubmodulesConfig": {
               "fetchSubmodules": boolean
            },
            "location": "string",
            "sourceIdentifier": "string",
            "type": "string"
         }
      ],
      "secondarySourceVersions": [
         {
            "sourceIdentifier": "string",
            "sourceVersion": "string"
         }
      ],
      "source": {
         "buildspec": "string",
         "gitCloneDepth": number,
         "gitSubmodulesConfig": {
            "fetchSubmodules": boolean
         },
         "location": "string",
         "sourceIdentifier": "string",
         "type": "string"
      },
      "sourceVersion": "string",
      "startTime": number,
      "timeoutInMinutes": number
   }
}
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

membangun

Tipe: Objek PublicBuild

Kesalahan

Untuk informasi tentang kesalahan yang umum terjadi pada semua tindakan, lihat Kesalahan Umum.

InvalidInputException

Kode Status HTTP: 400

ResourceNotFoundException

Kode Status HTTP: 400

GetPublicBuildBatch

> **Note**
>
> Elemen API ini tidak terkandung dalam AWS CLI atau AWS SDK.

Sintaksis Permintaan

```
{
   "publicBuildBatchAlias": "string"
}
```

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

Permintaan menerima data berikut dalam format JSON.

> **Note**
>
> Dalam daftar berikut, parameter yang diperlukan dijelaskan terlebih dahulu.

publik BuildBatchAlias

Tipe: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 1024.

Pola: `^[0-9a-zA-Z%+=]+:[a-z0-9]{8}(-[a-z0-9]{4}){3}-[a-z0-9]{12}$`

Diperlukan: Ya

Sintaksis Respons

```
{
   "publicBuildBatch": {
      "artifacts": {
         "artifactIdentifier": "string",
         "packaging": "string",
```

```
      "type": "string"
   },
   "buildBatchConfig": {
      "combineArtifacts": boolean,
      "restrictions": {
         "computeTypesAllowed": [ "string" ],
         "maximumBuildsAllowed": number
      },
      "timeoutInMins": number
   },
   "buildBatchNumber": number,
   "buildBatchStatus": "string",
   "buildGroups": [
      {
         "currentBuildSummary": {
            "buildStatus": "string",
            "primaryArtifact": {
               "identifier": "string",
               "location": "string",
               "type": "string"
            },
            "publicBuildAlias": "string",
            "requestedOn": number,
            "secondaryArtifacts": [
               {
                  "identifier": "string",
                  "location": "string",
                  "type": "string"
               }
            ]
         },
         "dependsOn": [ "string" ],
         "identifier": "string",
         "ignoreFailure": boolean,
         "priorBuildSummaryList": [
            {
               "buildStatus": "string",
               "primaryArtifact": {
                  "identifier": "string",
                  "location": "string",
                  "type": "string"
               },
               "publicBuildAlias": "string",
               "requestedOn": number,
```

```
                "secondaryArtifacts": [
                   {
                      "identifier": "string",
                      "location": "string",
                      "type": "string"
                   }
                ]
             }
          ]
       }
    ],
    "buildTimeoutInMinutes": number,
    "complete": boolean,
    "currentPhase": "string",
    "endTime": number,
    "environment": {
       "computeType": "string",
       "environmentVariables": [
          {
             "name": "string",
             "type": "string",
             "value": "string"
          }
       ],
       "image": "string",
       "type": "string"
    },
    "id": "string",
    "initiator": "string",
    "logsStatus": {
       "cloudWatchLogsStatus": "string",
       "s3LogsStatus": "string"
    },
    "phases": [
       {
          "contexts": [
             {
                "message": "string",
                "statusCode": "string"
             }
          ],
          "durationInSeconds": number,
          "endTime": number,
          "phaseStatus": "string",
```

```
            "phaseType": "string",
            "startTime": number
         }
      ],
      "projectName": "string",
      "publicBuildBatchAlias": "string",
      "queuedTimeoutInMinutes": number,
      "resolvedSourceVersion": "string",
      "secondaryArtifacts": [
         {
            "artifactIdentifier": "string",
            "packaging": "string",
            "type": "string"
         }
      ],
      "secondarySources": [
         {
            "buildspec": "string",
            "gitCloneDepth": number,
            "gitSubmodulesConfig": {
               "fetchSubmodules": boolean
            },
            "location": "string",
            "sourceIdentifier": "string",
            "type": "string"
         }
      ],
      "secondarySourceVersions": [
         {
            "sourceIdentifier": "string",
            "sourceVersion": "string"
         }
      ],
      "source": {
         "buildspec": "string",
         "gitCloneDepth": number,
         "gitSubmodulesConfig": {
            "fetchSubmodules": boolean
         },
         "location": "string",
         "sourceIdentifier": "string",
         "type": "string"
      },
      "sourceVersion": "string",
```

```
        "startTime": number
    }
}
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

publik BuildBatch

Tipe: Objek PublicBuildBatch

Kesalahan

Untuk informasi tentang kesalahan yang umum terjadi pada semua tindakan, lihat Kesalahan Umum.

InvalidInputException

Kode Status HTTP: 400

ResourceNotFoundException

Kode Status HTTP: 400

GetPublicProject

> **Note**
>
> Elemen API ini tidak terkandung dalam AWS CLI atau AWS SDK.

Sintaksis Permintaan

```
{
   "publicProjectAlias": "string"
}
```

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

Permintaan menerima data berikut dalam format JSON.

> **Note**
>
> Dalam daftar berikut, parameter yang diperlukan dijelaskan terlebih dahulu.

publik ProjectAlias

Tipe: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 1024.

Pola: `^[0-9a-zA-Z%+=]+$`

Diperlukan: Ya

Sintaksis Respons

```
{
   "project": {
      "artifacts": {
         "artifactIdentifier": "string",
         "location": "string",
```

```
      "type": "string"
   },
   "buildBatchConfig": {
      "combineArtifacts": boolean,
      "restrictions": {
         "computeTypesAllowed": [ "string" ],
         "maximumBuildsAllowed": number
      },
      "timeoutInMins": number
   },
   "concurrentBuildLimit": number,
   "description": "string",
   "environment": {
      "computeType": "string",
      "environmentVariables": [
         {
            "name": "string",
            "type": "string",
            "value": "string"
         }
      ],
      "image": "string",
      "type": "string"
   },
   "name": "string",
   "queuedTimeoutInMinutes": number,
   "secondaryArtifacts": [
      {
         "artifactIdentifier": "string",
         "location": "string",
         "type": "string"
      }
   ],
   "secondarySources": [
      {
         "buildspec": "string",
         "gitCloneDepth": number,
         "gitSubmodulesConfig": {
            "fetchSubmodules": boolean
         },
         "location": "string",
         "sourceIdentifier": "string",
         "type": "string"
      }
```

```
      ],
      "secondarySourceVersions": [
         {
            "sourceIdentifier": "string",
            "sourceVersion": "string"
         }
      ],
      "source": {
         "buildspec": "string",
         "gitCloneDepth": number,
         "gitSubmodulesConfig": {
            "fetchSubmodules": boolean
         },
         "location": "string",
         "sourceIdentifier": "string",
         "type": "string"
      },
      "sourceVersion": "string",
      "timeoutInMinutes": number,
      "webhook": {
         "branchFilter": "string",
         "buildType": "string",
         "filterGroups": [
            [
               {
                  "excludeMatchedPattern": boolean,
                  "pattern": "string",
                  "type": "string"
               }
            ]
         ],
         "payloadUrl": "string",
         "url": "string"
      }
   }
}
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

proyek

Tipe: Objek PublicProject

Kesalahan

Untuk informasi tentang kesalahan yang umum terjadi pada semua tindakan, lihat Kesalahan Umum.

InvalidInputException

Kode Status HTTP: 400

ResourceNotFoundException

Kode Status HTTP: 400

Tipe data build publik

> **Note**
>
> API build CodeBuild publik tidak terdapat dalam AWS CLI atau AWS SDK.

Tipe data berikut didukung oleh API build CodeBuild publik:

Jenis Data

- BuildBatchForDescribeBuildBatchesPublic
- PublicBuild
- PublicBuildArtifacts
- PublicBuildBatch
- PublicBuildGroup
- PublicBuildSummary
- PublicLogsStatus
- PublicProject
- PublicProjectArtifacts
- PublicProjectBuildBatchConfig

- PublicProjectEnvironment
- PublicProjectSource
- PublicWebhook
- S3Dapat Diunduh

BuildBatchForDescribeBuildBatchesPublic

> **Note**
>
> Elemen API ini tidak terkandung dalam AWS CLI atau AWS SDK.

Daftar Isi

> **Note**
>
> Dalam daftar berikut, parameter yang diperlukan dijelaskan terlebih dahulu.

membangun BatchNumber

Tipe: Panjang

Wajib: Tidak

membangun BatchStatus

Tipe: String

Nilai yang Valid: `PENDING | SUCCEEDED | FAILED | FAULT | TIMED_OUT | IN_PROGRESS | STOPPED`

Wajib: Tidak

EndTime

Tipe: Timestamp

Wajib: Tidak

publik BuildBatchAlias

Tipe: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 1024.

Pola: `^[0-9a-zA-Z%+=]+:[a-z0-9]{8}(-[a-z0-9]{4}){3}-[a-z0-9]{12}$`

Wajib: Tidak

SourceVersion

Tipe: String

Panjang Batasan: Panjang minimum 1.

Wajib: Tidak

startTime

Tipe: Timestamp

Wajib: Tidak

PublicBuild

> **Note**
>
> Elemen API ini tidak terkandung dalam AWS CLI atau AWS SDK.

Daftar Isi

> **Note**
>
> Dalam daftar berikut, parameter yang diperlukan dijelaskan terlebih dahulu.

artefak

Tipe: Objek PublicBuildArtifacts

Wajib: Tidak

BuildComplete

Tipe: Boolean

Wajib: Tidak

BuildNumber

Tipe: Panjang

Wajib: Tidak

BuildStatus

Tipe: String

Nilai yang Valid: `PENDING | SUCCEEDED | FAILED | FAULT | TIMED_OUT | IN_PROGRESS | STOPPED`

Wajib: Tidak

EndTime

Tipe: Timestamp

Wajib: Tidak

lingkungan

Tipe: Objek PublicProjectEnvironment

Wajib: Tidak

id

Tipe: String

Panjang Batasan: Panjang minimum 1.

Wajib: Tidak

inisiator

Tipe: String

Wajib: Tidak

LogStatus

Tipe: Objek PublicLogsStatus

Wajib: Tidak

fase

Tipe: Array objek BuildPhase

Wajib: Tidak

ProjectName

Tipe: String

Panjang Batasan: Panjang minimum 1.

Wajib: Tidak

mengantri TimeoutInMinutes

Tipe: Integer

Wajib: Tidak

diselesaikan SourceVersion

Tipe: String

Panjang Batasan: Panjang minimum 1.

Wajib: Tidak

SecondaryArtifacts

Tipe: Array objek PublicBuildArtifacts

Wajib: Tidak

SecondarySources

Tipe: Array objek PublicProjectSource

Anggota Array: Jumlah minimum 0 item. Jumlah maksimum 12 item.

Wajib: Tidak

sekunder SourceVersions

Tipe: Array objek ProjectSourceVersion

Anggota Array: Jumlah minimum 0 item. Jumlah maksimum 12 item.

Wajib: Tidak

sumber

Tipe: Objek PublicProjectSource

Wajib: Tidak

SourceVersion

Tipe: String

Panjang Batasan: Panjang minimum 1.

Wajib: Tidak

startTime

Tipe: Timestamp

Wajib: Tidak

batas waktu InMinutes

Tipe: Integer

Wajib: Tidak

PublicBuildArtifacts

> **Note**
>
> Elemen API ini tidak terkandung dalam AWS CLI atau AWS SDK.

Daftar Isi

> **Note**
>
> Dalam daftar berikut, parameter yang diperlukan dijelaskan terlebih dahulu.

ArtifactIdentifier

Tipe: String

Wajib: Tidak

Kemasan

Tipe: String

Wajib: Tidak

jenis

Tipe: String

Nilai yang Valid: `CODEPIPELINE | S3 | NO_ARTIFACTS | DOCKER_IMAGE`

Wajib: Tidak

PublicBuildBatch

> **Note**
>
> Elemen API ini tidak terkandung dalam AWS CLI atau AWS SDK.

Daftar Isi

> **Note**
>
> Dalam daftar berikut, parameter yang diperlukan dijelaskan terlebih dahulu.

artefak

Tipe: Objek PublicBuildArtifacts

Wajib: Tidak

membangun BatchConfig

Tipe: Objek PublicProjectBuildBatchConfig

Wajib: Tidak

membangun BatchNumber

Tipe: Panjang

Wajib: Tidak

membangun BatchStatus

Tipe: String

Nilai yang Valid: `PENDING | SUCCEEDED | FAILED | FAULT | TIMED_OUT | IN_PROGRESS | STOPPED`

Wajib: Tidak

BuildGroups

Tipe: Array objek PublicBuildGroup

Wajib: Tidak

membangun TimeoutInMinutes

Tipe: Integer

Wajib: Tidak

lengkap

Tipe: Boolean

Wajib: Tidak

Fase saat ini

Tipe: String

Wajib: Tidak

EndTime

Tipe: Timestamp

Wajib: Tidak

lingkungan

Tipe: Objek PublicProjectEnvironment

Wajib: Tidak

id

Tipe: String

Panjang Batasan: Panjang minimum 1.

Wajib: Tidak

inisiator

Tipe: String

Wajib: Tidak

LogStatus

Tipe: Objek PublicLogsStatus

Wajib: Tidak

fase

Tipe: Array objek BuildBatchPhase

Wajib: Tidak

ProjectName

Tipe: String

Panjang Batasan: Panjang minimum 1.

Wajib: Tidak

publik BuildBatchAlias

Tipe: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 1024.

Pola: `^[0-9a-zA-Z%+=]+:[a-z0-9]{8}(-[a-z0-9]{4}){3}-[a-z0-9]{12}$`

Wajib: Tidak

mengantri TimeoutInMinutes

Tipe: Integer

Wajib: Tidak

diselesaikan SourceVersion

Tipe: String

Panjang Batasan: Panjang minimum 1.

Wajib: Tidak

SecondaryArtifacts

Tipe: Array objek PublicBuildArtifacts

Wajib: Tidak

SecondarySources

Tipe: Array objek PublicProjectSource

Anggota Array: Jumlah minimum 0 item. Jumlah maksimum 12 item.

Wajib: Tidak

sekunder SourceVersions

Tipe: Array objek ProjectSourceVersion

Anggota Array: Jumlah minimum 0 item. Jumlah maksimum 12 item.

Wajib: Tidak

sumber

Tipe: Objek PublicProjectSource

Wajib: Tidak

SourceVersion

Tipe: String

Panjang Batasan: Panjang minimum 1.

Wajib: Tidak

startTime

Tipe: Timestamp

Wajib: Tidak

PublicBuildGroup

> **Note**
>
> Elemen API ini tidak terkandung dalam AWS CLI atau AWS SDK.

Daftar Isi

> **Note**
>
> Dalam daftar berikut, parameter yang diperlukan dijelaskan terlebih dahulu.

saat ini BuildSummary

Tipe: Objek PublicBuildSummary

Wajib: Tidak

DependsOn

Tipe: Array string

Batasan Panjang: Panjang minimum 1.

Wajib: Tidak

pengenal

Tipe: String

Wajib: Tidak

IgkanFailure

Tipe: Boolean

Wajib: Tidak

sebelumnya BuildSummaryList

Tipe: Array objek PublicBuildSummary

Wajib: Tidak

PublicBuildSummary

> **Note**
>
> Elemen API ini tidak terkandung dalam AWS CLI AWS SDK or.

Daftar Isi

> **Note**
>
> Dalam daftar berikut, parameter yang diperlukan dijelaskan terlebih dahulu.

Status Bangun

Tipe: String

Wajib: Tidak

Artefak Utama

Merupakan artefak build yang diselesaikan. Artefak yang diselesaikan adalah artefak yang dibangun dan digunakan ke tujuan, seperti Amazon S3.

