Add a permission의

Amazon EBS



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

Amazon EBS: Add a permission의

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

Amazon의 상표 및 브랜드 디자인은 Amazon 외 제품 또는 서비스와 관련하여 고객에게 혼동을 일으킬 수 있는 방식이나 Amazon 브랜드 이미지를 떨어뜨리는 방식으로 사용할 수 없습니다. Amazon이 소유하지 않은 기타 모든 상표는 Amazon과 제휴 관계이거나 관련이 있거나 후원 관계와 관계없이 해당 소유자의 자산입니다.

Table of Contents

Amazon EBS란 무엇입니까? 1
Amazon EBS의 기능 1
관련 서비스 2
Amazon EBS 액세스 2
가격 책정 3
Amazon EBS의 설정 4
에 가입 AWS 계정 4
(선택) Amazon EBS 암호화용 고객 관리형 키 생성 및 사용 4
(선택) Amazon EBS 스냅샷에 대한 퍼블릭 액세스 차단 사용 4
EBS 볼륨 7
기능 및 이점 8
데이터 가용성 8
데이터 지속성 9
데이터 암호화 9
데이터 보안 10
스냅샷 10
유연성 11
EBS 볼륨 유형 11
Solid State Drive(SSD) 볼륨 11
하드 디스크 드라이브(HDD) 볼륨 13
이전 세대 볼륨 14
범용 SSD 볼륨 15
Provisioned IOPS SSD 볼륨 20
처리량 최적화 HDD 및 콜드 HDD 볼륨 23
EBS 볼륨 제약 조건 33
스토리지 용량 34
서비스 제한 35
파티셔닝 체계 35
데이터 블록 크기 36
EBS 볼륨 및 NVMe 39
볼륨을 디바이스 이름에 매핑 39
I/O 작업 시간 제한 43
Abort 명령 44
볼륨 수명 주기 44

볼륨 생성 46
볼륨 공유 50
볼륨 복사 55
인스턴스에 볼륨 연결 61
여러 인스턴스에 볼륨 연결 63
볼륨을 사용할 수 있도록 만들기 72
볼륨 세부 정보 보기 85
볼륨 수정 91
인스턴스에서 볼륨 분리 115
볼륨 삭제 119
볼륨 바꾸기 121
상태 확인 124
볼륨 이벤트 127
손상된 볼륨 작업 128
I/O 자동 활성화 131
오류 테스트 133
사용 가능한 실험 134
고려 사항 134
I/O 오류 주입 일시 중지 134
지연 시간 주입 136
EBS 스냅샷 139
스냅샷 작동 방식 140
스냅샷 수명 주기 144
스냅샷 생성 145
스냅샷 정보 보기 151
스냅샷 복사 154
스냅샷 공유 167
스냅샷 아카이브 174
스냅샷 삭제 206
빠른 스냅샷 복원 210
고려 사항 211
요금 및 결제 212
볼륨 생성 크레딧 212
빠른 스냅샷 복원 구성 213
빠른 스냅샷 복원 상태 확인 216
빠른 스냅샷 복원을 사용하여 복원된 볼륨 보기 218

스냅샷 잠금 219
개념 219
고려 사항 222
액세스 제어 223
스냅샷 잠금 226
스냅샷 잠금 해제 229
스냅샷 잠금 설정 업데이트 230
스냅샷 잠금 모니터링 231
스냅샷에 대한 퍼블릭 액세스 차단 234
IAM 권한 236
퍼블릭 액세스 차단 구성 237
퍼블릭 액세스 차단 설정 보기 241
퍼블릭 액세스 차단 비활성화 243
퍼블릭 액세스 차단 모니터링 246
Outposts의 로컬 스냅샷 247
자주 묻는 질문(FAQ) 248
사전 조건 250
고려 사항 64
IAM을 통한 액세스 제어 251
로컬 스냅샷 작업 253
로컬 영역의 로컬 스냅샷 259
자주 묻는 질문(FAQ) 248
고려 사항 64
IAM을 통한 액세스 제어 261
EBS 암호화 266
EBS의 암호화 방식 266
스냅샷이 암호화된 경우에 EBS 암호화가 작동하는 방식 267
스냅샷이 암호화되지 않은 경우에 EBS 암호화가 작동하는 방식 267
사용할 수 없는 KMS 키가 데이터 키에 미치는 영향 268
로컬 영역, Outpost 및 Wavelength 영역 269
Local Zones 269
Outpost 및 Wavelength Zone 270
요구 사항 270
지원되는 볼륨 유형 270
지원되는 인스턴스 유형 270
사용자의 권한 270

인스턴스에 대한 권한 272
기본적으로 암호화 사용 272
기본 암호화 키 277
EBS 리소스 암호화 277
빈 볼륨 생성 시 암호화 278
암호화되지 않은 리소스 암호화 278
KMS 키 교체 279
예제 280
암호화되지 않은 볼륨(활성화되지 않은 암호화 기본 제공) 복원 280
암호화되지 않은 볼륨(활성화된 암호화 기본 제공) 복원 281
암호화되지 않은 스냅샷(활성화되지 않은 암호화 기본 제공) 복사 281
암호화되지 않은 스냅샷(활성화된 암호화 기본 제공) 복사 282
암호화된 볼륨의 재암호화 282
암호화된 스냅샷의 재암호화 283
암호화된 볼륨과 암호화되지 않은 볼륨 간 데이터 마이그레이션 284
암호화 결과 284
EBS 성능 287
Amazon EBS 성능 팁 287
EBS 최적화 인스턴스 사용 287
인스턴스 대역폭 구성 288
성능 계산 방법 이해 288
워크로드 이해 288
스냅샷에서 볼륨을 초기화하는 경우 성능 저하에 유의 288
HDD 성능을 저하시킬 수 있는 요인 288
st1 및 sc1에서 처리량이 많은 읽기 중심 워크로드의 미리 읽기 향상(Linux 인스턴스에만 해당) 289
최신 Linux 커널 사용(Linux 인스턴스에만 해당) 289
RAID 0을 사용하여 인스턴스 리소스 활용도 극대화 290
Amazon EBS 볼륨 성능 모니터링 290
EBS 최적화 291
볼륨 초기화 291
볼륨 초기화에 Amazon EBS 프로비저닝 속도 사용 292
빠른 스냅샷 복원이 활성화된 스냅샷을 사용합니다. 297
볼륨 수동 초기화 297
볼륨 초기화 모니터링 301
구성 가능한 인스턴스 대역폭 가중치 305

I/O 특성 및 모니터링 306
IOPS 307
볼륨 대기열 길이 및 지연 시간 308
I/O 크기 및 볼륨 처리량 제한이 없음 309
CloudWatch를 사용하여 I/O 특성 모니터링 310
실시간 I/O 성능 통계 모니터링 311
관련 리소스 311
RAID 구성 312
RAID 구성 옵션 312
RAID 0 어레이 생성 313
RAID 어레이에 볼륨 스냅샷 생성 321
EBS 볼륨 벤치마킹 322
인스턴스 설정 322
벤치마크 도구 설치 324
볼륨 대기열 길이 선택 325
C 상태 비활성화 326
벤치마킹 수행 327
Amazon Data Lifecycle Manager 331
할당량 331
작동 방식 332
정책 332
정책 일정 333
대상 리소스 태그 334
스냅샷 334
EBS-backed AMI 335
Amazon Data Lifecycle Manager 태그 335
기본 정책 대 사용자 지정 정책 335
EBS 스냅샷 정책 비교 336
EBS 지원 AMI 정책 비교 337
기본 정책 생성 339
기본 정책 고려 사항 339
Amazon EBS 스냅샷에 대한 기본 정책 생성 340
EBS 지원 AMI에 대한 기본 정책 생성 344
여러 계정 및 리전에서 기본 정책 활성화 347
스냅샷에 대한 사용자 지정 정책 생성 352
스냅샷 수명 주기 정책 생성 352

스냅샷 수명 주기 정책 고려 사항 367
추가 리소스 372
애플리케이션에 일관되게 적용되는 스냅샷 자동화 372
사전 및 사후 스크립트의 기타 사용 사례 408
사전 및 사후 스크립트 작동 방식 416
사전 및 사후 스크립트로 생성된 스냅샷 식별 419
사전 및 사후 스크립트 모니터링 420
AMI에 대한 사용자 지정 정책 생성 421
AMI 수명 주기 정책 생성 421
AMI 수명 주기 정책 고려 사항 428
추가 리소스 431
교차 계정 스냅샷 복사 자동화 431
교차 계정 스냅샷 복사 정책 생성 432
스냅샷 설명 필터 지정 443
교차 계정 스냅샷 복사 정책 고려 사항 443
추가 리소스 444
정책 수정 444
정책 삭제 447
액세스 제어 448
AWS 관리형 정책 450
IAM 서비스 역할 458
정책 모니터링 464
콘솔 및 AWS CLI 465
EventBridge를 사용하여 정책 모니터링 465
CloudWatch를 사용하여 정책 모니터링 467
을 사용하여 AWS CloudTrail API 호출 로깅 481
서비스 엔드포인트 485
IPv4 엔드포인트 485
이중 스택(IPv4 및 IPv6) 엔드포인트 486
FIPS 엔드포인트 486
엔드포인트 지정 486
인터페이스 VPC 엔드포인트 487
Amazon EBS VPC 엔드포인트에 대한 고려 사항 487
Amazon EBS에 대한 인터페이스 VPC 엔드포인트를 생성합니다. 488
문제 해결 488
오류: `Role with name already exists` 489

AWS Backup Amazon EBS용 ... 490
Amazon EBS 다이렉트 API ... 491
가격 책정 ... 492
API 요금 ... 492
네트워킹 비용 ... 492
개념 ... 493
스냅샷 ... 493
블록 ... 493
블록 인덱스 ... 493
블록 토큰 ... 493
체크섬 ... 493
암호화(Encryption) ... 494
API 작업 ... 494
서명 버전 4 서명 ... 494
액세스 제어 ... 495
스냅샷 읽기 ... 500
스냅샷 블록 나열 ... 501
두 스냅샷에서 차이가 있는 블록 나열 ... 504
스냅샷에서 블록 데이터 가져오기 ... 507
스냅샷 쓰기 ... 508
스냅샷 시작 ... 510
스냅샷에 데이터 추가 ... 511
스냅샷 완료 ... 513
암호화 결과 ... 514
암호화 결과: 암호화되지 않은 상위 스냅샷 ... 515
암호화 결과: 암호화된 상위 스냅샷 ... 515
암호화 결과: 상위 스냅샷 없음 ... 516
스냅샷 데이터 검증 ... 517
멱등성 보장 ... 518
오류 재시도 횟수 ... 519
성능 최적화 ... 521
서비스 엔드포인트 ... 522
IPv4 엔드포인트 ... 523
이중 스택(IPv4 및 IPv6) 엔드포인트 ... 523
FIPS 엔드포인트 ... 524
엔드포인트 지정 ... 524

SDK 코드 예제 526
StartSnapshot 526
PutSnapshotBlock 527
CompleteSnapshot 528
인터페이스 VPC 엔드포인트 529
EBS 다이렉트 API VPC 엔드포인트에 대한 고려 사항 529
EBS 다이렉트 API에 대한 인터페이스 VPC 엔드포인트 생성 530
CloudTrail 로그 531
CloudTrail의 EBS 다이렉트 API 데이터 이벤트 532
CloudTrail의 EBS 다이렉트 API 관리 이벤트 533
EBS 다이렉트 API 이벤트 예제 482
FAQ 539
휴지통 542
지원되는 리소스 543
어떻게 작동하나요? 543
고려 사항 544
할당량 548
관련 서비스 548
가격 책정 548
액세스 제어 549
휴지통 및 보존 규칙 작업을 위한 권한 549
휴지통의 리소스 작업을 위한 권한 551
휴지통에 사용되는 조건 키 551
보존 규칙 생성 554
보존 규칙 업데이트 558
보존 규칙 잠금 560
보존 규칙 잠금 해제 561
태그 보존 규칙 563
보존 규칙 태그 보기 564
보존 규칙에서 태그 제거 565
보존 규칙 삭제 566
삭제된 스냅샷 복구 566
휴지통의 스냅샷 작업을 위한 권한 567
휴지통의 스냅샷 보기 568
휴지통에서 스냅샷 복원 569
삭제된 볼륨 복구 570

휴지통의 볼륨 작업에 대한 권한 571
휴지통에서 볼륨 보기 572
휴지통에서 볼륨 복원 573
삭제된 AMI 복구 575
휴지통의 AMI 작업을 위한 권한 575
휴지통의 AMI 보기 576
휴지통에서 AMI 복원 578
EventBridge를 사용하여 모니터링 579
RuleLocked 579
RuleChangeAttempted 580
RuleUnlockScheduled 581
RuleUnlockingNotice 581
RuleUnlocked 582
CloudTrail을 사용하여 모니터링 583
CloudTrail의 휴지통 정보 481
휴지통 로그 파일 항목 이해 584
서비스 엔드포인트 597
IPv4 엔드포인트 523
이중 스택(IPv4 및 IPv6) 엔드포인트 598
FIPS 엔드포인트 599
엔드포인트 지정 599
인터페이스 VPC 엔드포인트 사용 600
휴지통용 인터페이스 VPC 엔드포인트 생성 600
휴지통에 대한 VPC 엔드포인트 정책 생성 600
보안 602
데이터 보호 602
Amazon EBS 데이터 보안 603
저장 데이터 및 전송 데이터 암호화 604
KMS 키 관리 604
ID 및 액세스 관리 605
대상 605
ID를 통한 인증 605
정책을 사용하여 액세스 관리 607
EBS에서 IAM을 사용하는 방법 608
예제 IAM 정책 613
문제 해결 631

규정 준수 확인 633
데이터 복원력 634
모니터링 635
Amazon CloudWatch 635
Amazon EBS 볼륨 지표 636
Amazon EBS 스냅샷에 대한 지표 653
Nitro 인스턴스 관련 지표 654
빠른 스냅샷 복원 관련 지표 657
Amazon EC2 콘솔 그래프 658
Amazon EventBridge 660
EBS 볼륨 이벤트 661
EBS 볼륨 초기화 이벤트 667
EBS 볼륨 수정 이벤트 668
EBS 스냅샷 이벤트 668
EBS 스냅샷 아카이브 이벤트 676
EBS 빠른 스냅샷 복원 이벤트 676
AWS Lambda 를 사용하여 EventBridge 이벤트 처리 678
EBS 세부 성능 통계 681
통계 681
통계 액세스 683
Amazon GuardDuty 685
결제 및 사용량 보고서 686
스냅샷 687
볼륨 스토리지 687
프로비저닝된 성능 688
EBS 다이렉트 API 690
빠른 스냅샷 복원(FSR) 690
EBS 볼륨 인벤토리 작성 690
볼륨 691
I/O 성능 693
EBS 데이터 볼륨의 콘텐츠 694
스냅샷 694
EBS 스냅샷의 콘텐츠 695
할당량 697
문서 기록 713
..... dccxxiii

Amazon Elastic Block Store란 무엇입니까?

Amazon Elastic Block Store(Amazon EBS)에서는 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스와 함께 사용할 수 있는 확장 가능한 고성능 블록 스토리지 리소스가 제공됩니다. Amazon Elastic Block Store에서 다음과 같은 블록 스토리지 리소스를 생성하고 관리할 수 있습니다.