Tipe: Objek ResolvedArtifact

Wajib: Tidak

publik BuildAlias

Tipe: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 1024.

Pola: `^[0-9a-zA-Z%+=]+:\p{XDigit}{8}(-\p{XDigit}{4}){3}-\p{XDigit}{12}$`

Wajib: Tidak

PermintaPada

Tipe: Timestamp

Wajib: Tidak

Artefak sekunder

Tipe: Array objek ResolvedArtifact

Wajib: Tidak

PublicLogsStatus

> **Note**
>
> Elemen API ini tidak terkandung dalam AWS CLI atau AWS SDK.

Daftar Isi

> **Note**
>
> Dalam daftar berikut, parameter yang diperlukan dijelaskan terlebih dahulu.

awan WatchLogsStatus

Tipe: String

Nilai yang Valid: `ENABLED | DISABLED`

Wajib: Tidak

s3LogsStatus

Tipe: String

Nilai yang Valid: `ENABLED | DISABLED`

Wajib: Tidak

PublicProject

> **Note**
>
> Elemen API ini tidak terkandung dalam AWS CLI atau AWS SDK.

Daftar Isi

> **Note**
>
> Dalam daftar berikut, parameter yang diperlukan dijelaskan terlebih dahulu.

artefak

Tipe: Objek PublicProjectArtifacts

Wajib: Tidak

membangun BatchConfig

Tipe: Objek PublicProjectBuildBatchConfig

Wajib: Tidak

bersamaan BuildLimit

Tipe: Integer

Wajib: Tidak

deskripsi

Tipe: String

Batasan Panjang: Panjang minimum 0. Panjang maksimum sebesar 255.

Wajib: Tidak

lingkungan

Tipe: Objek PublicProjectEnvironment

Wajib: Tidak

name

Tipe: String

Kendala Panjang: Panjang minimum 2. Panjang maksimum 150.

Pola: `[A-Za-z0-9][A-Za-z0-9\-_]{1,254}`

Wajib: Tidak

mengantri TimeoutInMinutes

Jenis: Integer

Rentang Valid: Nilai minimum 5. Nilai maksimum 480.

Wajib: Tidak

SecondaryArtifacts

Tipe: Array objek PublicProjectArtifacts

Anggota Array: Jumlah minimum 0 item. Jumlah maksimum 12 item.

Wajib: Tidak

SecondarySources

Tipe: Array objek PublicProjectSource

Anggota Array: Jumlah minimum 0 item. Jumlah maksimum 12 item.

Wajib: Tidak

sekunder SourceVersions

Tipe: Array objek ProjectSourceVersion

Anggota Array: Jumlah minimum 0 item. Jumlah maksimum 12 item.

Wajib: Tidak

sumber

Tipe: Objek PublicProjectSource

Wajib: Tidak

SourceVersion

Tipe: String

Panjang Batasan: Panjang minimum 1.

Wajib: Tidak

batas waktu InMinutes

Jenis: Integer

Rentang Valid: Nilai minimum 5. Nilai maksimum 480.

Wajib: Tidak

webhook

Tipe: Objek PublicWebhook

Wajib: Tidak

PublicProjectArtifacts

> **Note**
>
> Elemen API ini tidak terkandung dalam AWS CLI atau AWS SDK.

Daftar Isi

> **Note**
>
> Dalam daftar berikut, parameter yang diperlukan dijelaskan terlebih dahulu.

jenis

Tipe: String

Nilai Valid: `CODEPIPELINE | S3 | NO_ARTIFACTS | DOCKER_IMAGE`

Wajib: Ya

ArtifactIdentifier

Tipe: String

Wajib: Tidak

lokasi

Tipe: String

Wajib: Tidak

PublicProjectBuildBatchConfig

> **Note**
>
> Elemen API ini tidak terkandung dalam AWS CLI atau AWS SDK.

Daftar Isi

> **Note**
>
> Dalam daftar berikut, parameter yang diperlukan dijelaskan terlebih dahulu.

CombineArtiFacts

Tipe: Boolean

Wajib: Tidak

pembatasan

Menentukan batasan untuk pembangunan batch.

Tipe: Objek BatchRestrictions

Wajib: Tidak

batas waktu InMins

Tipe: Integer

Wajib: Tidak

PublicProjectEnvironment

> **Note**
>
> Elemen API ini tidak terkandung dalam AWS CLI atau AWS SDK.

Daftar Isi

> **Note**
>
> Dalam daftar berikut, parameter yang diperlukan dijelaskan terlebih dahulu.

ComputeType

Tipe: String

Nilai yang Valid: `BUILD_GENERAL1_SMALL | BUILD_GENERAL1_MEDIUM | BUILD_GENERAL1_LARGE | BUILD_GENERAL1_2XLARGE`

Wajib: Tidak

Variabel Lingkungan

Tipe: Array objek EnvironmentVariable

Wajib: Tidak

gambar

Tipe: String

Panjang Batasan: Panjang minimum 1.

Wajib: Tidak

jenis

Tipe: String

Nilai yang Valid: `WINDOWS_CONTAINER | WINDOWS_SERVER_2019_CONTAINER | LINUX_CONTAINER | LINUX_GPU_CONTAINER | ARM_CONTAINER | MAC`

Wajib: Tidak

PublicProjectSource

> **Note**
>
> Elemen API ini tidak terkandung dalam AWS CLI atau AWS SDK.

Daftar Isi

> **Note**
>
> Dalam daftar berikut, parameter yang diperlukan dijelaskan terlebih dahulu.

jenis

Tipe: String

Nilai Valid: `NO_SOURCE | CODECOMMIT | CODEPIPELINE | GITHUB | S3 | BITBUCKET | GITHUB_ENTERPRISE | GITLAB | GITLAB_SELF_MANAGED`

Wajib: Ya

buildspec

Tipe: String

Wajib: Tidak

git CloneDepth

Jenis: Integer

Rentang yang Valid: Nilai minimum 0.

Wajib: Tidak

git SubmodulesConfig

Informasi tentang konfigurasi submodul Git untuk proyek AWS CodeBuild build.

Tipe: Objek GitSubmodulesConfig

Wajib: Tidak

lokasi

Tipe: String

Wajib: Tidak

SourceIdentifier

Tipe: String

Wajib: Tidak

PublicWebhook

> **Note**
>
> Elemen API ini tidak terkandung dalam AWS CLI atau AWS SDK.

Daftar Isi

> **Note**
>
> Dalam daftar berikut, parameter yang diperlukan dijelaskan terlebih dahulu.

Cabang Filter

Tipe: String

Wajib: Tidak

BuildType

Tipe: String

Wajib: Tidak

FilterGroups

Jenis: Array array objek WebhookFilter

Wajib: Tidak

PayloadUrl

Tipe: String

Panjang Batasan: Panjang minimum 1.

Wajib: Tidak

url

Tipe: String

Panjang Batasan: Panjang minimum 1.

Wajib: Tidak

S3Dapat Diunduh

> **Note**
>
> Elemen API ini tidak terkandung dalam AWS CLI atau AWS SDK.

Daftar Isi

> **Note**
>
> Dalam daftar berikut, parameter yang diperlukan dijelaskan terlebih dahulu.

KedaluwarsaAT

Tipe: Timestamp

Wajib: Tidak

pengenal

Tipe: String

Wajib: Tidak

MD5Checksum

Tipe: String

Wajib: Tidak

PresignedUrl

Tipe: String

Wajib: Tidak

S3arn

Tipe: String

Wajib: Tidak

SHA256Checksum

Tipe: String

Wajib: Tidak

Pemecahan Masalah AWS CodeBuild

Gunakan informasi dalam topik ini untuk membantu Anda mengidentifikasi, mendiagnosis, dan mengatasi masalah. Untuk mempelajari cara mencatat dan memantau CodeBuild build untuk memecahkan masalah, lihat. Pencatatan log dan pemantauan

Topik

- Apache Maven membangun artefak referensi dari repositori yang salah
- Perintah build dijalankan sebagai root secara default
- Build mungkin gagal ketika nama file tidak memiliki. U.S Karakter bahasa Inggris
- Build mungkin gagal saat mendapatkan parameter dari Amazon EC2 Parameter Store
- Tidak dapat mengakses filter cabang di CodeBuild konsol
- Tidak dapat melihat keberhasilan atau kegagalan bangunan
- Status build tidak dilaporkan ke penyedia sumber
- Tidak dapat menemukan dan memilih gambar dasar platform Windows Server Core 2019
- Perintah sebelumnya dalam file buildspec tidak dikenali oleh perintah selanjutnya
- Kesalahan: “Akses ditolak” saat mencoba mengunduh cache
- Kesalahan: “BUILD_CONTAINER_UNABLE_TO_PULL_IMAGE” saat menggunakan gambar build khusus
- Kesalahan: “Build container ditemukan mati sebelum menyelesaikan build. build container mati karena kehabisan memori, atau image Docker tidak didukung. ErrorCode: 500”
- Kesalahan: “Tidak dapat terhubung ke daemon Docker” saat menjalankan build
- Kesalahan: "CodeBuild tidak diizinkan untuk melakukan: sts:AssumeRole" saat membuat atau memperbarui proyek build
- Kesalahan: “Kesalahan memanggil GetBucketAcl: Pemilik bucket telah berubah atau peran layanan tidak lagi memiliki izin untuk memanggil s3:GetBucketAcl”
- Kesalahan: “Gagal mengunggah artefak: Arn tidak valid” saat menjalankan build
- Kesalahan: “Gagal klon Git: Tidak dapat mengakses 'URL repositori-Anda': Masalah sertifikat SSL: Sertifikat yang ditandatangani sendiri”
- Kesalahan: “Bucket yang Anda coba akses harus ditangani menggunakan titik akhir yang ditentukan” saat menjalankan build
- Kesalahan: “Gambar build ini memerlukan pemilihan setidaknya satu versi runtime.”

- Kesalahan: “QUEUED: INSUFFICIENT_SUBNET” saat build dalam antrian build gagal
- Kesalahan: “Tidak dapat mengunduh cache: RequestError: Permintaan kirim gagal disebabkan oleh: x509: Gagal memuat akar sistem dan tidak ada akar yang disediakan”
- Kesalahan: “Tidak dapat mengunduh sertifikat dari S3. AccessDenied”
- Kesalahan: “Tidak dapat menemukan kredensil”
- RequestError kesalahan batas waktu saat berjalan CodeBuild di server proxy
- Cangkang bourne (sh) harus ada dalam gambar build
- Peringatan: “Melewatkan pemasangan runtime. pemilihan versi runtime tidak didukung oleh image build ini” saat menjalankan build
- Kesalahan: “Tidak dapat memverifikasi JobWorker identitas” saat membuka CodeBuild konsol
- Build gagal untuk memulai
- Mengakses GitHub metadata dalam build yang di-cache secara lokal
- AccessDenied: Pemilik bucket untuk grup laporan tidak cocok dengan pemilik bucket S3...
- Kesalahan: “Kredensyal Anda tidak memiliki satu atau lebih cakupan hak istimewa yang diperlukan” saat membuat proyek dengan CodeBuild AWS CodeConnections
- Kesalahan: “Maaf, tidak ada terminal sama sekali yang diminta - tidak bisa mendapatkan input” saat membangun dengan perintah instal Ubuntu
- Bitbucket build gagal dengan kesalahan otentikasi setelah menggunakan Secrets Manager OAuth

Apache Maven membangun artefak referensi dari repositori yang salah

Masalah: Saat Anda menggunakan Maven dengan lingkungan build Java yang AWS CodeBuild disediakan, Maven menarik dependensi build dan plugin dari repositori Maven pusat yang aman di. https://repo1.maven.org/maven2 Ini terjadi bahkan jika `pom.xml` file proyek build Anda secara eksplisit mendeklarasikan lokasi lain untuk digunakan sebagai gantinya.

Kemungkinan penyebabnya: Lingkungan build CodeBuild-provided Java menyertakan file bernama `settings.xml` yang sudah diinstal sebelumnya di `/root/.m2` direktori lingkungan build. `settings.xml`File ini berisi deklarasi berikut, yang menginstruksikan Maven untuk selalu menarik dependensi build dan plugin dari repositori Maven pusat aman di. https://repo1.maven.org/maven2

```
<settings>
  <activeProfiles>
```

```
    <activeProfile>securecentral</activeProfile>
  </activeProfiles>
  <profiles>
    <profile>
      <id>securecentral</id>
      <repositories>
        <repository>
          <id>central</id>
          <url>https://repo1.maven.org/maven2</url>
          <releases>
            <enabled>true</enabled>
          </releases>
        </repository>
      </repositories>
      <pluginRepositories>
        <pluginRepository>
          <id>central</id>
          <url>https://repo1.maven.org/maven2</url>
          <releases>
            <enabled>true</enabled>
          </releases>
        </pluginRepository>
      </pluginRepositories>
    </profile>
  </profiles>
</settings>
```

Solusi yang disarankan: Lakukan hal berikut:

1. Tambahkan `settings.xml` file ke kode sumber Anda.
2. Dalam `settings.xml` file ini, gunakan `settings.xml` format sebelumnya sebagai panduan untuk mendeklarasikan repositori yang Anda inginkan Maven untuk menarik dependensi build dan plugin.
3. Pada `install` fase proyek build Anda, instruksikan CodeBuild untuk menyalin `settings.xml` file Anda ke `/root/.m2` direktori lingkungan build. Misalnya, pertimbangkan cuplikan berikut dari `buildspec.yml` file yang menunjukkan perilaku ini.

   ```
   version 0.2

   phases:
     install:
       commands:
   ```

```
      - cp ./settings.xml /root/.m2/settings.xml
```

Perintah build dijalankan sebagai root secara default

Masalah: AWS CodeBuild menjalankan perintah build Anda sebagai pengguna root. Ini terjadi bahkan jika Dockerfile image build terkait Anda menyetel `USER` instruksi ke pengguna yang berbeda.

Penyebab: Secara default, CodeBuild menjalankan semua perintah build sebagai pengguna root.

Solusi yang disarankan: Tidak ada.

Build mungkin gagal ketika nama file tidak memiliki. U.S Karakter bahasa Inggris

Masalah: Saat Anda menjalankan build yang menggunakan file dengan nama file yang berisi non-U.S. Karakter bahasa Inggris (misalnya, karakter Mandarin), build gagal.

Kemungkinan penyebabnya: Lingkungan build yang disediakan oleh AWS CodeBuild mengatur lokal defaultnya. `POSIX` `POSIX`pengaturan lokalisasi kurang kompatibel dengan CodeBuild dan nama file yang berisi non-U.S. Karakter bahasa Inggris dan dapat menyebabkan build terkait gagal.

Solusi yang disarankan: Tambahkan perintah berikut ke `pre_build` bagian file buildspec Anda. Perintah ini membuat lingkungan build digunakan U.S. Bahasa Inggris UTF-8 untuk pengaturan lokalisasi, yang lebih kompatibel dengan CodeBuild dan nama file yang berisi non-U.S. Karakter bahasa Inggris.

Untuk lingkungan build berdasarkan Ubuntu:

```
pre_build:
  commands:
    - export LC_ALL="en_US.UTF-8"
    - locale-gen en_US en_US.UTF-8
    - dpkg-reconfigure -f noninteractive locales
```

Untuk lingkungan build berbasis Amazon Linux:

```
pre_build:
  commands:
```

```
    - export LC_ALL="en_US.utf8"
```

Build mungkin gagal saat mendapatkan parameter dari Amazon EC2 Parameter Store

Masalah: Saat build mencoba mendapatkan nilai dari satu atau beberapa parameter yang disimpan di Amazon EC2 Parameter Store, build gagal dalam `DOWNLOAD_SOURCE` fase dengan kesalahan. `Parameter does not exist`

Kemungkinan penyebabnya: Peran layanan yang diandalkan proyek build tidak memiliki izin untuk memanggil `ssm:GetParameters` tindakan atau proyek build menggunakan peran layanan yang dihasilkan oleh AWS CodeBuild dan memungkinkan pemanggilan `ssm:GetParameters` tindakan, tetapi parameter memiliki nama yang tidak dimulai`/CodeBuild/`.