- Amazon EBS 볼륨 - Amazon EC2 인스턴스에 연결하는 스토리지 볼륨입니다. 볼륨을 인스턴스에 연결하면 해당 볼륨을 컴퓨터에 연결된 로컬 하드 드라이브처럼 사용할 수 있습니다(예: 파일 저장 또는 애플리케이션 설치).
- Amazon EBS 스냅샷 - 볼륨 자체와 관계없이 지속되는 Amazon EBS 볼륨의 특정 시점 백업입니다. Amazon EBS 볼륨의 데이터를 백업하는 스냅샷을 생성할 수 있습니다. 그러면 언제든지 해당 스냅샷에서 새 볼륨을 복원할 수 있습니다.

주제

- Amazon EBS의 기능
- 관련 서비스
- Amazon EBS 액세스
- 가격 책정

Amazon EBS의 기능

Amazon EBS에서는 다음과 같은 기능과 이점이 제공됩니다.

- 여러 가지 볼륨 유형 - Amazon EBS에서는 광범위한 애플리케이션의 스토리지 성능과 비용을 최적화할 수 있는 여러 가지 볼륨 유형이 제공됩니다. 볼륨 유형은 트랜잭션 워크로드를 위한 SSD 지원 스토리지와 처리량 집약적 워크로드를 위한 HDD 지원 스토리지라는 두 가지 주요 범주로 나뉩니다.
- 확장성 - 필요성이 충족되는 용량 및 성능 사양으로 Amazon EBS 볼륨을 생성할 수 있습니다. 필요성이 변경되면 탄력적 볼륨 작업을 사용하여 가동 중지 시간 없이 동적으로 용량을 늘리거나 성능을 조정할 수 있습니다.
- 백업 및 복구 - Amazon EBS 스냅샷을 사용하여 볼륨에 저장된 데이터를 백업합니다. 그런 다음 이러한 스냅샷을 사용하여 볼륨을 즉시 복원하거나 AWS 계정, AWS 리전 또는 가용 영역 간에 데이터를 마이그레이션할 수 있습니다.

- 데이터 보호 - Amazon EBS 암호화를 사용하여 Amazon EBS 볼륨과 Amazon EBS 스냅샷을 암호화합니다. 암호화 작업은 Amazon EC2 인스턴스를 호스팅하는 서버에서 발생합니다. 이렇게 하면 인스턴스와 연결된 볼륨 및 후속 스냅샷 간에 data-at-rest data-in-transit 모두 보호할 수 있습니다.
- 데이터 가용성 및 내구성 - io2 Block Express 볼륨은 연간 장애율이 0.001%인 99.999% 내구성을 갖추고 있습니다. 기타 볼륨 유형의 내구성은 99.8~99.9%이며, 연간 장애율은 0.1~0.2%입니다. 또한 단일 구성 요소의 장애로 인한 데이터 손실이 방지되도록 볼륨 데이터가 가용 영역의 여러 서버에 자동으로 복제됩니다.
- 데이터 아카이브 - EBS 스냅샷 아카이브는 EBS 스냅샷의 전체 point-in-time 사본을 아카이브할 수 있는 저비용 스토리지 계층을 제공합니다. 규제, 규정 준수 또는 향후 프로젝트 릴리스를 위해 스냅샷을 90일 이상 보존해야 하는 경우 Archive를 사용합니다.

관련 서비스

Amazon EBS는 다음과 같은 서비스와 연동합니다.

- Amazon Elastic Compute Cloud - AWS 클라우드에서 가상 머신(Amazon EC2 인스턴스)을 시작하고 관리할 수 있는 서비스입니다. EBS 볼륨을 해당 인스턴스에 연결하여 로컬 하드 드라이브처럼 사용할 수 있습니다(예: 파일 저장 또는 애플리케이션 설치). 자세한 내용은 Amazon EC2란 무엇입니까?를 참조하세요.
- AWS Key Management Service - 암호화 키를 생성하고 관리할 수 있는 관리형 서비스입니다. AWS KMS 암호화 키를 사용하여 Amazon EBS 볼륨 및 Amazon EBS 스냅샷에 저장된 데이터를 암호화할 수 있습니다. 자세한 내용은 Amazon EBS에서 AWS KMS를 사용하는 방법을 참조하세요.
- Amazon Data Lifecycle Manager - EBS 스냅샷 및 EBS 지원 AMI의 생성, 유지 및 삭제를 자동화하는 관리형 서비스입니다. Amazon Data Lifecycle Manager를 사용하여 Amazon EBS 볼륨 및 Amazon EC2 인스턴스 백업을 자동화할 수 있습니다. 자세한 내용은 Amazon Data Lifecycle Manager를 사용하여 백업 자동화 단원을 참조하십시오.
- EBS 다이렉트 API - EBS 스냅샷을 생성하고, 스냅샷에 직접 데이터를 쓰고, 스냅샷에서 데이터를 읽고, 두 스냅샷 간의 차이점 또는 변경 사항을 식별할 수 있는 서비스입니다. 자세한 내용은 EBS 다이렉트 API를 사용하여 EBS 스냅샷 콘텐츠에 액세스 단원을 참조하십시오.
- 휴지통 - 실수로 삭제한 Amazon EBS 스냅샷과 EBS 지원 AMI를 복원할 수 있는 데이터 복구 서비스입니다. 자세한 내용은 휴지통을 참조하세요.

Amazon EBS 액세스

다음과 같은 인터페이스를 사용하여 Amazon EBS 리소스를 생성하고 관리할 수 있습니다.

Amazon EC2 콘솔

볼륨과 스냅샷을 생성하고 관리하는 웹 인터페이스입니다. AWS 계정에 가입한 경우 https://console.aws.amazon.com/ec2/ Amazon EC2 콘솔에 액세스할 수 있습니다.

AWS Command Line Interface

명령줄 쉘의 명령을 사용하여 Amazon EBS 리소스를 관리할 수 있는 명령줄 도구입니다. Windows, Mac, Linux에서 지원됩니다. 자세한 내용은 AWS Command Line Interface 사용 설명서 및 ec2 명령을 참조하세요.

AWS Tools for PowerShell

PowerShell 명령줄에서 Amazon EBS 리소스에 대한 작업을 스크립팅할 수 있는 PowerShell 모듈 세트입니다. 자세한 내용은 AWS Tools for PowerShell 사용 설명서와 AWS Tools for PowerShell Cmdlet 참조를 참조하세요.

CloudFormation

AWS 리소스를 설명하는 재사용 가능한 JSON 또는 YAML 템플릿을 생성한 다음 해당 리소스를 프로비저닝하고 구성할 수 있는 완전 관리형 AWS 서비스입니다. 자세한 내용은 AWS CloudFormation 사용 설명서를 참조하십시오.

Amazon EC2 쿼리 API

Amazon EC2 쿼리 API에서는 HTTP 동사 `GET` 또는 `POST` 및 이름이 `Action`인 쿼리 파라미터를 사용하는 HTTP 또는 HTTPS 요청이 제공됩니다. 자세한 내용은 Amazon EC2 API 참조를 참조하세요.

AWS SDKs

AWS 서비스와 통합된 애플리케이션을 구축할 수 있는 언어별 APIs. AWS SDKs는 널리 사용되는 많은 프로그래밍 언어에 사용할 수 있습니다. 자세한 내용은 빌드 기반 도구를 참조하세요 AWS.

가격 책정

Amazon EBS에서는 프로비저닝한 만큼만 지불하면 됩니다. 자세한 내용은 Amazon EBS 요금을 참조하세요.

Amazon EBS의 설정

Amazon EBS 리소스로 작업하도록 설정하려면 이 섹션의 태스크를 완료합니다.

작업

- 에 가입 AWS 계정
- (선택) Amazon EBS 암호화용 고객 관리형 키 생성 및 사용
- (선택) Amazon EBS 스냅샷에 대한 퍼블릭 액세스 차단 사용

에 가입 AWS 계정

를 시작하려면이 AWS필요합니다 AWS 계정. 생성에 대한 자세한 AWS 계정내용은 AWS Account Management 참조 안내서의 시작하기 AWS 계정를 참조하세요.

(선택) Amazon EBS 암호화용 고객 관리형 키 생성 및 사용

Amazon EBS 암호화는 AWS KMS 암호화 키를 사용하여 Amazon EBS 볼륨 및 Amazon EBS 스냅샷을 암호화하는 암호화 솔루션입니다. Amazon EBS는 각 리전에서 Amazon EBS 암호화를 위한 고유한 AWS 관리형 KMS 키를 자동으로 생성합니다. 이 KMS 키에는 별칭 `aws/ebs`가 있습니다. 기본 KMS 키를 교체하거나 해당 권한을 관리할 수 없습니다. Amazon EBS 암호화에 사용되는 KMS 키의 유연성과 제어를 강화하려면 고객 관리형 키를 생성하여 사용하는 것이 좋습니다.

Amazon EBS 암호화용 고객 관리형 키를 생성하고 사용하는 방법

1. 대칭 암호화 KMS 키를 생성합니다.
2. Amazon EBS 암호화용 기본 KMS 키인 KMS 키를 선택합니다.
3. Amazon EBS 암호화용 KMS 키를 사용하는 권한을 사용자에게 부여합니다.

(선택) Amazon EBS 스냅샷에 대한 퍼블릭 액세스 차단 사용

스냅샷의 퍼블릭 공유를 방지하려면 스냅샷에 대한 퍼블릭 액세스 차단을 활성화합니다. 리전에서 스냅샷에 대한 퍼블릭 액세스 차단을 활성화하면 해당 리전에서 스냅샷을 공개적으로 공유하려는 모든 시도가 자동으로 차단됩니다. 이를 통해 스냅샷의 보안을 강화하고 무단 액세스나 의도하지 않은 액세스로부터 스냅샷 데이터를 보호할 수 있습니다.

자세한 내용은 Amazon EBS 스냅샷에 대한 퍼블릭 액세스 차단 섹션을 참조하세요.

Console

스냅샷에 대한 퍼블릭 액세스 차단을 사용하는 방법

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 EC2 대시보드를 선택한 다음 계정 속성(오른쪽)에서 데이터 보호 및 보안을 선택합니다.
3. EBS 스냅샷에 대한 퍼블릭 액세스 차단 섹션에서 관리를 선택합니다.
4. 퍼블릭 액세스 차단을 선택한 후 다음 옵션 중 하나를 선택합니다.
   - 모든 퍼블릭 액세스 차단 - 스냅샷의 모든 퍼블릭 공유를 차단합니다. 계정의 사용자는 새 퍼블릭 공유를 요청할 수 없습니다. 또한 이미 공개적으로 공유된 스냅샷은 비공개로 취급되어 더 이상 공개적으로 사용할 수 없습니다.
   - 새 퍼블릭 공유 차단 - 스냅샷의 새 퍼블릭 공유만 차단합니다. 계정의 사용자는 새 퍼블릭 공유를 요청할 수 없습니다. 그러나 이미 공개적으로 공유된 스냅샷은 계속 공개적으로 사용할 수 있습니다.
5. 업데이트를 선택합니다.

AWS CLI

스냅샷에 대한 퍼블릭 액세스 차단을 사용하는 방법

enable-snapshot-block-public-access 명령을 사용합니다. `--state`에 대해 다음 값 중 하나를 지정합니다.

- `block-all-sharing` - 스냅샷의 모든 퍼블릭 공유를 차단합니다. 계정의 사용자는 새 퍼블릭 공유를 요청할 수 없습니다. 또한 이미 공개적으로 공유된 스냅샷은 비공개로 취급되어 더 이상 공개적으로 사용할 수 없습니다.
- `block-new-sharing` - 스냅샷의 모든 퍼블릭 공유만 차단합니다. 계정의 사용자는 새 퍼블릭 공유를 요청할 수 없습니다. 그러나 이미 공개적으로 공유된 스냅샷은 계속 공개적으로 사용할 수 있습니다.

```
aws ec2 enable-snapshot-block-public-access --state block-new-sharing
```

PowerShell

스냅샷에 대한 퍼블릭 액세스 차단 사용

Enable-EC2SnapshotBlockPublicAccess cmdlet을 사용합니다. `-State`에 대해 다음 값 중 하나를 지정합니다.

- `block-all-sharing` - 스냅샷의 모든 퍼블릭 공유를 차단합니다. 계정의 사용자는 새 퍼블릭 공유를 요청할 수 없습니다. 또한 이미 공개적으로 공유된 스냅샷은 비공개로 취급되어 더 이상 공개적으로 사용할 수 없습니다.
- `block-new-sharing` - 스냅샷의 모든 퍼블릭 공유만 차단합니다. 계정의 사용자는 새 퍼블릭 공유를 요청할 수 없습니다. 그러나 이미 공개적으로 공유된 스냅샷은 계속 공개적으로 사용할 수 있습니다.

```
Enable-EC2SnapshotBlockPublicAccess -State block-new-sharing
```

Amazon EBS 볼륨

Amazon EBS 볼륨은 내구성이 있는 블록 스토리지 디바이스이며 인스턴스를 연결하는 것이 가능합니다. 볼륨을 인스턴스에 연결하면 물리적 하드 드라이브처럼 사용할 수 있습니다. EBS 볼륨은 유연합니다. 현재 세대 인스턴스 유형에 연결된 현재 세대 볼륨의 경우 크기를 동적으로 늘리고 프로비저닝된 IOPS 용량을 수정하며 라이브 프로덕션 볼륨의 볼륨 유형을 변경할 수 있습니다.

인스턴스의 시스템 드라이브 또는 데이터베이스 애플리케이션용 스토리지 등 자주 업데이트해야 하는 데이터의 경우 EBS 볼륨을 기본 스토리지로 사용할 수 있습니다. 연속으로 디스크 스캔을 수행하는 처리량 집약적 애플리케이션에도 해당 볼륨을 사용할 수 있습니다. EBS 볼륨은 EC2 인스턴스의 실행 주기와는 독립적으로 유지됩니다.

여러 EBS 볼륨을 단일 인스턴스에 연결할 수 있습니다. 볼륨 및 인스턴스는 동일 가용 영역에 위치해야 합니다. 볼륨 및 인스턴스 유형에 따라 다중 연결을 사용하여 볼륨을 여러 인스턴스에 동시에 탑재할 수 있습니다.

Amazon EBS는 범용 SSD(gp2 및 gp3), 프로비저닝된 IOPS SSD(io1 및 io2), 처리량 최적화 HDD(st1), 콜드 HDD(sc1) 및 마그네틱(standard) 등의 볼륨 유형을 제공합니다. 이 두 유형은 성능 특성과 가격이 다르므로 애플리케이션의 필요에 맞게 스토리지 성능과 비용을 조정할 수 있습니다. 자세한 내용은 Amazon EBS 볼륨 유형 섹션을 참조하세요.

계정에 사용할 수 있는 총 저장 용량에는 제한이 있습니다. 이러한 제한값 및 제한값 증가 요청 방법에 대한 자세한 내용은 Amazon EBS 엔드포인트 및 할당량을 참조하세요.

관리형 EBS 볼륨은 Amazon EKS Auto Mode와 같은 서비스 제공업체가 관리합니다. 관리형 EBS 볼륨의 설정은 직접 수정할 수 없습니다. 관리형 EBS 볼륨은 관리형 필드에서 true 값으로 식별됩니다. 자세한 내용은 Amazon EC2 관리형 인스턴스를 참조하세요.

요금에 대한 자세한 내용은 Amazon EBS 요금을 참조하세요.

내용

- Amazon EBS 볼륨의 기능 및 이점
- Amazon EBS 볼륨 유형
- Amazon EBS 볼륨 제약 조건
- Amazon EBS 볼륨 및 NVMe
- Amazon EBS 볼륨 수명 주기
- 스냅샷을 사용하여 Amazon EBS 볼륨 바꾸기

- Amazon EBS 볼륨 상태 확인
- Amazon EBS에서 오류 테스트

Amazon EBS 볼륨의 기능 및 이점

EBS 볼륨은 인스턴스 스토어 볼륨과 차별화된 이점을 제공합니다.