Solusi yang direkomendasikan:

- Jika peran layanan tidak dihasilkan oleh CodeBuild, perbarui definisinya untuk memungkinkan CodeBuild untuk memanggil `ssm:GetParameters` tindakan. Misalnya, pernyataan kebijakan berikut memungkinkan pemanggilan `ssm:GetParameters` tindakan untuk mendapatkan parameter dengan nama yang dimulai dengan`/CodeBuild/`:

  JSON

  ```
  {
      "Version":"2012-10-17",
      "Statement": [
          {
              "Action": "ssm:GetParameters",
              "Effect": "Allow",
              "Resource": "arn:aws:ssm:us-east-1:111122223333:parameter/
  CodeBuild/*"
          }
      ]
  }
  ```

- Jika peran layanan dihasilkan oleh CodeBuild, perbarui definisinya untuk memungkinkan CodeBuild untuk mengakses parameter di Amazon EC2 Parameter Store dengan nama selain yang dimulai dengan. `/CodeBuild/` Misalnya, pernyataan kebijakan berikut memungkinkan pemanggilan `ssm:GetParameters` tindakan untuk mendapatkan parameter dengan nama yang ditentukan:

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Action": "ssm:GetParameters",
            "Effect": "Allow",
            "Resource": "arn:aws:ssm:us-
east-1:111122223333:parameter/PARAMETER_NAME"
        }
    ]
}
```

Tidak dapat mengakses filter cabang di CodeBuild konsol

Masalah: Opsi filter cabang tidak tersedia di konsol saat Anda membuat atau memperbarui AWS CodeBuild proyek.

Kemungkinan penyebabnya: Opsi filter cabang tidak digunakan lagi. Ini telah digantikan oleh grup filter webhook, yang memberikan kontrol lebih besar atas peristiwa webhook yang memicu build baru. CodeBuild

Solusi yang disarankan: Untuk memigrasikan filter cabang yang Anda buat sebelum pengenalan filter webhook, buat grup filter webhook dengan `HEAD_REF` filter dengan ekspresi reguler. `^refs/heads/branchName$` Misalnya, jika ekspresi reguler filter cabang Anda adalah`^branchName$`, maka ekspresi reguler yang diperbarui yang Anda masukkan ke dalam `HEAD_REF` filter adalah`^refs/heads/branchName$`. Untuk informasi selengkapnya, lihat Acara webhook Bitbucket dan Filter acara GitHub webhook (konsol).

Tidak dapat melihat keberhasilan atau kegagalan bangunan

Masalah: Anda tidak dapat melihat keberhasilan atau kegagalan build yang dicoba lagi.

Kemungkinan penyebabnya: Opsi untuk melaporkan status build Anda tidak diaktifkan.

Solusi yang disarankan: Aktifkan status pembuatan Laporan saat Anda membuat atau memperbarui CodeBuild proyek. Opsi ini memberi tahu CodeBuild untuk melaporkan kembali status saat Anda

memicu build. Untuk informasi selengkapnya, lihat laporan BuildStatus di Referensi AWS CodeBuild API.

Status build tidak dilaporkan ke penyedia sumber

Masalah: Setelah mengizinkan pelaporan status build ke penyedia sumber, seperti GitHub atau Bitbucket, status build tidak diperbarui.

Kemungkinan penyebabnya: Pengguna yang terkait dengan penyedia sumber tidak memiliki akses tulis ke repo.

Solusi yang disarankan: Untuk dapat melaporkan status build ke penyedia sumber, pengguna yang terkait dengan penyedia sumber harus memiliki akses tulis ke repo. Jika pengguna tidak memiliki akses tulis, status build tidak dapat diperbarui. Untuk informasi selengkapnya, lihat Akses penyedia sumber.

Tidak dapat menemukan dan memilih gambar dasar platform Windows Server Core 2019

Masalah: Anda tidak dapat menemukan atau memilih gambar dasar platform Windows Server Core 2019.

Kemungkinan penyebabnya: Anda menggunakan AWS Wilayah yang tidak mendukung gambar ini.

Solusi yang disarankan: Gunakan salah satu AWS Wilayah berikut di mana gambar dasar platform Windows Server Core 2019 didukung:

- Timur AS (N. Virginia)
- AS Timur (Ohio)
- AS Barat (Oregon)
- Eropa (Irlandia)

Perintah sebelumnya dalam file buildspec tidak dikenali oleh perintah selanjutnya

Masalah: Hasil dari satu atau beberapa perintah dalam file buildspec Anda tidak dikenali oleh perintah selanjutnya dalam file buildspec yang sama. Misalnya, perintah mungkin menyetel variabel

lingkungan lokal, tetapi perintah yang dijalankan nanti mungkin gagal mendapatkan nilai variabel lingkungan lokal tersebut.

Kemungkinan penyebabnya: Dalam file buildspec versi 0.1, AWS CodeBuild jalankan setiap perintah dalam instance terpisah dari shell default di lingkungan build. Ini berarti bahwa setiap perintah berjalan secara terpisah dari semua perintah lainnya. Secara default, Anda tidak dapat menjalankan satu perintah yang bergantung pada status perintah sebelumnya.

Solusi yang disarankan: Kami menyarankan Anda menggunakan buildspec versi 0.2, yang memecahkan masalah ini. Jika Anda harus menggunakan buildspec versi 0.1, kami sarankan Anda menggunakan operator rantai perintah shell (misalnya, && di Linux) untuk menggabungkan beberapa perintah menjadi satu perintah. Atau sertakan skrip shell dalam kode sumber Anda yang berisi beberapa perintah, lalu panggil skrip shell itu dari satu perintah di file buildspec. Untuk informasi selengkapnya, lihat Shell dan perintah di lingkungan build dan Variabel lingkungan di lingkungan build.

Kesalahan: “Akses ditolak” saat mencoba mengunduh cache

Masalah: Saat mencoba mengunduh cache pada proyek build yang memiliki cache diaktifkan, Anda menerima `Access denied` kesalahan.

Kemungkinan penyebabnya:

- Anda baru saja mengonfigurasi caching sebagai bagian dari proyek build Anda.
- Cache baru-baru ini dibatalkan melalui API. `InvalidateProjectCache`
- Peran layanan yang digunakan oleh CodeBuild tidak memiliki `s3:GetObject` dan `s3:PutObject` izin ke bucket S3 yang menahan cache.

Solusi yang disarankan: Untuk penggunaan pertama kali, normal untuk melihat ini segera setelah memperbarui konfigurasi cache. Jika kesalahan ini berlanjut, maka Anda harus memeriksa untuk melihat apakah peran layanan Anda memiliki `s3:GetObject` dan `s3:PutObject` izin ke bucket S3 yang menahan cache. Untuk informasi selengkapnya, lihat Menentukan izin S3 di Panduan Pengembang Amazon S3.

Kesalahan: “BUILD_CONTAINER_UNABLE_TO_PULL_IMAGE” saat menggunakan gambar build khusus

Masalah: Saat Anda mencoba menjalankan build yang menggunakan image build kustom, build gagal dengan kesalahan tersebut`BUILD_CONTAINER_UNABLE_TO_PULL_IMAGE`.

Kemungkinan penyebabnya: Ukuran keseluruhan gambar build yang tidak terkompresi lebih besar daripada ruang disk tipe komputasi lingkungan build yang tersedia. Untuk memeriksa ukuran image build Anda, gunakan Docker untuk menjalankan *`docker images REPOSITORY:TAG`* perintah. Untuk daftar ruang disk yang tersedia menurut jenis komputasi, lihatMembangun mode dan tipe komputasi lingkungan.

> Solusi yang disarankan: Gunakan jenis komputasi yang lebih besar dengan lebih banyak ruang disk yang tersedia, atau kurangi ukuran image build kustom Anda.

Kemungkinan penyebabnya: AWS CodeBuild tidak memiliki izin untuk menarik image build dari Amazon Elastic Container Registry (Amazon ECR).

> Solusi yang disarankan: Perbarui izin di repositori Anda di Amazon ECR sehingga CodeBuild dapat menarik image build kustom Anda ke lingkungan build. Untuk informasi selengkapnya, lihat Sampel Amazon ECR.

Kemungkinan penyebabnya: Gambar Amazon ECR yang Anda minta tidak tersedia di AWS Wilayah yang digunakan AWS akun Anda.

> Solusi yang disarankan: Gunakan gambar Amazon ECR yang berada di AWS Wilayah yang sama dengan yang digunakan AWS akun Anda.

Kemungkinan penyebabnya: Anda menggunakan registri pribadi di VPC yang tidak memiliki akses internet publik. CodeBuild tidak dapat menarik gambar dari alamat IP pribadi di VPC. Untuk informasi selengkapnya, lihat Registri pribadi dengan AWS Secrets Manager sampel untuk CodeBuild.

> Solusi yang disarankan: Jika Anda menggunakan registri pribadi di VPC, pastikan VPC memiliki akses internet publik.

Kemungkinan penyebabnya: Jika pesan kesalahan berisi "toomanyrequests“, dan gambar diperoleh dari Docker Hub, kesalahan ini berarti batas tarik Docker Hub telah tercapai.

> Solusi yang disarankan: Gunakan registri pribadi Docker Hub, atau dapatkan gambar Anda dari Amazon ECR. Untuk informasi selengkapnya tentang menggunakan registri pribadi, lihatRegistri pribadi dengan AWS Secrets Manager sampel untuk CodeBuild. Untuk informasi selengkapnya tentang menggunakan Amazon ECR, lihatSampel Amazon ECR untuk CodeBuild.

Kesalahan: “Build container ditemukan mati sebelum menyelesaikan build. build container mati karena kehabisan memori, atau image Docker tidak didukung. ErrorCode: 500”

Masalah: Saat Anda mencoba menggunakan wadah Microsoft Windows atau Linux AWS CodeBuild, kesalahan ini terjadi selama fase PENYEDIAAN.

Kemungkinan penyebabnya:

- Versi OS kontainer tidak didukung oleh CodeBuild.
- `HTTP_PROXY`,`HTTPS_PROXY`, atau keduanya ditentukan dalam wadah.

Solusi yang direkomendasikan:

- Untuk Microsoft Windows, gunakan wadah Windows dengan OS kontainer yang versi:10.0.x microsoft/windowsservercore (misalnya ,:10.0.14393.2125). microsoft/windowsservercore
- Untuk Linux, hapus `HTTPS_PROXY` pengaturan `HTTP_PROXY` dan pada image Docker Anda, atau tentukan konfigurasi VPC di project build Anda.

Kesalahan: “Tidak dapat terhubung ke daemon Docker” saat menjalankan build

Masalah: Build Anda gagal dan Anda menerima kesalahan yang mirip dengan `Cannot connect to the Docker daemon at unix:/var/run/docker.sock. Is the docker daemon running?` di log build.

Kemungkinan penyebabnya: Anda tidak menjalankan build dalam mode istimewa.

Solusi yang disarankan: Untuk memperbaiki kesalahan ini, Anda harus mengaktifkan mode istimewa dan memperbarui buildspec Anda menggunakan instruksi berikut.

Untuk menjalankan build Anda dalam mode istimewa, ikuti langkah-langkah berikut:

1. Buka CodeBuild konsol di https://console.aws.amazon.com/codebuild/.
2. Di panel navigasi, pilih Build projects, lalu pilih build project Anda.
3. Dari Edit, pilih Lingkungan.

4. Pilih Konfigurasi tambahan.
5. Dari Privileged, pilih Aktifkan bendera ini jika Anda ingin membuat gambar Docker atau ingin build Anda mendapatkan hak istimewa yang lebih tinggi. .
6. Pilih Perbarui lingkungan.
7. Pilih Mulai membangun untuk mencoba lagi build Anda.

Anda juga harus memulai daemon Docker di dalam wadah Anda. `install`Fase buildspec Anda mungkin terlihat mirip dengan ini.

```
phases:
  install:
    commands:
      - nohup /usr/local/bin/dockerd --host=unix:///var/run/docker.sock --
host=tcp://127.0.0.1:2375 --storage-driver=overlay2 &
      - timeout 15 sh -c "until docker info; do echo .; sleep 1; done"
```

Untuk informasi selengkapnya tentang driver penyimpanan OverlayFS yang direferensikan dalam file buildspec, lihat Menggunakan driver penyimpanan OverlayFS di situs web Docker.

> **Note**
>
> Jika sistem operasi dasar adalah Alpine Linux, `buildspec.yml` tambahkan `-t` argumen `ketimeout`:
>
> ```
> - timeout -t 15 sh -c "until docker info; do echo .; sleep 1; done"
> ```

Untuk mempelajari lebih lanjut tentang cara membuat dan menjalankan image Docker dengan menggunakan AWS CodeBuild, lihatDocker dalam sampel gambar khusus untuk CodeBuild.

Kesalahan: "CodeBuild tidak diizinkan untuk melakukan: sts:AssumeRole" saat membuat atau memperbarui proyek build

Masalah: Saat Anda mencoba membuat atau memperbarui proyek build, Anda menerima kesalahan`Code:InvalidInputException, Message:CodeBuild is not authorized to perform: sts:AssumeRole on arn:aws:iam::`*`account-ID`*`:role/`*`service-role-name`*.

Kemungkinan penyebabnya:

- The AWS Security Token Service (AWS STS) telah dinonaktifkan untuk AWS wilayah tempat Anda mencoba membuat atau memperbarui proyek build.
- Peran AWS CodeBuild layanan yang terkait dengan proyek build tidak ada atau tidak memiliki izin yang cukup untuk dipercaya CodeBuild.
- Casing peran AWS CodeBuild layanan yang terkait dengan proyek build tidak cocok dengan peran IAM yang sebenarnya.

Solusi yang direkomendasikan:

- Pastikan AWS STS diaktifkan untuk AWS wilayah tempat Anda mencoba membuat atau memperbarui proyek build. Untuk informasi selengkapnya, lihat Mengaktifkan dan menonaktifkan AWS STS di AWS Wilayah di Panduan Pengguna IAM.
- Pastikan peran CodeBuild layanan target ada di AWS akun Anda. Jika Anda tidak menggunakan konsol, pastikan Anda tidak salah mengeja Nama Sumber Daya Amazon (ARN) peran layanan saat Anda membuat atau memperbarui proyek build. Perhatikan bahwa peran IAM peka huruf besar/kecil, jadi periksa apakah casing peran IAM sudah benar.
- Pastikan peran CodeBuild layanan target memiliki izin yang cukup untuk dipercaya CodeBuild. Untuk informasi selengkapnya, lihat pernyataan kebijakan hubungan kepercayaan diMemungkinkan CodeBuild untuk berinteraksi dengan orang lain AWS layanan.

Kesalahan: “Kesalahan memanggil GetBucketAcl: Pemilik bucket telah berubah atau peran layanan tidak lagi memiliki izin untuk memanggil s3:GetBucketAcl”

Masalah: Saat menjalankan build, Anda menerima error tentang perubahan kepemilikan bucket dan `GetBucketAcl` izin S3.

Kemungkinan penyebabnya: Anda menambahkan `s3:GetBucketAcl` dan `s3:GetBucketLocation` izin ke peran IAM Anda. Izin ini mengamankan bucket S3 proyek Anda dan memastikan bahwa hanya Anda yang dapat mengaksesnya. Setelah Anda menambahkan izin ini, pemilik bucket S3 berubah.

Solusi yang disarankan: Pastikan Anda adalah pemilik bucket S3, lalu tambahkan izin ke peran IAM Anda lagi. Untuk informasi selengkapnya, lihat Akses aman ke bucket S3.

Kesalahan: “Gagal mengunggah artefak: Arn tidak valid” saat menjalankan build

Masalah: Saat Anda menjalankan build, fase `UPLOAD_ARTIFACTS` build gagal dengan kesalahan`Failed to upload artifacts: Invalid arn`.

Kemungkinan penyebabnya: Bucket keluaran S3 Anda (bucket tempat AWS CodeBuild menyimpan outputnya dari build) berada di AWS Wilayah yang berbeda dari proyek CodeBuild build.

Solusi yang disarankan: Perbarui pengaturan proyek build untuk menunjuk ke bucket keluaran yang berada di AWS Wilayah yang sama dengan proyek build.

Kesalahan: “Gagal klon Git: Tidak dapat mengakses **`'URL repositori-Anda'`**: Masalah sertifikat SSL: Sertifikat yang ditandatangani sendiri”

Masalah: Saat Anda mencoba menjalankan proyek build, build gagal dengan kesalahan ini.

Kemungkinan penyebabnya: Repositori sumber Anda memiliki sertifikat yang ditandatangani sendiri, tetapi Anda belum memilih untuk menginstal sertifikat dari bucket S3 sebagai bagian dari proyek build Anda.

Solusi yang direkomendasikan:

- Edit proyek Anda. Untuk Sertifikat, pilih Instal sertifikat dari S3. Untuk Bucket of certificate, pilih bucket S3 tempat sertifikat SSL Anda disimpan. Untuk kunci Objek sertifikat, masukkan nama kunci objek S3 Anda.
- Edit proyek Anda. Pilih SSL Tidak Aman untuk mengabaikan peringatan SSL saat menghubungkan ke repositori proyek GitHub Enterprise Server Anda.

> **Note**
>
> Kami menyarankan Anda menggunakan SSL Insecure hanya untuk pengujian. Ini tidak boleh digunakan di lingkungan produksi.

Kesalahan: “Bucket yang Anda coba akses harus ditangani menggunakan titik akhir yang ditentukan” saat menjalankan build

Masalah: Saat Anda menjalankan build, fase DOWNLOAD_SOURCE build gagal dengan kesalahanThe `bucket you are attempting to access must be addressed using the specified endpoint. Please send all future requests to this endpoint`.

Kemungkinan penyebabnya: Kode sumber bawaan Anda disimpan dalam bucket S3, dan bucket tersebut berada di AWS Wilayah yang berbeda dari proyek AWS CodeBuild build.

Solusi yang disarankan: Perbarui pengaturan proyek build untuk menunjuk ke bucket yang berisi kode sumber bawaan Anda. Pastikan bucket berada di AWS Region yang sama dengan proyek build.

Kesalahan: “Gambar build ini memerlukan pemilihan setidaknya satu versi runtime.”

Masalah: Saat Anda menjalankan build, fase DOWNLOAD_SOURCE build gagal dengan kesalahan`YAML_FILE_ERROR: This build image requires selecting at least one runtime version`.

Kemungkinan penyebabnya: Build Anda menggunakan image standar Amazon Linux 2 (AL2) versi 1.0 atau yang lebih baru, atau image standar Ubuntu versi 2.0 atau yang lebih baru, dan runtime tidak ditentukan dalam file buildspec.

Solusi yang disarankan: Jika Anda menggunakan image `aws/codebuild/standard:2.0` CodeBuild terkelola, Anda harus menentukan versi runtime di `runtime-versions` bagian file buildspec. Misalnya, Anda mungkin menggunakan file buildspec berikut untuk proyek yang menggunakan PHP:

```
version: 0.2

phases:
  install:
    runtime-versions:
        php: 7.3
  build:
    commands:
      - php --version
```

```
artifacts:
  files:
    -  README.md
```

> **Note**
>
> Jika Anda menentukan `runtime-versions` bagian dan menggunakan gambar selain Ubuntu Standard Image 2.0 atau yang lebih baru, atau Amazon Linux 2 (AL2) image standar 1.0 atau yang lebih baru, build akan mengeluarkan peringatan, "`Skipping install of runtimes. Runtime version selection is not supported by this build image`."

Untuk informasi selengkapnya, lihat Specify runtime versions in the buildspec file.

Kesalahan: “QUEUED: INSUFFICIENT_SUBNET” saat build dalam antrian build gagal

Masalah: Build dalam antrian build gagal dengan kesalahan yang mirip `QUEUED: INSUFFICIENT_SUBNET` dengan.

Kemungkinan penyebabnya: Blok CIDR IPv4 yang ditentukan untuk VPC Anda menggunakan alamat IP yang dipesan. Empat alamat IP pertama dan alamat IP terakhir di setiap blok CIDR subnet tidak tersedia untuk Anda gunakan dan tidak dapat ditetapkan ke sebuah instance. Sebagai contoh, dalam sebuah subnet dengan blok CIDR `10.0.0.0/24`, lima alamat IP berikut dicadangkan:

- `10.0.0.0:` Alamat jaringan.
- `10.0.0.1`: Dicadangkan oleh AWS untuk router VPC.
- `10.0.0.2`: Dicadangkan oleh AWS. Alamat IP server DNS selalu menjadi basis rentang jaringan VPC ditambah dua; Namun, kami juga mencadangkan basis setiap rentang subnet ditambah dua. Untuk VPC dengan beberapa blok CIDR, alamat IP server DNS terletak di CIDR utama. Lihat informasi yang lebih lengkap di Server DNS Amazon dalam Panduan Pengguna Amazon VPC.
- `10.0.0.3`: Dicadangkan oleh AWS untuk penggunaan masa depan.
- `10.0.0.255`: Alamat siaran jaringan. Kami tidak mendukung siaran dalam VPC. Alamat ini dipesan.

Solusi yang disarankan: Periksa apakah VPC Anda menggunakan alamat IP yang dipesan. Ganti alamat IP yang dicadangkan dengan alamat yang tidak dicadangkan. Untuk informasi selengkapnya, lihat Ukuran VPC dan subnet dalam Panduan Pengguna Amazon VPC.

Kesalahan: “Tidak dapat mengunduh cache: RequestError: Permintaan kirim gagal disebabkan oleh: x509: Gagal memuat akar sistem dan tidak ada akar yang disediakan”

Masalah: Saat Anda mencoba menjalankan proyek build, build gagal dengan kesalahan ini.

Kemungkinan penyebabnya: Anda mengonfigurasi caching sebagai bagian dari proyek build dan menggunakan image Docker lama yang menyertakan sertifikat root yang kedaluwarsa.

Solusi yang disarankan: Perbarui gambar Docker yang sedang digunakan dalam AWS CodeBuild proyek Anda. Untuk informasi selengkapnya, lihat Gambar Docker disediakan oleh CodeBuild.

Kesalahan: “Tidak dapat mengunduh sertifikat dari S3. AccessDenied”

Masalah: Saat Anda mencoba menjalankan proyek build, build gagal dengan kesalahan ini.

Kemungkinan penyebabnya:

- Anda telah memilih bucket S3 yang salah untuk sertifikat Anda.
- Anda telah memasukkan kunci objek yang salah untuk sertifikat Anda.

Solusi yang direkomendasikan:

- Edit proyek Anda. Untuk Bucket of certificate, pilih bucket S3 tempat sertifikat SSL Anda disimpan.
- Edit proyek Anda. Untuk kunci Objek sertifikat, masukkan nama kunci objek S3 Anda.

Kesalahan: “Tidak dapat menemukan kredensil”

Masalah: Saat Anda mencoba menjalankan AWS CLI, menggunakan AWS SDK, atau memanggil komponen serupa lainnya sebagai bagian dari build, Anda mendapatkan error build yang terkait

langsung dengan AWS CLI, AWS SDK, atau komponen. Misalnya, Anda mungkin mendapatkan kesalahan build sepertiUnable to locate credentials.

Kemungkinan penyebabnya:

- Versi AWS CLI, AWS SDK, atau komponen di lingkungan build tidak kompatibel dengannya. AWS CodeBuild
- Anda menjalankan wadah Docker dalam lingkungan build yang menggunakan Docker, dan penampung tidak memiliki akses ke AWS kredensyal secara default.

Solusi yang direkomendasikan:

- Pastikan lingkungan build Anda memiliki versi, AWS SDK AWS CLI, atau komponen berikut atau yang lebih tinggi.
  - AWS CLI: 1.10.47
  - AWS SDK for C++: 0.2.19
  - AWS SDK for Go: 1.2.5
  - AWS SDK for Java: 1.11.16
  - AWS SDK untuk JavaScript: 2.4.7
  - AWS SDK for PHP: 3.18.28
  - AWS SDK untuk Python (Boto3): 1.4.0
  - AWS SDK for Ruby: 2.3.22
  - Botocore: 1.4.37
  - CoreCLR: 3.2.6-beta
  - Node.js: 2.4.7
- Jika Anda perlu menjalankan container Docker di lingkungan build dan container memerlukan AWS kredensyal, Anda harus melewati kredensyal dari lingkungan build ke container. Dalam file buildspec Anda, sertakan `run` perintah Docker seperti berikut ini. Contoh ini menggunakan `aws s3 ls` perintah untuk membuat daftar bucket S3 Anda yang tersedia. `-e`Opsi melewati variabel lingkungan yang diperlukan untuk penampung Anda untuk mengakses AWS kredensyal.

```
docker run -e AWS_DEFAULT_REGION -e AWS_CONTAINER_CREDENTIALS_RELATIVE_URI your-image-tag aws s3 ls
```

- Jika Anda membuat image Docker dan build memerlukan AWS kredensyal (misalnya, untuk mengunduh file dari Amazon S3), Anda harus meneruskan kredensyal dari lingkungan build ke proses build Docker sebagai berikut.

  1. Di Dockerfile kode sumber Anda untuk gambar Docker, tentukan instruksi berikut. `ARG`

     ```
     ARG AWS_DEFAULT_REGION
     ARG AWS_CONTAINER_CREDENTIALS_RELATIVE_URI
     ```

  2. Dalam file buildspec Anda, sertakan `build` perintah Docker seperti berikut ini. `--build-arg`Opsi menetapkan variabel lingkungan yang diperlukan untuk proses pembuatan Docker Anda untuk mengakses AWS kredensyal.

     ```
     docker build --build-arg AWS_DEFAULT_REGION=$AWS_DEFAULT_REGION --build-arg
      AWS_CONTAINER_CREDENTIALS_RELATIVE_URI=$AWS_CONTAINER_CREDENTIALS_RELATIVE_URI -
     t your-image-tag .
     ```

RequestError kesalahan batas waktu saat berjalan CodeBuild di server proxy

Masalah: Anda menerima `RequestError` kesalahan yang mirip dengan salah satu dari berikut ini:

- `RequestError: send request failed caused by: Post https://logs.<your-region>.amazonaws.com/: dial tcp 52.46.158.105:443: i/o timeout`dari CloudWatch Log.
- `Error uploading artifacts: RequestError: send request failed caused by: Put https://your-bucket.s3.your-aws-region.amazonaws.com/*: dial tcp 52.219.96.208:443: connect: connection refused`dari Amazon S3.

Kemungkinan penyebabnya:

- `ssl-bump`tidak dikonfigurasi dengan benar.
- Kebijakan keamanan organisasi Anda tidak memungkinkan Anda untuk menggunakannya`ssl_bump`.
- File buildspec Anda tidak memiliki pengaturan proxy yang ditentukan menggunakan elemen. `proxy`

Solusi yang direkomendasikan:

- Pastikan `ssl-bump` dikonfigurasi dengan benar. Jika Anda menggunakan Squid untuk server proxy Anda, lihatKonfigurasikan Squid sebagai server proxy eksplisit.
- Ikuti langkah-langkah berikut untuk menggunakan titik akhir pribadi untuk Amazon S3 CloudWatch dan Log:

  1. Di tabel perutean subnet pribadi Anda, hapus aturan yang Anda tambahkan yang mengarahkan lalu lintas yang ditujukan untuk internet ke server proxy Anda. Untuk selengkapnya, lihat Membuat subnet di VPC Anda di Panduan Pengguna Amazon VPC.
  2. Buat endpoint Amazon S3 pribadi dan titik akhir CloudWatch Log dan kaitkan dengan subnet pribadi VPC Amazon Anda. Untuk selengkapnya, lihat layanan titik akhir VPC di Panduan Pengguna Amazon VPC.
  3. Konfirmasikan Aktifkan Nama DNS Pribadi di Amazon VPC Anda dipilih. Untuk informasi selengkapnya, lihat Membuat titik akhir antarmuka dalam Panduan Pengguna Amazon VPC.
- Jika Anda tidak menggunakan `ssl-bump` server proxy eksplisit, tambahkan konfigurasi proxy ke file buildspec Anda menggunakan elemen. `proxy` Untuk informasi selengkapnya, lihat Jalan CodeBuild kan di server proxy eksplisit dan Sintaks Buildspec.

```
version: 0.2
proxy:
  upload-artifacts: yes
  logs: yes
phases:
  build:
    commands:
```

Cangkang bourne (sh) harus ada dalam gambar build

Masalah: Anda menggunakan image build yang tidak disediakan oleh AWS CodeBuild, dan build Anda gagal dengan pesan `Build container found dead before completing the build` tersebut.

Kemungkinan penyebabnya: Bourne shell (`sh`) tidak disertakan dalam image build Anda. CodeBuild `sh`perlu menjalankan perintah dan skrip build.

Solusi yang disarankan: Jika `sh` tidak ada dalam image build Anda, pastikan untuk menyertakannya sebelum memulai build lagi yang menggunakan image Anda. (CodeBuild sudah termasuk `sh` dalam gambar build-nya.)

Peringatan: “Melewatkan pemasangan runtime. pemilihan versi runtime tidak didukung oleh image build ini” saat menjalankan build

Masalah: Saat Anda menjalankan build, log build berisi peringatan ini.

Kemungkinan penyebabnya: Build Anda tidak menggunakan image standar Amazon Linux 2 (AL2) versi 1.0 atau yang lebih baru, atau image standar Ubuntu versi 2.0 atau yang lebih baru, dan runtime ditentukan di `runtime-versions` bagian dalam file buildspec Anda.

Solusi yang disarankan: Pastikan file buildspec Anda tidak berisi bagian. `runtime-versions` `runtime-versions`Bagian ini hanya diperlukan jika Anda menggunakan gambar standar Amazon Linux 2 (AL2) atau yang lebih baru atau gambar standar Ubuntu versi 2.0 atau yang lebih baru.

Kesalahan: “Tidak dapat memverifikasi JobWorker identitas” saat membuka CodeBuild konsol

Masalah: Saat Anda membuka CodeBuild konsol, pesan kesalahan “Tidak dapat memverifikasi JobWorker identitas” akan ditampilkan.

Kemungkinan penyebabnya: Peran IAM yang digunakan untuk akses konsol memiliki tag dengan `jobId` kunci. Kunci tag ini dicadangkan untuk CodeBuild dan akan menyebabkan kesalahan ini jika ada.

Solusi yang disarankan: Ubah tag peran IAM kustom apa pun yang memiliki kunci `jobId` untuk memiliki kunci yang berbeda, seperti`jobIdentifier`.

Build gagal untuk memulai

Masalah: Saat memulai build, Anda menerima pesan galat Build failed to start.

Kemungkinan penyebabnya: Jumlah build bersamaan telah tercapai.

Solusi yang disarankan: Tunggu hingga build lain selesai, atau tingkatkan batas build bersamaan untuk proyek, dan mulai build lagi. Untuk informasi selengkapnya, lihat Konfigurasi proyek.

Mengakses GitHub metadata dalam build yang di-cache secara lokal

Masalah: Dalam beberapa kasus, direktori.git dalam build cache adalah file teks dan bukan direktori.

Kemungkinan penyebabnya: Saat caching sumber lokal diaktifkan untuk build, CodeBuild buat gitlink untuk direktori. `.git` Ini berarti bahwa `.git` direktori sebenarnya adalah file teks yang berisi path ke direktori.

Solusi yang disarankan: Dalam semua kasus, gunakan perintah berikut untuk mendapatkan direktori metadata Git. Perintah ini akan berfungsi apa pun format`.git`:

```
git rev-parse --git-dir
```

AccessDenied: Pemilik bucket untuk grup laporan tidak cocok dengan pemilik bucket S3...

Masalah: Saat mengunggah data pengujian ke bucket Amazon S3 CodeBuild , tidak dapat menulis data pengujian ke bucket.

Kemungkinan penyebabnya:

- Akun yang ditentukan untuk pemilik bucket grup laporan tidak cocok dengan pemilik bucket Amazon S3.
- Peran layanan tidak memiliki akses tulis ke bucket.

Solusi yang direkomendasikan:

- Ubah pemilik bucket grup laporan agar sesuai dengan pemilik bucket Amazon S3.
- Ubah peran layanan untuk mengizinkan akses tulis ke bucket Amazon S3.

Kesalahan: “Kredensyal Anda tidak memiliki satu atau lebih cakupan hak istimewa yang diperlukan” saat membuat proyek dengan CodeBuild AWS CodeConnections

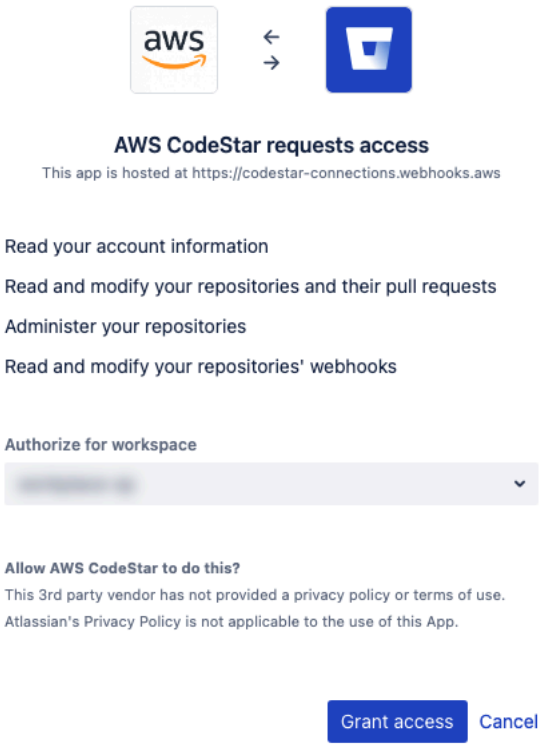
Masalah: Saat membuat CodeBuild proyek dengan AWS CodeConnections, Anda tidak memiliki izin untuk menginstal webhook Bitbucket.

Kemungkinan penyebabnya:

- Cakupan izin baru mungkin belum diterima di akun Bitbucket Anda.

Solusi yang direkomendasikan:

- Untuk menerima izin baru, Anda seharusnya telah menerima email dengan subjek berjudul Tindakan diperlukan - Lingkup untuk AWS CodeStar telah diubah dikirim oleh Bitbucket,. `notifications-noreply@bitbucket.org` Email berisi tautan untuk memberikan izin webhook ke instalasi aplikasi AWS CodeConnections Bitbucket Anda yang ada.
- Jika Anda tidak dapat menemukan email, Anda dapat memberikan izin dengan menavigasi kehttps://bitbucket.org/site/addons/reauthorize?account=*<workspace-name>*&addon_key=aws-codestar, atau `https://bitbucket.org/site/addons/reauthorize?addon_key=aws-codestar` dan memilih ruang kerja yang ingin Anda berikan izin webhook.



Kesalahan: “Maaf, tidak ada terminal sama sekali yang diminta - tidak bisa mendapatkan input” saat membangun dengan perintah instal Ubuntu

Masalah: Jika Anda menjalankan build istimewa kontainer GPU, Anda mungkin menginstal NVIDIA Container Toolkit mengikuti prosedur ini. Dalam rilis CodeBuild gambar terbaru, CodeBuild pra-instal dan konfigurasi docker `nvidia-container-toolkit` dengan gambar terbaru dan dikurasi. `amazonlinux` `ubuntu` Mengikuti prosedur ini akan menyebabkan build dengan perintah install Ubuntu gagal dengan kesalahan berikut:

```
Running command curl -fsSL https://nvidia.github.io/libnvidia-container/gpgkey | gpg --
dearmor --no-tty -o /usr/share/keyrings/nvidia-container-toolkit-keyring.gpg
gpg: Sorry, no terminal at all requested - can't get input
curl: (23) Failed writing body
```

Kemungkinan penyebabnya: Kunci gpg sudah ada di lokasi yang sama.

Solusi yang **`nvidia-container-toolkit`** disarankan: Sudah diinstal pada gambar. Jika Anda melihat kesalahan ini, Anda dapat melewati proses install dan restart docker di buildspec Anda.

Bitbucket build gagal dengan kesalahan otentikasi setelah menggunakan Secrets Manager OAuth

Masalah: Pembuatan menggunakan Bitbucket sebagai penyedia sumber gagal dengan kesalahan kredensyal sumber. Koneksi Bitbucket dikonfigurasi menggunakan OAuth dengan AWS Secrets Manager sebagai layanan penyimpanan.

Kemungkinan penyebabnya: Atlassian memberlakukan token penyegaran berputar sekali pakai untuk Bitbucket OAuth. Setiap kali Anda menggunakan token penyegaran, Bitbucket membatalkannya dan mengembalikan yang baru. Untuk informasi selengkapnya, lihat Bitbucket OAuth sekali pakai token refresh (CHANGE-3052) di situs web pengembang Atlassian.

Solusi yang disarankan: Tambahkan `secretsmanager:PutSecretValue` izin ke peran CodeBuild layanan Anda. Untuk informasi selengkapnya tentang izin IAM yang diperlukan, lihatTindakan yang diperlukan untuk Manager-stored kredensil Rahasia.

Kuota untuk AWS CodeBuild

Tabel berikut mencantumkan kuota saat ini di AWS CodeBuild. Kuota ini untuk setiap AWS Wilayah yang didukung untuk setiap AWS akun, kecuali ditentukan lain.

Kuota layanan

Berikut ini adalah kuota default untuk AWS CodeBuild layanan ini.

Nama	Default	Dapat disesuaikan	Deskripsi
Tag terkait per proyek	Setiap Wilayah yang didukung: 50	Tidak	Jumlah maksimum tag yang dapat Anda kaitkan dengan proyek build
Proyek build	Setiap Wilayah yang didukung: 5.000	Ya	Jumlah maksimum proyek pembangunan
Batas waktu Build dalam menit	Setiap Wilayah yang didukung: 2.160	Tidak	Batas waktu build maksimum dalam hitungan menit
Permintaan bersamaan untuk informasi tentang build	Setiap Wilayah yang didukung: 100	Tidak	Jumlah maksimum build yang dapat Anda minta informasi kapan saja menggunakan AWS CLI atau AWS SDK.
Permintaan bersamaan untuk informasi tentang proyek build	Setiap Wilayah yang didukung: 100	Tidak	Jumlah maksimum proyek build yang dapat Anda minta informasi kapan saja menggunakan AWS CLI atau SDK. AWS

Nama	Default	Dapat disesuaikan	Deskripsi
Menjalankan build secara bersamaan untuk lingkungan ARM Lambda/10GB	Setiap Wilayah yang didukung: 1	Ya	Jumlah maksimum build yang berjalan secara bersamaan untuk lingkungan ARM Lambda/10GB
Menjalankan build secara bersamaan untuk lingkungan ARM Lambda/1GB	Setiap Wilayah yang didukung: 1	Ya	Jumlah maksimum build yang berjalan secara bersamaan untuk lingkungan ARM Lambda/1GB
Menjalankan build secara bersamaan untuk lingkungan ARM Lambda/2GB	Setiap Wilayah yang didukung: 1	Ya	Jumlah maksimum build yang berjalan secara bersamaan untuk lingkungan ARM Lambda/2GB
Menjalankan build secara bersamaan untuk lingkungan ARM Lambda/4GB	Setiap Wilayah yang didukung: 1	Ya	Jumlah maksimum build yang berjalan secara bersamaan untuk lingkungan ARM Lambda/4GB
Menjalankan build secara bersamaan untuk lingkungan ARM Lambda/8GB	Setiap Wilayah yang didukung: 1	Ya	Jumlah maksimum build yang berjalan secara bersamaan untuk lingkungan ARM Lambda/8GB
Menjalankan build untuk lingkungan secara bersamaan ARM/2XLarge	Setiap Wilayah yang didukung: 1	Ya	Jumlah maksimum build yang berjalan secara bersamaan untuk lingkungan ARM/2XLarge

Nama	Default	Dapat disesuaikan	Deskripsi
Menjalankan build untuk lingkungan secara bersamaan ARM/Large	Setiap Wilayah yang didukung: 1	Ya	Jumlah maksimum build yang berjalan secara bersamaan untuk lingkungan ARM/Large
Menjalankan build untuk lingkungan secara bersamaan ARM/Medium	Setiap Wilayah yang didukung: 1	Ya	Jumlah maksimum build yang berjalan secara bersamaan untuk lingkungan ARM/Medium
Menjalankan build untuk lingkungan secara bersamaan ARM/Small	Setiap Wilayah yang didukung: 1	Ya	Jumlah maksimum build yang berjalan secara bersamaan untuk lingkungan ARM/Small
Menjalankan build untuk lingkungan secara bersamaan ARM/XLarge	Setiap Wilayah yang didukung: 1	Ya	Jumlah maksimum build yang berjalan secara bersamaan untuk lingkungan ARM/XLarge
Menjalankan build secara bersamaan untuk lingkungan Linux GPU Large	Setiap Wilayah yang didukung: 0	Ya	Jumlah maksimum build yang berjalan secara bersamaan untuk lingkungan Linux GPU/ Large
Menjalankan build secara bersamaan untuk lingkungan Linux GPU Small	Setiap Wilayah yang didukung: 0	Ya	Jumlah maksimum build yang berjalan secara bersamaan untuk lingkungan Linux GPU/ Small

Nama	Default	Dapat disesuaikan	Deskripsi
Menjalankan build secara bersamaan untuk lingkungan Linux Lambda/10GB	Setiap Wilayah yang didukung: 1	Ya	Jumlah maksimum build yang berjalan secara bersamaan untuk lingkungan Linux Lambda/10GB
Menjalankan build secara bersamaan untuk lingkungan Linux Lambda/1GB	Setiap Wilayah yang didukung: 1	Ya	Jumlah maksimum build yang berjalan secara bersamaan untuk lingkungan Linux Lambda/1GB
Menjalankan build secara bersamaan untuk lingkungan Linux Lambda/2GB	Setiap Wilayah yang didukung: 1	Ya	Jumlah maksimum build yang berjalan secara bersamaan untuk lingkungan Linux Lambda/2GB
Menjalankan build secara bersamaan untuk lingkungan Linux Lambda/4GB	Setiap Wilayah yang didukung: 1	Ya	Jumlah maksimum build yang berjalan secara bersamaan untuk lingkungan Linux Lambda/4GB
Menjalankan build secara bersamaan untuk lingkungan Linux Lambda/8GB	Setiap Wilayah yang didukung: 1	Ya	Jumlah maksimum build yang berjalan secara bersamaan untuk lingkungan Linux Lambda/8GB
Menjalankan build untuk lingkungan secara bersamaan Linux/2XLarge	Setiap Wilayah yang didukung: 0	Ya	Jumlah maksimum build yang berjalan secara bersamaan untuk lingkungan Linux/2XLarge

Nama	Default	Dapat disesuaikan	Deskripsi
Menjalankan build untuk lingkungan secara bersamaan Linux/Large	Setiap Wilayah yang didukung: 1	Ya	Jumlah maksimum build yang berjalan secara bersamaan untuk lingkungan Linux/Large
Menjalankan build untuk lingkungan secara bersamaan Linux/Medium	Setiap Wilayah yang didukung: 1	Ya	Jumlah maksimum build yang berjalan secara bersamaan untuk lingkungan Linux/Medium
Menjalankan build untuk lingkungan secara bersamaan Linux/Small	Setiap Wilayah yang didukung: 1	Ya	Jumlah maksimum build yang berjalan secara bersamaan untuk lingkungan Linux/Small
Menjalankan build untuk lingkungan secara bersamaan Linux/XLarge	Setiap Wilayah yang didukung: 1	Ya	Jumlah maksimum build yang berjalan secara bersamaan untuk lingkungan Linux/XLarge
Menjalankan build secara bersamaan untuk lingkungan Windows Server 2019/ Large	Setiap Wilayah yang didukung: 1	Ya	Jumlah maksimum build yang berjalan secara bersamaan untuk lingkungan Windows Server 2019/Large
Menjalankan build secara bersamaan untuk lingkungan Windows Server 2019/ Medium	Setiap Wilayah yang didukung: 1	Ya	Jumlah maksimum build yang berjalan secara bersamaan untuk lingkungan Windows Server 2019/Medium

Nama	Default	Dapat disesuaikan	Deskripsi
Menjalankan build secara bersamaan untuk lingkungan Windows Server 2022/2XLarge	Setiap Wilayah yang didukung: 1	Ya	Jumlah maksimum build yang berjalan secara bersamaan untuk lingkungan Windows Server 2022/2XLarge
Menjalankan build secara bersamaan untuk lingkungan Windows Server 2022/Large	Setiap Wilayah yang didukung: 1	Ya	Jumlah maksimum build yang berjalan secara bersamaan untuk lingkungan Windows Server 2022/Large
Menjalankan build secara bersamaan untuk lingkungan Windows Server 2022/Medium	Setiap Wilayah yang didukung: 1	Ya	Jumlah maksimum build yang berjalan secara bersamaan untuk lingkungan Windows Server 2022/Medium
Menjalankan build secara bersamaan untuk lingkungan Windows Server 2022/XLarge	Setiap Wilayah yang didukung: 1	Ya	Jumlah maksimum build yang berjalan secara bersamaan untuk lingkungan Windows Server 2022/XLarge
Menjalankan build untuk lingkungan secara bersamaan Windows/Large	Setiap Wilayah yang didukung: 1	Ya	Jumlah maksimum build yang berjalan secara bersamaan untuk lingkungan Windows/Large

Nama	Default	Dapat disesuaikan	Deskripsi
Menjalankan build untuk lingkungan secara bersamaan Windows/Medium	Setiap Wilayah yang didukung: 1	Ya	Jumlah maksimum build yang berjalan secara bersamaan untuk lingkungan Windows/Medium
Periode minimum untuk batas waktu build dalam menit	Setiap Wilayah yang didukung: 5	Tidak	Batas waktu build minimum dalam hitungan menit
Grup keamanan di bawah konfigurasi VPC	Setiap Wilayah yang didukung: 5	Tidak	Grup keamanan tersedia untuk konfigurasi VPC
Subnet di bawah konfigurasi VPC	Setiap Wilayah yang didukung: 16	Tidak	Subnet tersedia untuk konfigurasi VPC

> **Note**
>
> Metrik internal akan menentukan kuota default untuk build yang berjalan bersamaan.

Kuota untuk jumlah maksimum build yang berjalan bersamaan bervariasi, tergantung pada jenis komputasi. Untuk beberapa platform dan jenis komputasi, defaultnya adalah 20. Untuk meminta kuota build bersamaan yang lebih tinggi, atau jika Anda mendapatkan kesalahan “Tidak dapat memiliki lebih dari X build aktif untuk akun”, gunakan tautan di atas untuk membuat permintaan. Untuk informasi lebih lanjut tentang harga, lihat AWS CodeBuild harga.

Batas lainnya

Proyek build

Sumber daya	Default
Karakter yang diizinkan dalam deskripsi proyek build	Setiap
Karakter yang diizinkan dalam nama proyek build	Huruf `A-Z` dan`a-z`, angka-angka`0-9`, dan karakter khusus `-` dan `_`
Panjang nama proyek build	2 hingga 150 karakter, inklusif
Panjang maksimum deskripsi proyek build	255 karakter
Jumlah maksimum laporan yang dapat Anda tambahkan ke proyek	5
Jumlah menit yang dapat Anda tentukan dalam proyek build untuk batas waktu build semua build terkait	5 hingga 2160 (36 jam)

Bangunan

Sumber daya	Default
Waktu maksimum sejarah bangunan dipertaha nkan	1 tahun
Jumlah menit yang dapat Anda tentukan untuk batas waktu build dari satu build	5 hingga 2160 (36 jam)

Hitung armada

Sumber daya	Default
Jumlah armada komputasi bersamaan	10
Menjalankan instance secara bersamaan untuk ARM/Small armada lingkungan	1
Menjalankan instance secara bersamaan untuk ARM/Large armada lingkungan	1
Menjalankan instance secara bersamaan untuk Linux/Small armada lingkungan	1
Menjalankan instance secara bersamaan untuk Linux/Medium armada lingkungan	1
Menjalankan instance secara bersamaan untuk Linux/Large armada lingkungan	1
Menjalankan instance secara bersamaan untuk Linux/XLarge armada lingkungan	1
Menjalankan instance secara bersamaan untuk Linux/2XLarge armada lingkungan	0
Menjalankan instance secara bersamaan untuk armada lingkungan Linux GPU/Small	0
Menjalankan instance secara bersamaan untuk armada lingkungan Linux GPU/Large	0
Menjalankan instance secara bersamaan untuk armada lingkungan Windows Server 2019/Medium	1

Sumber daya	Default
Menjalankan instance secara bersamaan untuk armada lingkungan Windows Server 2019/Large	1
Menjalankan instance secara bersamaan untuk armada lingkungan Windows Server 2022/Medium	1
Menjalankan instance secara bersamaan untuk armada lingkungan Windows Server 2022/Large	1
Menjalankan instance secara bersamaan untuk armada lingkungan Mac ARM/Medium	1
Menjalankan instance secara bersamaan untuk armada lingkungan Mac ARM/Large	1

Laporan

Sumber daya	Default
Durasi maksimum laporan pengujian tersedia setelah dibuat	30 hari
Panjang maksimum pesan kasus uji	5.000 karakter
Panjang maksimum nama kasus uji	1.000 karakter
Jumlah maksimum grup laporan per AWS akun	5.000
Jumlah maksimum kasus uji per laporan	500

Tanda

Batas tag berlaku untuk tag pada proyek CodeBuild pembuatan dan sumber daya grup CodeBuild laporan.

Sumber daya	Default
Nama kunci tag sumber daya	Kombinasi huruf Unicode, angka, spasi, dan karakter yang diizinkan dengan UTF-8 panjang antara 1 dan 127 karakter. Karakter yang diizinkan adalah `+ - = . _ : / @` Nama kunci tag harus unik, dan setiap kunci hanya dapat memiliki satu nilai. Nama kunci tag tidak dapat: • dimulai dengan `aws:` • hanya terdiri dari spasi • diakhiri dengan spasi • berisi spasi atau setiap karakter berikut: `? ^ * [ \ ~ ! # $ % & * ( ) > < | " ' ` [ ] { } ;`
Nilai tag sumber daya	Kombinasi huruf Unicode, angka, spasi, dan karakter yang diizinkan dengan UTF-8 panjang antara 0 dan 255 karakter. Karakter yang diizinkan adalah `+ - = . _ : / @` Kunci hanya dapat memiliki satu nilai, tetapi banyak kunci dapat memiliki nilai yang sama. Nilai kunci tag tidak dapat berisi emoji atau salah satu karakter berikut: `? ^ * [ \ ~ ! # $ % & * ( ) > < | " ' ` [ ] { } ;`

AWS CodeBuild Riwayat dokumen Panduan Pengguna

Tabel berikut menjelaskan perubahan penting pada dokumentasi sejak rilis terakhir AWS CodeBuild. Untuk notifikasi tentang pembaruan dokumentasi ini, Anda dapat berlangganan ke umpan RSS.

- Versi API terbaru: 2016-10-06

Perubahan	Deskripsi	Tanggal
Konten yang diperbarui: Runtime yang tersedia untuk gambar CodeBuild standar	Menambahkan runtime Node.js 26 dan Rust 1.97 untuk gambar standar x86_64.	September 22, 2026
Konten yang diperbarui: Runtime yang tersedia untuk gambar CodeBuild standar	Menambahkan gambar Amazon Linux 2023 AArch64 standar:4.0 ke runtime yang didukung.	September 22, 2026
Konten yang diperbarui: sintaks ekspresi reguler filter webhook	Mengklarifikasi bahwa pola filter webhook menggunakan sintaks ekspresi reguler RE2 dan konstruksi mana yang tidak didukung.	Agustus 3, 2026
Konten yang diperbarui: Gambar komputasi Lambda	Menambahkan gambar komputasi Lambda untuk.NET 10,,Go 1.25, Corretto 25Node.js 24, dan. Python 3.14	Juli 29, 2026
Konten yang diperbarui: Kata sandi aplikasi Bitbucket, token API, atau token akses	Tambahkan dukungan untuk token API Bitbucket, yang menggantikan kata sandi aplikasi yang tidak digunakan lagi.	Juli 19, 2026

Pemetaan peran GitHub Perusahaan khusus dan perbaikan nama peran MAINTAIN	Menambahkan dokumentasi tentang bagaimana peran repositori kustom GitHub perusahaan dipetakan ke peran standar berdasarkan hierarki izin. Mengoreksi nama peran tingkat pemeliharaan menjadi nilai dan `GITHUB_MAINTAIN` contoh default.	April 13, 2026
Konten yang diperbarui: Runtime yang tersedia untuk gambar CodeBuild standar	Memperbarui versi runtime yang tersedia dan menambahkan gambar baru.	7 April 2026
Konten yang diperbarui: Gambar komputasi EC2	Menambahkan gambar komputasi Amazon Linux 2023 x86_64 standar:6.0 dan Ubuntu 24.04 standar:8.0.	7 April 2026
Konten yang diperbarui: Perilaku pemilihan versi runtime	Menambahkan panduan bahwa versi runtime default dapat berubah ketika versi default sebelumnya mencapai akhir masa pakai (EOL).	Maret 25, 2026
Konten yang diperbarui: Runtime yang tersedia untuk gambar CodeBuild standar	Memperbarui versi runtime yang tersedia dan menambahkan gambar baru.	Maret 25, 2026

Konten yang diperbarui: kebijakan AWS yang dikelola (telah ditentukan sebelumnya) untuk AWS CodeBuild	AWSCodeBuildDeveloperAccess Kebijakan AWSCodeBuildAdminAccess dan telah diperbarui untuk menyertakan `ssmmessages:OpenDataChannel` izin yang diperlukan untuk debugging build interaktif Manajer Sesi.	Desember 1, 2025
Persetujuan komentar permintaan tarik	CodeBuild mendukung kebijakan pembuatan permintaan tarik yang memberikan kontrol tambahan atas build yang dipicu oleh permintaan tarik.	Agustus 7, 2025
Konten yang diperbarui: kebijakan AWS yang dikelola (telah ditentukan sebelumnya) untuk AWS CodeBuild	Kebijakan AWSCodeBuildAdminAccess AWSCodeBuildDeveloperAccess,, dan AWSCodeBuildReadOnlyAccess telah diperbarui. Sumber daya asli `arn:aws:codeconnections:*:*:connection/*` telah diperbarui ke`arn:aws:codeconnections:*:*:*` .	Juni 10, 2025
Referensi baru untuk kunci CodeBuild kondisi	Menambahkan halaman referensi baru dengan contoh untuk menggunakan kunci CodeBuild kondisi. Lihat kunci AWS CodeBuild kondisi.	Mei 15, 2025

Konten baru: Contoh server pembuatan gambar Docker untuk CodeBuild	CodeBuild sekarang mendukung pemuatan build Docker Anda ke server pembuatan gambar terkelola.	Mei 15, 2025
Jenis komputasi baru: `CUSTOM_INSTANCE_TYPE`	CodeBuild sekarang memungkinkan Anda untuk membuat armada kapasitas cadangan dengan jenis instance tertentu dengan menggunakan`CUSTOM_INSTANCE_TYPE` .	April 23, 2025
Dukungan baru untuk CodeBuild kotak pasir	Menambahkan informasi tentang menggunakan kotak CodeBuild pasir baru. Lihat Men -debug build dengan CodeBuild kotak pasir.	7 April 2025
Jenis lingkungan Windows baru	CodeBuild sekarang mendukung jenis lingkungan Windows XL dan 2XL. Untuk informasi selengkapnya, lihat Mem bangun tipe komputasi lingkungan.	Maret 31, 2025
Caching Amazon S3 yang diperbarui	CodeBuild sekarang mendukung perilaku caching baru untuk caching Amazon S3.	Maret 28, 2025

Konten baru: GitHub Opsi konfigurasi Actions runner	CodeBuild sekarang mendukung `CODEBUILD_CONFIG_GITHUB_ACTIONS_ENTERPRISE_REGISTRATION_NAME` pendaftaran di tingkat perusahaan.	Maret 11, 2025