이점

- 데이터 가용성
- 데이터 지속성
- 데이터 암호화
- 데이터 보안
- 스냅샷
- 유연성

데이터 가용성

EBS 볼륨을 생성하면 단일 하드웨어 구성 요소의 장애로 인한 데이터 손실을 방지하기 위해 해당 가용 영역 내에서 자동으로 복제됩니다. 동일한 가용 영역에 있는 EC2 인스턴스에 EBS 볼륨을 연결할 수 있습니다. 볼륨을 연결한 후에 인스턴스는 하드 드라이브 또는 기타 물리 드라이브와 같은 원시 블록 디바이스처럼 보입니다. 이 시점에 인스턴스는 로컬 드라이브와 동일한 방식으로 볼륨과 상호 작용할 수 있습니다. 인스턴스에 연결하고 파일 시스템(예: Linux 인스턴스는 `Ext4`, Windows 인스턴스는 NTFS)으로 EBS 볼륨의 형식을 지정한 다음에 애플리케이션을 설치할 수 있습니다.

사용자가 명명한 디바이스에 다중 볼륨이 연결된 경우 사용자는 I/O 및 처리 성능을 향상하기 위해 전체 볼륨에서 데이터를 스트라이프할 수 있습니다.

`io1` 및 `io2` EBS 볼륨을 최대 16개의 Nitro 기반 인스턴스에 연결할 수 있습니다. 자세한 내용은 다중 연결을 사용하여 여러 EC2 인스턴스에 EBS 볼륨 연결 섹션을 참조하세요. 아니면 EBS 볼륨을 단일 인스턴스에 연결할 수 있습니다.

추가 비용 없이 EBS 볼륨 및 EBS 기반 인스턴스의 루트 디바이스 볼륨의 데이터를 모니터링할 수 있습니다. 측정치의 모니터링에 대한 자세한 내용은 Amazon EBS에 대한 Amazon CloudWatch 지표 섹션을 참조하세요. 볼륨 상태 추적에 대한 자세한 내용은 Amazon EBS용 Amazon EventBridge 이벤트 섹션을 참조하세요.

데이터 지속성

EBS 볼륨은 인스턴스의 수명에 관계없이 유지되는 오프 인스턴스 스토리지입니다. 사용자는 데이터가 유지되는 동안 볼륨 사용량에 대한 비용을 계속해서 지불합니다.

실행 중인 인스턴스에 연결된 EBS 볼륨은 사용자가 EC2 콘솔에서 해당 인스턴스에 대한 EBS 볼륨을 구성할 때 종료 시 삭제 확인란을 선택하지 않은 경우 인스턴스가 종료될 때 해당 데이터가 원래 상태를 유지한 채로 해당 인스턴스에서 자동으로 분리될 수 있습니다. 그러면 해당 볼륨은 새 인스턴스로 재연결되어 빠른 복구가 가능합니다. 종료 시 삭제 확인란이 선택된 경우에는 EC2 인스턴스 종료 시 볼륨이 삭제됩니다. EBS 기반 인스턴스를 사용하는 경우 연결된 볼륨에 저장된 데이터에 영향을 주지 않고 해당 인스턴스를 중지하고 다시 시작할 수 있습니다. 해당 볼륨은 정지-시작 주기 동안 연결 상태를 유지합니다. 이를 통해 사용자는 필요할 때 처리 및 스토리지 리소스만을 사용하여 볼륨에서 데이터를 무기한으로 처리 및 저장할 수 있습니다. 데이터는 볼륨이 완전히 삭제될 때까지 볼륨에서 유지됩니다. 삭제된 EBS 볼륨에서 사용하는 물리적 블록 스토리지는 새 볼륨에 할당되기 전에 0 또는 암호화된 의사 난수 데이터로 덮어씁니다. 민감한 데이터를 사용하는 경우 데이터를 직접 암호화하거나 Amazon EBS 암호화로 보호되는 볼륨에 데이터를 저장해야 합니다. 자세한 내용은 Amazon EBS 암호화 섹션을 참조하세요.

기본적으로, 실행 시 생성되어 인스턴스에 연결된 루트 EBS 볼륨은 해당 인스턴스가 종료되면 삭제됩니다. 사용자는 인스턴스 시작 시 플래그 값을 `DeleteOnTermination`에서 `false`로 변경하여 해당 동작을 수정할 수 있습니다. 값이 수정되면 인스턴스가 종료된 후에도 볼륨이 유지되어 해당 볼륨에 다른 인스턴스를 연결할 수 있습니다.

기본적으로, 실행 시 생성되어 인스턴스에 연결된 추가 EBS 볼륨은 해당 인스턴스가 종료되면 삭제되지 않습니다. 사용자는 인스턴스 시작 시 플래그 값을 `DeleteOnTermination`에서 `true`로 변경하여 해당 동작을 수정할 수 있습니다. 이 수정된 값으로 인해 인스턴스가 종료될 때 볼륨이 삭제됩니다.

데이터 암호화

단순 데이터 암호화의 경우 Amazon EBS 암호화 기능으로 암호화된 EBS 볼륨을 생성할 수 있습니다. 모든 EBS 볼륨 유형은 암호화를 지원합니다. 암호화된 EBS 볼륨을 사용하여 규제/감사 데이터 및 애플리케이션에 대한 다양한 유휴 데이터 암호화 요구 사항을 충족할 수 있습니다. Amazon EBS 암호화는 256비트 고급 암호화 표준 알고리즘(AES-256)과 Amazon 관리형 키 인프라를 사용합니다. 암호화는 EC2 인스턴스를 호스트하는 서버에서 수행되므로 EC2 인스턴스에서 Amazon EBS 스토리지로 전송되는 데이터가 암호화됩니다. 자세한 내용은 Amazon EBS 암호화 섹션을 참조하세요.

Amazon EBS 암호화는 암호화된 볼륨과 암호화된 볼륨에서 생성된 스냅샷을 생성할 AWS KMS keys 때를 사용합니다. 리전에서 암호화된 EBS 볼륨을 처음 생성하면 기본 AWS 관리형 KMS 키가 자동으로 생성됩니다. 고객 관리형 키를 생성하여 사용하지 않는 한 이 키는 Amazon EBS 암호화에 사용

됩니다. 고객 관리형 키를 직접 생성하면 액세스 제어를 생성, 교체, 비활성화, 정의하고 데이터를 보호하는 데 사용된 암호화 키를 감사하는 등 보다 폭넓은 작업이 가능합니다. 자세한 내용은 AWS Key Management Service 개발자 안내서를 참조하세요.

데이터 보안

Amazon EBS 볼륨은 포맷되지 않은 원시 블록 디바이스로 제공됩니다. 이러한 디바이스는 EBS 인프라에서 생성되는 논리적 디바이스이며 Amazon EBS 서비스는 고객이 사용하거나 재사용하기 전에 디바이스가 논리적으로 비어 있는지(즉, 원시 블록이 0이 되거나 암호화된 의사 난수 데이터를 포함하는지) 확인합니다.

DoD 5220.22-M(국가 산업 보안 프로그램 운영 매뉴얼) 또는 NIST 800-88(미디어 삭제 지침)에 자세히 설명된 것과 같이 사용 후, 사용 전 또는 사용 전후에 특정 방법을 사용하여 모든 데이터를 지워야 하는 절차가 있는 경우 Amazon EBS에서 해당 작업을 수행할 수 있습니다. 해당 블록 수준 활동은 Amazon EBS 서비스 내의 기본 스토리지 미디어에 반영됩니다.