Konten baru: Tambahkan jenis filter webhook baru	Tambahkan dukungan untuk jenis filter webhook baru (`ORGANIZATION_NAME`).	Maret 11, 2025
Konten baru: Tutorial untuk penandatanganan kode Apple dengan Fastlane dengan penyimpanan sertifikat S3	Tambahkan tutorial baru untuk penandatanganan kode Apple dengan Fastlane dalam CodeBuild menggunakan S3 untuk penyimpanan sertifikat	Februari 5, 2025
Konten baru: Tutorial untuk penandatanganan kode Apple dengan Fastlane dengan penyimpanan GitHub sertifikat	Tambahkan tutorial baru untuk penandatanganan kode Apple dengan Fastlane dalam CodeBuild penggunaan GitHub untuk penyimpanan sertifikat	Februari 5, 2025
Konten baru: Buildkite runner	Tambahkan konten baru untuk pelari Buildkite	Januari 31, 2025
Konten baru: Buildkite manual webhooks	Tambahkan dukungan untuk webhook manual Buildkite.	Januari 31, 2025
Konten baru: Referensi buildspec build batch	Tambahkan dukungan untuk build batch di armada kapasitas cadangan dan lingkungan Lambda.	Januari 8, 2025

Konten baru: Jalankan tes paralel dalam build batch	Tambahkan konten baru untuk pengujian paralel dalam build batch.	Januari 2, 2025
Konten baru: Coba ulang dibuat secara otomatis	CodeBuild sekarang mendukung coba ulang otomatis untuk build webhook.	Desember 18, 2024
Konten baru: Konfigurasikan kredensi registri pribadi untuk pelari yang di-host sendiri	Tambahkan dukungan untuk mengatur kredenSIAL registri saat menggunakan gambar khusus dari registri non-priba di.	Desember 13, 2024
Konten baru: GitHub Opsi konfigurasi Actions runner	CodeBuild GitHub Actions self-host runner sekarang memungkinkan Anda mendaftarkan pelari Anda di tingkat organisasi dan mengonfigurasi ID grup pelari tertentu.	Desember 12, 2024
Konten baru: Tambahkan atribut on-failure RETRY	CodeBuild sekarang memungkinkan Anda untuk mengonfigurasi atribut on-failure ke RETRY dalam buildspec Anda.	Desember 12, 2024
Konten baru: GitLab manual webhook	Tambahkan dukungan untuk web GitLab hook manual.	Desember 11, 2024
Konten yang diperbarui: Alias yang diperbarui	Perbarui alias untuk gambar runtime Linux-based standar.	November 22, 2024
Konten yang diperbarui: Penggantian label didukung dengan pelari CodeBuild-hosted GitLab	Tambahkan dukungan untuk penggantian label gambar khusus untuk GitLab pelari.	November 22, 2024

Konten yang diperbarui: Penggantian label didukung dengan Ac CodeBuild-hosted GitHub tions runner	Tambahkan dukungan untuk penggantian label gambar khusus untuk pelari T GitHub indakan.	November 22, 2024
Konten yang diperbarui: kebijakan AWS yang dikelola (telah ditentukan sebelumnya) untuk AWS CodeBuild	Kebijakan AWSCodeBuildAdminAccess AWSCodeBuildDeveloperAccess,, dan AWSCodeBuildReadOnlyAccess telah diperbarui. Sumber daya asli `arn:aws:codebuild:*` telah diperbarui kearn:aws:codebuild:*:*:project/* .	November 15, 2024
Konten yang diperbarui: Kapasitas cadangan	Armada kapasitas cadangan sekarang mendukung build non-kontainer: ARM EC2, Linux EC2, dan Windows EC2.	November 12, 2024
Konten yang diperbarui: Kapasitas cadangan	Armada kapasitas cadangan sekarang mendukung komputasi berbasis atribut.	6 November 2024
Konten baru: Coba ulang dibuat secara otomatis	CodeBuild sekarang memungkinkan Anda untuk mengaktifkan coba ulang otomatis untuk build Anda.	Oktober 25, 2024
Konten baru: Jalan CodeBuildkan di server proxy terkelola untuk armada kapasitas cadangan	Tambahkan dukungan konfigurasi proxy untuk armada kapasitas cadangan.	Oktober 15, 2024
Konten baru: Self-managed GitLab pelari	Tambahkan konten baru untuk pelari yang dikelola sendiri GitLab	September 17, 2024

Konten baru: GitLab grup webhook	Tambahkan dukungan untuk web GitLab hook grup.	September 17, 2024
Konten baru: Jalankan perintah buildspec dalam fase INSTALL, PRE_BUILD, dan POST_BUILD	Tambahkan dukungan untuk `-with-buildspec` .	Agustus 20, 2024
Konten yang diperbarui: Kapasitas cadangan	Armada kapasitas cadangan sekarang mendukung macOS.	Agustus 19, 2024
Konten baru: GitHub Koneksi aplikasi	Tambahkan dukungan untuk koneksi GitHub Aplikasi.	Agustus 14, 2024
Konten baru: Koneksi Aplikasi Bitbucket	Tambahkan dukungan untuk koneksi Aplikasi Bitbucket.	Agustus 14, 2024
Konten baru: Beberapa token akses di CodeBuild	Tambahkan dukungan untuk sumber token akses ke penyedia pihak ketiga dari rahasia di dalam AWS Secrets Manager atau melalui AWS CodeConnections koneksi.	Agustus 14, 2024
Konten yang diperbarui: Kapasitas cadangan	Armada kapasitas cadangan sekarang mendukung tipe komputasi ARM Medium, ARM XLarge, dan ARM 2xLarge.	Agustus 5, 2024
Konten yang diperbarui: Kapasitas cadangan	CodeBuild sekarang mendukung konektivitas VPC untuk armada kapasitas cadangan di Windows.	Agustus 1, 2024

Jenis komputasi ARM baru	CodeBuild sekarang mendukung tipe komputasi ARM Medium, ARM XLarge, dan ARM 2xLarge. Untuk informasi selengkapnya, lihat Mem bangun tipe komputasi lingkungan.	Juli 10, 2024
Konten yang diperbarui: tanda tangan SHA	Perbarui tanda tangan Secure Hash Algorithm (SHA) untuk x86_64 dan ARM.	Juni 19, 2024
Konten baru: webhook GitHub global dan organisasi	Tambahkan dukungan untuk web GitHub hook global dan organisasi.	Juni 17, 2024
Konten baru: Tambahkan jenis filter webhook baru	Tambahkan dukungan untuk jenis filter webhook baru (`REPOSITORY_NAME`).	Juni 17, 2024
Ruang disk yang diperbarui	Jenis `ARM Small` dan `ARM Large` komputasi sekarang telah meningkatkan ruang disk.	Juni 4, 2024
Konten baru: GitHub manual webhook	Tambahkan dukungan untuk web GitHub hook manual.	Mei 23, 2024
Konten yang diperbarui: Kapasitas cadangan	CodeBuild sekarang mendukung konektivitas VPC untuk armada kapasitas cadangan di Amazon Linux.	Mei 15, 2024
Konten yang diperbarui: Gambar komputasi Lambda	Tambahkan dukungan Lambda untuk.NET 8 (`al-lambda/aarch64/dotnet8 danal-lambda/x86_64/dotnet8`)	8 Mei 2024

Kuota yang diperbarui: Bangun waktu tunggu	Perbarui kuota batas waktu pembuatan maksimum menjadi 2160 menit (36 jam).	1 Mei 2024
Konten yang diperbarui: kebijakan AWS yang dikelola (telah ditentukan sebelumnya) untuk AWS CodeBuild	Kebijakan AWSCodeBuildAdminAccess, AWSCodeBuildDeveloperAccess, dan AWSCodeBuildReadOnlyAccess telah diperbarui untuk mencerminkan AWS CodeConnections rebranding.	April 30, 2024
Konten baru: Kata sandi aplikasi Bitbucket, token API, atau token akses	Tambahkan dukungan untuk token akses Bitbucket.	April 11, 2024
Konten baru: Auto-discover laporan di CodeBuild	CodeBuild sekarang mendukung pencarian otomatis laporan.	April 4, 2024
Konten baru: Self-hosted GitHub Actions runner	Tambahkan konten baru untuk pelari Actions yang di-host sendiri GitHub	April 2, 2024
Konten baru: GitLab koneksi	Tambahkan dukungan untuk GitLab dan koneksi GitHub yang dikelola sendiri.	Maret 25, 2024
Konten baru: Tambahkan acara webhook baru dan jenis filter	Tambahkan dukungan untuk acara webhook baru (`RELEASED`dan`PRERELEASED`) dan jenis filter (`TAG_NAME`dan`RELEASE_NAME`).	Maret 15, 2024

Konten baru: Tambahkan acara webhook baru: `PULL_REQUEST_CLOSED`	Tambahkan dukungan untuk acara webhook baru:`PULL_REQUEST_CLOSED` .	Februari 20, 2024
Konten yang diperbarui: Gambar Docker disediakan oleh CodeBuild	Tambahkan dukungan untuk Windows Server Core 2019 (`windows-base:2019-3.0`)	Februari 7, 2024
Konten yang diperbarui: Gambar Docker disediakan oleh CodeBuild	Tambahkan dukungan untuk runtime baru untuk Amazon Linux 2023 () `al2/aarch64/standard/3.0`	Januari 29, 2024
Konten baru: Kapasitas cadangan	CodeBuild sekarang mendukung armada kapasitas cadangan di CodeBuild.	Januari 18, 2024
Jenis komputasi baru	CodeBuild sekarang mendukung tipe komputasi Linux XLarge. Untuk informasi selengkapnya, lihat Mem bangun tipe komputasi lingkungan.	8 Januari 2024
Konten yang diperbarui: Gambar Docker disediakan oleh CodeBuild	Tambahkan dukungan untuk runtime baru untuk Amazon Linux 2 (`al2/standard/5.0`) dan Ubuntu (`ubuntu/standard/7.0`)	14 Desember 2023
Konten yang diperbarui: Gambar Docker disediakan oleh CodeBuild	Tambahkan dukungan untuk gambar komputasi Lambda baru	8 Desember 2023
Konten baru: AWS Lambda komputasi	Tambahkan konten baru untuk AWS Lambda komputasi	6 November 2023

Konten yang diperbarui: Gambar Docker disediakan oleh CodeBuild	Tambahkan dukungan untuk Amazon Linux 2 (`al2/standard/5.0`)	Mei 17, 2023
Perubahan kebijakan terkelola untuk CodeBuild	Rincian tentang pembaruan AWS kebijakan terkelola untuk sekarang CodeBuild tersedia. Untuk informasi selengkapnya, lihat CodeBuild pembaruan kebijakan AWS terkelola.	Mei 16, 2023
Konten yang diperbarui: Gambar Docker disediakan oleh CodeBuild	Hapus dukungan untuk Amazon Linux 2 (`al2/standard/3.0`) dan tambahkan dukungan untuk Amazon Linux 2 (`al2/standard/corretto8`) dan Amazon Linux 2 (`al2/standard/corretto11`)	9 Mei 2023
Konten yang diperbarui: Gambar Docker disediakan oleh CodeBuild	Tambahkan dukungan untuk Ubuntu 22.04 () `ubuntu/standard/7.0`	13 April 2023
Konten yang diperbarui: Gambar Docker disediakan oleh CodeBuild	Hapus dukungan untuk Ubuntu 18.04 (`ubuntu/standard/4.0`) dan Amazon Linux 2 (`al2/aarch64/standard/1.0`)	31 Maret 2023

Konten yang diperbarui: Hapus batasan VPC	Menghapus batasan berikut: Jika Anda mengonfigurasi CodeBuild untuk bekerja dengan VPC, caching lokal tidak didukung. Mulai 02/28 /22, build VPC Anda akan memakan waktu lebih lama karena instans Amazon EC2 baru akan digunakan untuk setiap build.	1 Maret 2023
Konten yang diperbarui: Gambar Docker disediakan oleh CodeBuild	Hapus dukungan untuk Ubuntu 18.04 (`ubuntu/standard/3.0`) dan Amazon Linux 2 (`al2/standard/2.0`)	30 Juni 2022
Contoh Amazon ECR: Batasi akses gambar	Saat CodeBuild kredenSIAL digunakan untuk menarik gambar Amazon ECR, Anda dapat membatasi akses gambar ke CodeBuild proyek tertentu. Untuk informasi selengkapnya, lihat contoh Amazon ECR.	10 Maret 2022
Menambahkan dukungan wilayah	Jenis `ARM_CONTAINER` komputasi sekarang didukung di wilayah tambahan berikut: Asia Pasifik (Seoul), Kanada (Tengah), Eropa (London), dan Eropa (Paris). Untuk informasi selengkapnya, lihat Mem bangun tipe komputasi lingkungan.	10 Maret 2022

Batasan VPC baru	Jika Anda mengkonfigurasi CodeBuild untuk bekerja dengan VPC, caching lokal tidak didukung. Mulai 02/28 /22, build VPC Anda akan memakan waktu lebih lama karena instans Amazon EC2 baru akan digunakan untuk setiap build.	25 Februari 2022
Mode laporan batch	CodeBuild sekarang memungkinkan Anda untuk memilih bagaimana status build batch dikirim ke penyedia sumber untuk sebuah proyek. Untuk informasi selengkapnya, lihat Mode laporan batch.	4 Oktober 2021
Jenis komputasi baru	CodeBuild sekarang mendukung tipe komputasi ARM kecil. Untuk informasi selengkapnya, lihat Mem bangun tipe komputasi lingkungan.	13 September 2021
Proyek pembangunan publik	CodeBuild sekarang memungkinkan Anda membuat hasil build untuk proyek build Anda tersedia untuk umum tanpa memerluka n akses ke AWS akun. Untuk informasi selengkapnya, lihat Proyek pembangunan publik.	Agustus 11, 2021

Debugging sesi untuk pembuatan batch	CodeBuild sekarang mendukung sesi debugging untuk build batch. Untuk informasi selengkapnya, lihat build-graph dan build-list.	3 Maret 2021
Batas pembuatan bersamaan tingkat proyek	CodeBuild sekarang memungkinkan Anda untuk membatasi jumlah build bersamaan untuk proyek build. Untuk informasi selengkapnya, lihat Konfigurasi proyek dan bersamaan BuildLimit.	16 Februari 2021
Properti buildspec baru: s3-prefix	CodeBuild sekarang menyediakan properti `s3-prefix` buildspec untuk artefak yang memungkinkan Anda menentukan awalan jalur untuk artefak yang diunggah ke Amazon S3. Untuk informasi selengkapnya, lihat prefik s S3.	9 Februari 2021
Properti buildspec baru: on-failure	CodeBuild sekarang menyediakan properti `on-failure` buildspec untuk fase build yang memungkinkan Anda menentukan apa yang terjadi ketika fase build gagal. Untuk informasi selengkapnya, lihat on-failure.	9 Februari 2021

Properti buildspec baru: jalur yang dikecualikan	CodeBuild sekarang menyediakan properti `exclude-paths` buildspec untuk artefak yang memungkinkan Anda mengecualikan jalur dari artefak build Anda. Untuk informasi selengkapnya, lihat peng ecualian-jalur.	9 Februari 2021
Properti buildspec baru: enable-symlinks	CodeBuild sekarang menyediakan properti `enable-symlinks` buildspec untuk artefak yang memungkinkan Anda melestarikan tautan simbolis dalam artefak ZIP. Untuk informasi selengkapnya, lihat enab le-symlink.	9 Februari 2021
Peningkatan nama artefak buildspec	CodeBuild sekarang memungkinkan `artifacts/name` properti untuk berisi informasi jalur. Untuk informasi selengkapnya, lihat nama.	9 Februari 2021
Pelaporan cakupan kode	CodeBuild sekarang menyediakan laporan cakupan kode. Untuk informasi selengkapnya, lihat Laporan cakupan kode.	30 Juli 2020
Pembuatan batch	CodeBuild sekarang mendukung menjalankan build proyek secara bersamaan dan terkoordinasi. Untuk informasi selengkapnya, lihat Batch build in CodeBuild.	30 Juli 2020

Gambar Windows Server 2019	CodeBuild sekarang menyediakan gambar build Windows Server Core 2019. Untuk informasi selengkapnya, lihat Gambar Docker yang disediakan oleh CodeBuild.	20 Juli 2020
Manajer Sesi	CodeBuild sekarang memungkinkan Anda untuk menjeda build yang sedang berjalan dan kemudian menggunakan AWS Systems Manager Session Manager untuk terhubung ke wadah build dan melihat status wadah. Untuk informasi selengkapnya, lihat Pengelola Sesi .	20 Juli 2020
Topik yang diperbarui	CodeBuild sekarang mendukung menentukan shell untuk digunakan di lingkungan build mereka di file buildspec. Untuk informasi selengkapnya, lihat Referensi spesifikasi build.	25 Juni 2020
Uji pelaporan dengan kerangka pengujian	Menambahkan beberapa topik yang menjelaskan cara menghasilkan laporan pengujian CodeBuild dengan beberapa kerangka pengujian. Untuk informasi selengkapnya, lihat Menguji pelaporan dengan kerangka kerja pengujian.	29 Mei 2020

Topik yang diperbarui	CodeBuild sekarang mendukung penambahan tag ke grup laporan. Untuk informasi selengkapnya, lihat ReportGroup.	21 Mei 2020
Dukungan untuk pelaporan pengujian	CodeBuild dukungan untuk pelaporan pengujian sekarang tersedia secara umum.	21 Mei 2020
Topik yang diperbarui	CodeBuild sekarang mendukung pembuatan filter webhook buat untuk Github dan Bitbucket yang memicu build hanya ketika pesan komit kepala cocok dengan ekspresi yang ditentukan. Untuk informasi selengkap nya, lihat GitHub pull request dan webhook filter sample dan Bitbucket pull request dan webhook filter sample.	6 Mei 2020
Topik baru	CodeBuild sekarang mendukung berbagi proyek build dan laporan sumber daya grup. Untuk informasi selengkapnya, lihat Bekerja dengan proyek bersama dan Bekerja dengan grup laporan bersama.	13 Desember 2019

Topik baru dan diperbarui	CodeBuild sekarang mendukung pelaporan pengujian selama menjalankan proyek build. Untuk informasi selengkapnya, lihat Bekerja dengan pelaporan pengujian, Membuat laporan pengujian, dan Membuat laporan pengujian menggunakan AWS CLI sampel.	25 November 2019
Topik yang diperbarui	CodeBuild sekarang mendukung jenis lingkungan GPU dan Arm Linux, dan tipe `2xlarge` komputasi. Untuk informasi selengkapnya, lihat Membangun tipe komputasi lingkungan.	19 November 2019
Topik yang diperbarui	CodeBuild sekarang mendukung nomor build pada semua build, mengekspor variabel lingkungan, dan AWS Secrets Manager integrasi. Untuk informasi selengkapnya, lihat Variabel yang diekspor dan Manajer Rahasia dalam sintaks Buildspec.	6 November 2019

Topik baru	CodeBuild sekarang mendukung aturan notifikasi. Anda dapat menggunakan aturan notifikasi untuk memberi tahu pengguna tentang perubahan penting dalam proyek build. Untuk informasi selengkapnya, lihat Membuat aturan notifikasi.	5 November 2019
Topik yang diperbarui	CodeBuild sekarang mendukung runtime Android versi 29 dan Go versi 1.13. Untuk informasi selengkapnya, lihat gambar Docker yang disediakan oleh CodeBuild dan sintaks Buildspec.	10 September 2019
Topik yang diperbarui	Saat Anda membuat proyek, Anda sekarang dapat memilih gambar yang dikelola Amazon Linux 2 (AL2). Untuk informasi selengkapnya, lihat gambar Docker yang disediakan oleh CodeBuild dan versi Runtime dalam contoh file buildspec untuk. CodeBuild	16 Agustus 2019
Topik yang diperbarui	Saat Anda membuat proyek, Anda sekarang dapat memilih untuk menonaktifkan enkripsi log S3 dan, jika Anda menggunakan repositori Git-based sumber, sertakan submodul Git. Untuk informasi selengkapnya, lihat Membuat proyek build di CodeBuild.	8 Maret 2019

Topik baru	CodeBuild sekarang mendukung caching lokal. Anda dapat menentukan caching lokal dalam satu atau lebih dari empat mode saat Anda membuat build. Untuk informasi selengkapnya, lihat Membangun caching di CodeBuild.	21 Februari 2019