스냅샷

Amazon EBS를 사용하면 모든 EBS 볼륨의 스냅샷(백업)을 생성하고 볼륨 내 데이터 사본을 다중 가용 영역에 중복 저장이 가능한 Amazon S3에 작성할 수 있습니다. 볼륨이 실행 중인 인스턴스에 연결되어 있지 않아도 스냅샷을 만드는 데는 문제가 없습니다. 볼륨에 데이터를 계속에서 작성하면 새 볼륨의 기준으로 사용될 볼륨 스냅샷을 주기적으로 생성할 수 있습니다. 이 스냅샷을 사용하여 새로운 EBS 볼륨을 여러 개 생성하거나 가용 영역 간에 볼륨을 이동할 수 있습니다. 암호화된 EBS 볼륨의 스냅샷은 자동으로 암호화됩니다.

스냅샷에서 새로운 볼륨을 생성하는 경우 새로 생성된 스냅샷이 생성될 시점의 원본 볼륨 사본과 정확히 일치합니다. 암호화된 스냅샷에서 생성된 EBS 볼륨은 자동으로 암호화됩니다. 다양한 가용 영역을 지정하는 옵션이 있습니다. 이러한 기능을 사용하여 이 영역에 복제 볼륨을 생성할 수 있습니다. 스냅샷은 특정 AWS 계정과 공유하거나 공개될 수 있습니다. 스냅샷을 생성하는 경우 소스 볼륨의 크기가 아니라 백업되는 데이터의 크기에 따라 Amazon S3에서 비용이 발생합니다. 동일한 볼륨의 후속 스냅샷은 증분 스냅샷입니다. 마지막 스냅샷이 생성된 이후 볼륨에 작성된 변경된 데이터 및 새 데이터에만 포함되며, 이러한 변경된 데이터 및 새 데이터에 대해서만 비용이 청구됩니다.

스냅샷은 마지막 스냅샷 이후 변경된 볼륨의 블록만 저장되는 증분식 백업입니다. 100GiB 데이터를 가진 볼륨이 있지만 마지막 스냅샷 이후 5GiB만이 변경된 경우 변경된 5GiB만이 Amazon S3에 작성됩니다. 스냅샷은 증분식으로 저장되지만, 스냅샷 삭제 프로세스는 가장 최근의 스냅샷만 유지하도록 설계되어 있습니다.

볼륨 및 스냅샷을 쉽게 범주화하고 관리할 수 있도록 사용자는 원하는 메타데이터로 볼륨 및 스냅샷에 태그를 사용할 수 있습니다.

볼륨을 자동으로 백업하려는 경우 Amazon Data Lifecycle Manager 또는 AWS Backup를 사용할 수 있습니다.

유연성

EBS 볼륨은 프로덕션 중에 라이브 구성 변경을 지원합니다. 서비스 중단 없이 볼륨 유형, 볼륨 크기, IOPS 용량을 수정할 수 있습니다. 자세한 내용은 탄력적 볼륨 작업을 사용하여 Amazon EBS 볼륨 수정 섹션을 참조하세요.

Amazon EBS 볼륨 유형

Amazon EBS는 다음의 볼륨 유형을 제공하고 이러한 볼륨 유형은 성능 특성과 가격이 다르므로 애플리케이션의 필요에 맞게 스토리지 성능과 비용을 조정할 수 있습니다.

> **Important**
>
> 인스턴스 구성, I/O 특성 및 워크로드 요구량 등 여러 가지 요인이 EBS 볼륨의 성능에 영향을 미칠 수 있습니다. EBS 볼륨에서 프로비저닝된 IOPS를 완전히 사용하려면 EBS 최적화 인스턴스를 사용합니다. EBS 볼륨을 최대한 이용하는 방법에 대한 자세한 내용은 Amazon EBS 볼륨 성능을 참조하세요.

요금에 대한 자세한 내용은 Amazon EBS 요금을 참조하세요.

볼륨 유형

- Solid State Drive(SSD) 볼륨
- 하드 디스크 드라이브(HDD) 볼륨
- 이전 세대 볼륨

Solid State Drive(SSD) 볼륨

SSD 지원 볼륨은 작은 I/O 크기의 읽기/쓰기 작업을 자주 처리하며 기준 성능 속성은 IOPS인 트랜잭션 워크로드에 최적화되어 있습니다. SSD 지원 볼륨 유형으로는 범용 SSD와 프로비저닝된 IOPS SSD가 있습니다. 다음은 SSD 기반 볼륨의 사용 사례 및 특성에 대한 요약입니다.

	Amazon EBS 범용 SSD 볼륨		Amazon EBS 프로비저닝된 IOPS SSD 볼륨	
볼륨 유형	gp3 [6]	gp2	io2 Block Express	io1
내구성	99.8%~99.9% 내구성(연간 장애율 0.1%~0.2%)		99.999% 내구성(연간 장애율 0.001%)	99.8%~99.9% 내구성(연간 장애율 0.1%~0.2%)
사용 사례	• 트랜잭션 워크로드 • 가상 데스크톱 • 중간 규모의 단일 인스턴스 데이터베이스 • 짧은 대기 시간 대화형 애플리케이션 • 부트 볼륨 • 개발 및 테스트 환경		다음이 필요한 워크로드: • 평균 지연 시간이 500마이크로초 미만인 일관적인 밀리초 미만의 지연 시간 [5] • 지속적인 IOPS 성능 • 80,000 IOPS 또는 2,000MiB/s 이상의 처리량	• 지속적인 IOPS 성능 또는 16,000 IOPS 이상이 필요한 워크로드 • I/O 집약적 데이터베이스 워크로드
볼륨 크기	1GiB~64TiB	1GiB - 16TiB	4GiB~64TiB	4GiB - 16TiB
최대 IOPS	80,000 [3](25.6KiB I/O [4])	16,000(16KiB I/O[4])	256,000[3](16 KiB I/O[4])	64,000(16KiB I/O[4])
최대 처리량	2,000MiB/s	250MiB/s [1]	4,000MiB/s	1,000MiB/s [2]
Amazon EBS 다중 연결	지원되지 않음		지원됨	

	Amazon EBS 범용 SSD 볼륨	Amazon EBS 프로비저닝된 IOPS SSD 볼륨	
NVMe 예약	지원되지 않음	지원	지원됨
부트 볼륨	지원됨		

[1] 처리량 한도는 볼륨 크기에 따라 128MiB/s~250 MiB/s입니다. 자세한 내용은 gp2 볼륨 성능 섹션을 참조하세요. 2018년 12월 3일 이전에 생성되었으며 생성 이후 수정되지 않은 볼륨은 해당 볼륨을 수정하지 않는 한 전체 성능에 도달하지 못할 수 있습니다.

[2] 1,000MiB/s의 최대 처리량을 달성하려면 볼륨을 64,000 IOPS로 프로비저닝하고 Nitro기반 인스턴스에 연결해야 합니다. 2017년 12월 6일 이전에 생성되었으며 생성 이후 수정되지 않은 볼륨은 해당 볼륨을 수정하지 않는 한 전체 성능에 도달하지 못할 수 있습니다.

[3] Nitro 기반 인스턴스는 최대 256,000 IOPS로 프로비저닝된 볼륨을 지원합니다. 다른 인스턴스 유형은 최대 64,000 IOPS로 프로비저닝된 볼륨에 연결할 수 있지만 최대 32,000 IOPS를 달성할 수 있습니다.

[4] 볼륨의 처리량 한도 내에서 최대 IOPS에 도달하기 위해 필요한 I/O 크기를 나타냅니다.

[5] io2 Block Express 볼륨은 16KiB I/O 작업에 500마이크로초 미만의 평균 지연 시간을 제공하도록 설계되었습니다.

[6] Outposts에서 gp3 볼륨은 최대 16TiB의 크기, 최대 16,000 IOPS, 최대 1,000MiB/s의 처리량을 지원합니다.

SSD 기반 볼륨 유형에 대한 자세한 내용은 다음을 참조하세요.

- Amazon EBS 범용 SSD 볼륨
- Amazon EBS 프로비저닝된 IOPS SSD 볼륨

하드 디스크 드라이브(HDD) 볼륨

HDD 기반 볼륨은 기준 성능 속성이 스루풋인 대규모 스트리밍 워크로드에 최적화되어 있습니다. HDD 볼륨 유형으로는 스루풋 최적화 HDD와 콜드 HDD가 있습니다. 다음은 HDD 기반 볼륨의 사용 사례 및 특성에 대한 요약입니다.

	처리량 최적화 HDD 볼륨	콜드 HDD 볼륨
볼륨 유형	st1	sc1
내구성	99.8%~99.9% 내구성(연간 장애율 0.1%~0.2%)	
사용 사례	• 빅 데이터 • 데이터 웨어하우스 • 로그 처리	• 자주 액세스하지 않는 데이터를 위한 처리량 중심의 스토리지 • 스토리지 비용이 최대한 낮아야 하는 시나리오
볼륨 크기	125GiB ~ 16TiB	
볼륨당 최대 IOPS(1MiB I/O)	500	250
볼륨당 최대 처리량	500MiB/s	250MiB/s
Amazon EBS 다중 연결	지원되지 않음	
부트 볼륨	지원되지 않음	

하드 디스크 드라이브(HDD) 볼륨에 대한 자세한 내용은 Amazon EBS 처리량 최적화 HDD 및 콜드 HDD 볼륨 섹션을 참조하세요.

이전 세대 볼륨

마그네틱(standard) 볼륨은 마그네틱 드라이브로 지원되는 이전 세대 볼륨입니다. 데이터에 자주 액세스하지 않고 성능이 그다지 중요하지 않은 소규모 데이터 세트가 있는 워크로드에 적합합니다. Magnetic 볼륨의 평균 IOPS는 약 100 정도이며, 버스팅 시 몇백 수준으로 증가합니다. 크기는 1GiB에서 1TiB까지입니다.

> **Tip**
>
> 마그네틱은 이전 세대 볼륨 유형입니다. 이전 세대 볼륨이 제공할 수 있는 것보다 더 높은 성능 또는 성능 일관성이 필요한 경우 현재 세대의 볼륨 유형 중 하나를 사용하는 것이 좋습니다.

다음 표에서는 이전 세대 EBS 볼륨 유형을 설명합니다.

	Magnetic
볼륨 유형	`standard`
사용 사례	데이터에 자주 액세스하지 않는 워크로드
볼륨 크기	1GiB - 1TiB
볼륨당 최대 IOPS	40–200
볼륨당 최대 처리량	40-90MiB/s
부트 볼륨	지원됨

Amazon EBS 범용 SSD 볼륨

범용 SSD(gp2 및 gp3) 볼륨은 SSD(Solid-State Drive)로 지원됩니다. 다양한 트랜잭션 워크로드를 위한 가격과 성능의 균형을 유지합니다. 여기에는 가상 데스크톱, 중간 크기의 단일 인스턴스 데이터베이스, 지연 시간에 민감한 대화형 애플리케이션, 개발 및 테스트 환경, 부팅 볼륨이 포함됩니다. 대부분의 워크로드에 이 볼륨을 사용하는 것이 좋습니다.

Amazon EBS는 다음 유형의 범용 SSD 볼륨을 제공합니다.

유형

- 범용 SSD(gp3) 볼륨
- 범용 SSD(gp2) 볼륨

범용 SSD(gp3) 볼륨

범용 SSD(gp3) 볼륨은 최신 세대의 범용 SSD 볼륨이며 Amazon EBS에서 제공하는 가장 저렴한 SSD 볼륨입니다. 이 볼륨 유형은 대부분의 애플리케이션에 적절한 가격과 성능의 균형을 제공하는 데 도움이 됩니다. 또한 볼륨 크기와 관계없이 볼륨 성능을 확장하는 데 도움이 됩니다. 즉, 추가 블록 스토리지 용량을 프로비저닝할 필요 없이 필요한 성능을 프로비저닝할 수 있습니다. 또한 gp3 볼륨은 범용 SSD(gp2) 볼륨보다 GiB당 20% 저렴한 가격을 제공합니다.

gp3 볼륨은 10밀리초 미만의 지연 시간과 99.8% ~ 99.9%의 볼륨 내구성을 제공하며 연간 장애율(AFR)은 0.2% 이하입니다. 즉, 1년 동안 실행 중인 볼륨 1,000개당 최대 2개의 볼륨 장애가 발생합니다.는 gp3 볼륨을 AWS 설계하여 프로비저닝된 성능을 99%의 시간 동안 제공합니다.

> **Tip**
>
> 지연 시간에 민감한 워크로드의 경우, io2 Block Express 볼륨을 사용할 것이 권장됩니다. io2 Block Express 볼륨은 16KiB I/O 작업에 500마이크로초 미만의 평균 지연 시간을 제공하도록 설계되었습니다. io2 또한 Block Express 볼륨은 범용 볼륨에 비해 더 나은 이상 지연 시간을 제공하여 800마이크로초를 초과하는 I/O의 빈도를 10배 이상 줄여 줍니다. 자세한 내용은 프로비저닝된 IOPS SSD(io2) Block Express 볼륨 섹션을 참조하세요.

내용

- gp3 볼륨 성능
- gp3 볼륨 크기
- gp2에서 gp3로 마이그레이션

gp3 볼륨 성능

> **Tip**
>
> gp3 볼륨은 버스트 성능을 사용하지 않습니다. 전체 프로비저닝된 IOPS 및 처리량 성능을 무기한 유지할 수 있습니다.

IOPS 성능

gp3 볼륨은 스토리지 가격에 포함된 3,000 IOPS의 일관된 기본 IOPS 성능을 제공합니다. 볼륨 크기 GiB당 500 IOPS 비율의 추가 비용으로 추가 IOPS(최대 8,000)를 프로비저닝할 수 있습니다. 최대 IOPS는 160GiB 이상의 볼륨에 프로비저닝할 수 있습니다(GiB당 500 IOPS × 160GiB = 80,000 IOPS).

처리량 성능

gp3 볼륨은 스토리지 가격에 포함된 125MiB/s의 일관된 기본 처리 성능을 제공합니다. 프로비저닝된 IOPS당 0.25MiB/s 비율의 추가 비용으로 추가 처리량(최대 2,000MiB/s)을 프로비저닝할 수 있습니다. 최대 처리량은 8,000 IOPS 이상 및 16GiB 이상(8,000 IOPS × IOPS당 0.25MiB/s = 2,000MiB/s)으로 프로비저닝할 수 있습니다.

> **Note**
>
> Outposts에서 gp3 볼륨은 최대 16TiB의 크기, 최대 16,000 IOPS, 최대 1,000MiB/s의 처리량을 지원합니다.

gp3 볼륨 크기

gp3 볼륨의 크기는 1GiB~64TiB일 수 있습니다.

gp2에서 gp3로 마이그레이션

현재 gp2 볼륨을 사용 중인 경우 탄력적 볼륨 작업을 사용하여 Amazon EBS 볼륨 수정 작업을 사용하여 볼륨을 gp3로 마이그레이션할 수 있습니다. Amazon EBS Elastic Volumes 작업을 사용하여 Amazon EC2 인스턴스를 중단하지 않고 기존 볼륨의 볼륨 유형, IOPS 및 처리량을 수정할 수 있습니다. 콘솔을 사용하여 볼륨을 생성하거나 스냅샷에서 AMI를 생성할 때 범용 SSD gp3가 볼륨 유형에 대한 기본 선택 사항입니다. 다른 경우에는 gp2가 기본 선택 사항입니다. 이러한 경우에는 gp2를 사용하는 대신 볼륨 유형으로 gp3를 선택할 수 있습니다.

gp2 볼륨을 gp3로 마이그레이션하여 얼마나 절약할 수 있는지 알아보려면 Amazon EBS gp2에서 gp3로 마이그레이션 비용 절감 계산기를 사용합니다.

범용 SSD(gp2) 볼륨

광범위한 트랜잭션 워크로드에 이상적인 비용 효율적인 스토리지를 제공합니다. gp2 볼륨을 사용하면 볼륨 크기에 따라 성능이 확장됩니다.