Topik baru	CodeBuild sekarang mendukung grup filter webhook untuk menentukan peristiwa yang memicu build. Untuk informasi selengkap nya, lihat Mem filter peristiwa GitHub webhook dan Mem filter peristiwa webhook Bitbucket.	8 Februari 2019
Topik baru	Panduan CodeBuild Pengguna sekarang menunjukkan cara menggunakan CodeBuild dengan server proxy. Untuk informasi selengkapnya, lihat Menggunakan CodeBuild dengan server proxy.	4 Februari 2019

Topik yang diperbarui	CodeBuild sekarang mendukung penggunaan gambar Amazon ECR yang ada di AWS akun lain. Beberapa topik telah diperbarui untuk mencerminkan perubahan ini, termasuk contoh Amazon ECR untuk CodeBuild, Buat proyek build, dan Buat peran CodeBuild layanan.	24 Januari 2019
Dukungan untuk registri Docker pribadi	CodeBuild sekarang mendukung penggunaan gambar Docker yang disimpan dalam registri pribadi sebagai lingkungan runtime Anda. Untuk informasi selengkapnya, lihat Registri pribadi dengan AWS Secrets Manager sampel.	24 Januari 2019
Topik yang diperbarui	CodeBuild sekarang mendukung penggunaan token akses untuk terhubung ke GitHub (dengan token akses pribadi) dan Bitbucket (dengan kata sandi aplikasi) repositori. Untuk informasi selengkapnya, lihat Membuat proyek build (konsol) dan Menggunakan token akses dengan penyedia sumber Anda.	6 Desember 2018

Topik yang diperbarui	CodeBuild sekarang mendukung metrik build baru yang mengukur durasi setiap fase dalam build. Untuk informasi selengkapnya, lihat CodeBuild CloudWatch metrik.	15 November 2018
Topik kebijakan titik akhir VPC	Titik akhir Amazon VPC untuk CodeBuild saat ini mendukung kebijakan. Untuk informasi selengkapnya, lihat Membuat kebijakan titik akhir VPC untuk CodeBuild.	9 November 2018
Konten yang diperbarui	Topik telah diperbarui untuk mencerminkan pengalaman konsol baru.	30 Oktober 2018
Sampel Amazon EFS	CodeBuild dapat memasang sistem file Amazon EFS selama pembuatan menggunakan perintah dalam file buildspec proyek. Untuk informasi selengkapnya, lihat contoh Amazon EFS untuk CodeBuild.	26 Oktober 2018
Webhook Bitbucket	CodeBuild sekarang mendukung webhook saat Anda menggunakan Bitbucket untuk repositori Anda. Untuk informasi selengkapnya, lihat contoh permintaan tarik Bitbucket untuk CodeBuild.	2 Oktober 2018

Log S3	CodeBuild sekarang mendukung build log di bucket S3. Sebelumnya, Anda hanya bisa membuat log menggunakan CloudWatch Log. Untuk informasi selengkapnya, lihat Membuat proyek.	17 September 2018
Beberapa sumber input dan beberapa artefak keluaran	CodeBuild sekarang mendukung proyek yang menggunakan lebih dari satu sumber input dan menerbitkan lebih dari satu set artefak. Untuk informasi selengkapnya, lihat Beberapa sumber masukan dan sampel artefak masukan dan CodePipeline integrasi dengan CodeBuild dan beberapa sumber masukan dan sampel artefak keluaran.	30 Agustus 2018
Contoh versi semantik	Panduan CodeBuild Pengguna sekarang memiliki sampel berbasis kasus penggunaan yang menunjukkan cara menggunakan versi semantik untuk membuat nama artefak pada waktu pembuatan. Untuk informasi selengkapnya, lihat Menggunakan versi semantik untuk memberi nama sampel artefak build.	14 Agustus 2018

Contoh situs web statis baru	Panduan CodeBuild Pengguna sekarang memiliki sampel berbasis kasus penggunaan yang menunjukkan cara meng-host output build di bucket S3. Sampel mengambil keuntungan dari dukungan terbaru dari artefak build yang tidak terenkripsi. Untuk informasi selengkapnya, lihat Membuat situs web statis dengan keluaran build yang dihosting di bucket S3.	14 Agustus 2018
Dukungan untuk mengganti nama artefak dengan versi semantik	Anda sekarang dapat menggunakan versi semantik untuk menentukan format yang digunakan untuk CodeBuild memberi nama artefak build. Ini berguna karena artefak build dengan nama hardcode menggantikan artefak build sebelumnya yang menggunakan nama hardcode yang sama. Misalnya, jika build dipicu beberapa kali sehari, Anda sekarang dapat menambahkan stempel waktu ke nama artefaknya. Setiap nama artefak build unik dan tidak mengganti artefak build sebelumnya.	7 Agustus 2018

Dukungan artefak build yang tidak terenkripsi	CodeBuild sekarang mendukung build dengan artefak build yang tidak terenkripsi. Untuk informasi selengkapnya, lihat Membuat proyek build (konsol).	26 Juli 2018
Dukungan untuk CloudWatch metrik dan alarm Amazon	CodeBuild sekarang menyediakan integrasi dengan CloudWatch metrik dan alarm. Anda dapat menggunakan CloudWatch konsol CodeBuild atau untuk memantau build di tingkat proyek dan akun. Untuk informasi selengkapnya, lihat Mem antau build.	19 Juli 2018
Dukungan untuk melaporkan status build	CodeBuild sekarang dapat melaporkan status awal dan penyelesaian build ke penyedia sumber Anda. Untuk informasi selengkapnya, lihat Membuat proyek build di CodeBuild.	10 Juli 2018
Variabel lingkungan ditambahkan ke CodeBuild dokumentasi	Variabel Lingkungan di halaman lingkungan build diperbarui dengan variabel lingkungan CODEBUILD_BUILD_ID, CODEBUILD_LOG_PATH, dan CODEBUILD_START_TIME.	9 Juli 2018

Dukungan untuk `finally` blok dalam file buildspec	CodeBuild Dokumentasi diperbarui dengan detail tentang `finally` blok opsional dalam file buildspec. Perintah di blok terakhir selalu berjalan setelah perintah di blok perintah yang sesuai. Untuk informasi selengkapnya, lihat sintaks Buildspec.	20 Juni 2018
CodeBuild pemberitahuan pembaruan agen	CodeBuild Dokumentasi telah diperbarui dengan detail tentang cara Anda dapat menggunakan Amazon SNS untuk diberi tahu ketika versi baru CodeBuild agen dirilis. Untuk informasi selengkapnya, lihat Menerima notifikasi untuk versi AWS CodeBuild agen baru.	15 Juni 2018

Pembaruan lebih awal

Tabel berikut menjelaskan perubahan penting dalam setiap rilis Panduan Pengguna AWS CodeBuild sebelum Juni 2018.

Perubahan	Deskripsi	Date
Dukungan untuk build Windows	CodeBuild sekarang mendukung build untuk platform Microsoft Windows Server, termasuk lingkunga n build yang sudah dikemas untuk .NET Core 2.0 di Windows. Untuk informasi selengkapnya, lihat Jalankan	25 Mei 2018

Perubahan	Deskripsi	Date
	sampel Microsoft Windows untuk CodeBuild.	
Dukungan untuk membangun idempotensi	Ketika Anda menjalankan `start-build` perintah dengan AWS Command Line Interface (AWS CLI), Anda dapat menentukan bahwa build tersebut idempotent. Untuk informasi selengkap nya, lihat Jalankan build (AWS CLI).	15 Mei 2018
Dukungan untuk mengesamp ingkan lebih banyak pengatura n proyek build	Anda sekarang dapat mengganti lebih banyak pengaturan proyek build saat Anda membuat build. Penggantian hanya untuk build itu. Untuk informasi selengkapnya, lihat Jalankan AWS CodeBuild build secara manual.	15 Mei 2018
Dukungan VPC Endpoint	Anda sekarang dapat menggunakan titik akhir VPC untuk meningkatkan keamanan build Anda. Untuk informasi selengkapnya, lihat Gunakan titik akhir VPC.	Maret 18, 2018
Dukungan pemicu	Anda sekarang dapat membuat pemicu untuk menjadwalkan build pada frekuensi reguler. Untuk informasi selengkapnya, lihat Buat AWS CodeBuild pemicu.	28 Maret 2018

Perubahan	Deskripsi	Date
Dokumentasi titik akhir FIPS	Anda sekarang dapat mempelajari cara menggunakan AWS Command Line Interface (AWS CLI) atau AWS SDK untuk memberitahu CodeBuild untuk menggunakan salah satu dari empat titik akhir Standar Pemrosesan Informasi Federal (FIPS). Untuk informasi selengkapnya, lihat Tentukan AWS CodeBuild titik akhir.	28 Maret 2018
AWS CodeBuild tersedia di Asia Pasifik (Mumbai), Eropa (Paris), dan Amerika Selatan (São Paulo)	AWS CodeBuild sekarang tersedia di wilayah Asia Pasifik (Mumbai), Eropa (Paris), dan Amerika Selatan (São Paulo). Untuk informasi selengkapnya, lihat AWS CodeBuild di Referensi Umum Amazon Web Services.	28 Maret 2018
GitHub Dukungan Server Perusahaan	CodeBuild sekarang dapat membangun dari kode sumber yang disimpan dalam repositori Server GitHub Perusahaan. Untuk informasi selengkapnya, lihat Jalankan sampel Server GitHub Perusahaan.	Januari, 25, 2018

Perubahan	Deskripsi	Date
Dukungan kedalaman klon Git	CodeBuild sekarang mendukung pembuatan klon dangkal dengan riwayat terpotong ke jumlah commit yang ditentukan. Untuk informasi selengkapnya, lihat Membuat proyek build.	Januari, 25, 2018
Dukungan VPC	VPC-enabled build sekarang dapat mengakses sumber daya di dalam VPC Anda. Untuk informasi selengkapnya, lihat Dukungan VPC.	November, 27, 2017
Dukungan caching ketergant ungan	CodeBuild sekarang mendukung caching ketergant ungan. Ini memungkinkan CodeBuild untuk menyimpan bagian tertentu yang dapat digunakan kembali dari lingkungan build di cache dan menggunakannya di seluruh build.	November, 27, 2017
Membangun dukungan lencana	CodeBuild sekarang mendukung penggunaa n lencana build, yang menyediakan gambar (badge) yang dapat disematkan dan dihasilkan secara dinamis yang menampilkan status build terbaru untuk sebuah proyek. Untuk informasi selengkapnya, lihat Bangun contoh lencana.	27 November 2017

Perubahan	Deskripsi	Date
AWS Config integrasi	AWS Config sekarang mendukung CodeBuild sebagai AWS sumber daya, yang berarti layanan dapat melacak CodeBuild proyek Anda. Untuk informasi lebih lanjut tentang AWS Config, lihatAWS Config sampel.	20 Oktober 2017
Secara otomatis membangun kembali kode sumber yang diperbarui di GitHub repositori	Jika kode sumber Anda disimpan dalam GitHub repositori, Anda dapat mengaktifkan AWS CodeBuild untuk membangun kembali kode sumber Anda setiap kali perubahan kode didorong ke repositori. Untuk informasi selengkapnya, lihat Jalankan permintaan GitHub tarik dan sampel filter webhook.	21 September 2017

Perubahan	Deskripsi	Date
Cara baru untuk menyimpan dan mengambil variabel lingkungan sensitif atau besar di Amazon EC2 Systems Manager Parameter Store	Anda sekarang dapat menggunakan AWS CodeBuild konsol atau AWS CLI untuk mengambil variabel lingkungan sensitif atau besar yang disimpan di Amazon EC2 Systems Manager Parameter Store. Sekarang Anda juga dapat menggunakan AWS CodeBuild konsol untuk menyimpan jenis variabel lingkungan ini di Amazon EC2 Systems Manager Parameter Store. Sebelumnya, Anda hanya dapat mengambil jenis variabel lingkunga n ini dengan memasukka nnya ke dalam buildspec atau dengan menjalank an perintah build untuk mengotomatiskan. AWS CLI Anda hanya dapat menyimpan jenis variabel lingkunga n ini dengan menggunak an konsol Amazon EC2 Systems Manager Parameter Store. Untuk informasi lebih lanjut, lihatMembuat proyek build,Ubah pengaturan proyek build, danJalankan build secara manual.	14 September 2017

Perubahan	Deskripsi	Date
Membangun dukungan penghapusan	Anda sekarang dapat menghapus build di AWS CodeBuild. Untuk informasi selengkapnya, lihat Hapus build.	31 Agustus 2017
Cara yang diperbarui untuk mengambil variabel lingkungan sensitif atau besar yang disimpan di Amazon EC2 Systems Manager Parameter Store dengan menggunakan buildspec	AWS CodeBuild sekarang memudahkan penggunaan buildspec untuk mengambil variabel lingkungan sensitif atau besar yang disimpan di Amazon EC2 Systems Manager Parameter Store. Sebelumnya, Anda hanya dapat mengambil jenis variabel lingkungan ini dengan menjalankan perintah build untuk mengotomatiskan AWS CLI. Untuk informasi lebih lanjut, lihat pem `parameter-store` etaan diSintaks Buildspec.	10 Agustus 2017
AWS CodeBuild mendukung Bitbucket	CodeBuild sekarang dapat membangun dari kode sumber yang disimpan dalam repositori Bitbucket. Untuk informasi lebih lanjut, lihat Membuat proyek build danJalankan build secara manual.	10 Agustus 2017

Perubahan	Deskripsi	Date
AWS CodeBuild tersedia di AS Barat (California Utara), Eropa (London), dan Kanada (Tengah)	AWS CodeBuild sekarang tersedia di wilayah AS Barat (California Utara), Eropa (London), dan Kanada (Tengah). Untuk informasi selengkapnya, lihat AWS CodeBuild di Referensi Umum Amazon Web Services.	29 Juni 2017
Nama dan lokasi file buildspec alternatif didukung	Anda sekarang dapat menentukan nama file alternatif atau lokasi file buildspec untuk digunakan untuk proyek build, bukan file buildspec default yang dinamai `buildspec.yml` di root kode sumber. Untuk informasi selengkapnya, lihat Nama file buildspec dan lokasi penyimpanan.	27 Juni 2017
Contoh pemberitahuan build yang diperbarui	CodeBuild sekarang menyediakan dukungan bawaan untuk pembuatan pemberitahuan melalui Amazon Ev CloudWatch ents dan Amazon Simple Notificat ion Service (Amazon SNS). Yang sebelumnya Buat contoh notifikasi telah diperbarui untuk menunjukkan perilaku baru ini.	22 Juni 2017

Perubahan	Deskripsi	Date
Docker dalam sampel gambar khusus ditambahkan	Contoh yang menunjukkan cara menggunakan CodeBuild dan image build Docker khusus untuk membangun dan menjalankan image Docker telah ditambahkan. Untuk informasi lebih lanjut, lihatDocker dalam sampel gambar khusus.	7 Juni 2017
Ambil kode sumber untuk permintaan GitHub tarik	Saat Anda menjalankan build CodeBuild yang bergantun g pada kode sumber yang disimpan dalam GitHub repositori, Anda sekarang dapat menentukan ID permintaan GitHub tarik untuk membangun. Anda juga dapat menentukan ID komit, nama cabang, atau nama tag sebagai gantinya. Untuk informasi selengkap nya, lihat nilai versi Sumber di Jalankan build (konsol) atau `sourceVersion` nilai diJalankan build (AWS CLI).	6 Juni 2017

Perubahan	Deskripsi	Date
Versi spesifikasi build diperbarui	Versi baru dari format buildspec telah dirilis. Versi 0.2 mengatasi masalah CodeBuild menjalankan setiap perintah build dalam instance terpisah dari shell default. Juga di versi 0.2, `environment_variables` diganti namanya menjadi`env`, dan `plaintext` diganti namanya menjadi`variables` . Untuk informasi selengkapnya, lihat Buat referensi spesifikasi untuk CodeBuild.	9 Mei 2017
Dockerfiles untuk membangun gambar tersedia di GitHub	Definisi untuk banyak gambar build yang disediakan oleh AWS CodeBuild tersedia sebagai Dockerfiles di. GitHub Untuk informasi selengkap nya, lihat kolom Definisi tabel diGambar Docker disediakan oleh CodeBuild.	2 Mei 2017
AWS CodeBuild tersedia di Eropa (Frankfurt), Asia Pasifik (Singapura), Asia Pasifik (Sydney), dan Asia Pasifik (Tokyo)	AWS CodeBuild sekarang tersedia di wilayah Eropa (Frankfurt), Asia Pasifik (Singapura), Asia Pasifik (Sydney), dan Asia Pasifik (Tokyo). Untuk informasi selengkapnya, lihat AWS CodeBuild di Referensi Umum Amazon Web Services.	21 Maret 2017

Perubahan	Deskripsi	Date
CodePipeline dukungan tindakan uji untuk CodeBuild	Anda sekarang dapat menambahkan ke pipeline CodePipeline dalam tindakan pengujian yang menggunakan CodeBuild. Untuk informasi selengkapnya, lihat Menambahkan tindakan CodeBuild pengujian ke pipeline (CodePipeline konsol).	8 Maret 2017
File buildspec mendukung pengambilan output build dari dalam direktori tingkat atas yang dipilih	File buildspec sekarang memungkinkan Anda untuk menentukan direktori tingkat atas individual yang isinya dapat Anda instruksikan CodeBuild untuk disertakan dalam artefak keluaran build. Anda melakukan ini dengan menggunakan pem `base-directory` etaan. Untuk informasi selengkapnya, lihat Sintaks Buildspec.	8 Februari 2017

Perubahan	Deskripsi	Date
Built-in variabel lingkungan	AWS CodeBuild menyediakan variabel lingkungan bawaan tambahan untuk digunakan build Anda. Ini termasuk variabel lingkungan yang menggambarkan entitas yang memulai pembuatan, URL ke repositori kode sumber, ID versi kode sumber, dan banyak lagi. Untuk informasi selengkapnya, lihat Variabel lingkungan di lingkungan build.	30 Januari 2017
AWS CodeBuild tersedia di AS Timur (Ohio)	AWS CodeBuild sekarang tersedia di wilayah AS Timur (Ohio). Untuk informasi selengkapnya, lihat AWS CodeBuild di Referensi Umum Amazon Web Services.	Januari 19, 2017
Informasi perilaku shell dan perintah	CodeBuild menjalankan setiap perintah yang Anda tentukan dalam instance terpisah dari shell default lingkunga n build. Perilaku default ini dapat menghasilkan beberapa efek samping yang tidak terduga untuk perintah Anda. Kami merekomendasikan beberapa pendekatan untuk mengatasi perilaku default ini jika diperlukan. Untuk informasi selengkapnya, lihat Shell dan perintah di lingkunga n build.	9 Desember 2016

Perubahan	Deskripsi	Date
Informasi variabel lingkungan	CodeBuild menyediakan beberapa variabel lingkungan yang dapat Anda gunakan dalam perintah build Anda. Anda juga dapat menentukan variabel lingkungan Anda sendiri. Untuk informasi selengkapnya, lihat Variabel lingkungan di lingkungan build.	Desember 7, 2016
Topik pemecahan masalah	Informasi pemecahan masalah sekarang tersedia. Untuk informasi selengkapnya, lihat Pemecahan Masalah AWS CodeBuild.	5 Desember 2016
Rilis awal plugin Jenkins	Ini adalah rilis awal plugin CodeBuild Jenkins. Untuk informasi selengkapnya, lihat Gunakan AWS CodeBuild dengan Jenkins.	5 Desember 2016
Panduan Pengguna rilis awal	Ini adalah rilis awal Panduan Pengguna CodeBuild .	1 Desember 2016

Terjemahan disediakan oleh mesin penerjemah. Jika konten terjemahan yang diberikan bertentangan dengan versi bahasa Inggris aslinya, utamakan versi bahasa Inggris.