> **Tip**
>
> gp3 볼륨은 최신 세대의 범용 SSD 볼륨입니다. gp2 볼륨보다 최대 20% 낮은 더 예측 가능한 성능 조정 및 가격을 제공합니다. 자세한 내용은 범용 SSD(gp3) 볼륨 섹션을 참조하세요. gp2 볼륨을 gp3로 마이그레이션하여 얼마나 절약할 수 있는지 알아보려면 Amazon EBS gp2에서 gp3로 마이그레이션 비용 절감 계산기를 사용합니다.

gp2 볼륨은 10밀리초 미만의 지연 시간과 99.8% ~ 99.9%의 볼륨 내구성을 제공하며 연간 장애율(AFR)은 0.2%를 넘지 않습니다. 즉, 1년 동안 실행 중인 볼륨 1,000개당 최대 2개의 볼륨 장애가 발생합니다.는 프로비저닝된 성능을 99%의 시간 동안 제공하도록 gp2 볼륨을 AWS 설계합니다.

내용

- gp2 볼륨 성능
- gp2 볼륨 크기

gp2 볼륨 성능

IOPS 성능

기준 IOPS 성능은 볼륨 크기 GiB당 3IOPS의 비율로 최소 100에서 최대 16,000 사이에서 선형으로 조정됩니다. IOPS 성능은 다음과 같이 프로비저닝됩니다.

- 33.33GiB 이하의 볼륨은 최소 100 IOPS로 프로비저닝됩니다.
- 33.33GiB보다 큰 볼륨은 최대 16,000 IOPS(5,334GiB(3 X 5,334)에서 도달)까지 볼륨 크기의 GiB당 3 IOPS로 프로비저닝됩니다.
- 5,334GiB 이상의 볼륨은 16,000 IOPS로 프로비저닝됩니다.

1TiB보다 작은 gp2 볼륨(3,000 IOPS 미만으로 프로비저닝됨)은 장기간 필요할 때 3,000 IOPS로 버스트할 수 있습니다. 볼륨의 버스트 기능은 I/O 크레딧에 의해 제어됩니다. I/O 수요가 기준 성능보다 크면 볼륨은 I/O 크레딧을 소비하여 필요한 성능 수준(최대 3,000 IOPS)으로 버스트합니다. 버스트하는 동안 I/O 크레딧은 누적되지 않으며 기준 IOPS 이상으로 사용되는 IOPS 비율로 소비됩니다(소비율 = 버스트 IOPS - 기준 IOPS) 볼륨에 누적된 I/O 크레딧이 많을수록 버스트 성능을 더 오래 유지할 수 있습니다. 다음과 같이 버스트 지속 시간을 계산할 수 있습니다.

```
                        (I/O credit balance)
Burst duration  =  -------------------------------
                  (Burst IOPS) - (Baseline IOPS)
```

I/O 수요가 기준 성능 수준 이하로 떨어지면 볼륨은 초당 볼륨 크기 GiB당 3 I/O 크레딧의 비율로 I/O 크레딧을 받기 시작합니다. 볼륨의 I/O 크레딧 발생 제한은 540만 I/O 크레딧으로, 최소 30분 동안 3,000 IOPS의 최대 버스트 성능을 유지할 수 있는 수준입니다.

> **Note**
>
> 각 볼륨은 540만 I/O 크레딧의 초기 I/O 크레딧 밸런스를 수신하여 부트 볼륨에 대한 빠른 초기 부팅 주기와 다른 애플리케이션에 대한 우수한 부트스트랩 경험을 제공합니다.

다음 표에는 예제 볼륨 크기 및 볼륨의 관련 기준 성능, 버스트 지속 시간(540만 I/O 크레딧으로 시작하는 경우) 및 빈 I/O 크레딧 잔고를 다시 채우는 데 필요한 시간이 나와 있습니다.

볼륨 크기(GiB)	기준 성능(IOPS)	3,000 IOPS(초)에서 버스트 지속 시간	빈 크레딧 밸런스를 채우는데 소요되는 시간(초)
1~33.33	100	1,862	54,000
100	300	2,000	18,000
334(최대 처리량에 대한 최소 크기)	1,002	2,703	5,389
750	2,250	7,200	2,400
1,000	3,000	해당 사항 없음*	해당 사항 없음*
5,334(최대 IOPS에 대한 최소 크기) 이상	16,000	해당 사항 없음*	해당 사항 없음*

* 볼륨의 기준 성능이 최대 버스트 성능을 초과합니다.

Amazon CloudWatch의 Amazon EBS `BurstBalance` 지표를 사용하여 볼륨에 대한 I/O 크레딧 밸런스를 모니터링할 수 있습니다. 이 지표는 gp2에 대한 I/O 크레딧의 나머지 비율을 보여줍니다. 자세한 내용은 Amazon EBS I/O 기능 및 모니터링 섹션을 참조하세요. `BurstBalance` 값이 특정 수준으로 떨어질 때를 알려주는 경보를 설정할 수 있습니다. 자세한 내용을 알아보려면 CloudWatch 경보 생성을 참조하세요.

처리량 성능

gp2 볼륨은 볼륨 크기에 따라 128~250MiB/s의 처리량을 제공합니다. 처리량 성능은 다음과 같이 프로비저닝됩니다.

- 170GiB 이하의 볼륨은 최대 128MiB/s의 처리량을 제공합니다.
- 170~334GiB의 볼륨은 최대 처리량 250MiB/s로 버스트할 수 있습니다.
- 334GiB 이상의 볼륨은 250MiB/s를 제공합니다.

gp2 볼륨의 처리량은 다음 공식을 사용하여 계산할 수 있으며 최대 처리량 제한은 250MiB/s입니다.

```
Throughput in MiB/s = IOPS performance × I/O size in KiB / 1,024
```

gp2 볼륨 크기

gp2 볼륨 크기는 1GiB~16TiB입니다. 볼륨 성능은 볼륨 크기에 따라 선형적으로 조정됩니다.

Amazon EBS 프로비저닝된 IOPS SSD 볼륨

프로비저닝된 IOPS SSD 볼륨은 SSD(Solid-State Drive)로 지원됩니다. 짧은 지연 시간이 필요한 중요하고 IOPS 집약적이며 처리량 집약적 워크로드를 위해 설계된 최고 성능의 Amazon EBS 스토리지 볼륨입니다. 프로비저닝된 IOPS SSD 볼륨은 99.9%의 시간 동안 프로비저닝된 IOPS 성능을 제공합니다.

Amazon EBS는 2가지 유형의 프로비저닝된 IOPS SSD 볼륨을 제공합니다.

- 프로비저닝된 IOPS SSD(io2) Block Express 볼륨
- 프로비저닝된 IOPS SSD(io1) 볼륨

프로비저닝된 IOPS SSD(**io2**) Block Express 볼륨

io2 Block Express는 차세대 Amazon EBS 스토리지 서버 아키텍처를 기반으로 구축되었습니다. Nitro System에 구축된 인스턴스에서 실행되는 가장 까다로운 I/O 집약적 애플리케이션의 성능 요구 사항을 충족하도록 구축되었습니다. 내구성이 가장 높고 지연 시간이 가장 Block Express는 Oracle, SAP HANA, Microsoft SQL Server 및 SAS Analytics와 같이 성능 집약적이고 미션 크리티컬한 워크로드를 실행하는 데 이상적입니다.

Block Express 아키텍처는 io2 볼륨의 성능과 규모를 개선합니다. Block Express 서버는 Scalable Reliable Datagram(SRD) 네트워킹 프로토콜을 사용하여 Nitro 기반 인스턴스와 통신합니다. 이 인터페이스는 인스턴스의 호스트 하드웨어에 있는 Amazon EBS I/O 기능 전용 Nitro Card에 구현됩니다. I/O 지연 및 지연 시간 변화(네트워크 지터)를 최소화하여 애플리케이션에 보다 빠르고 일관된 성능을 제공합니다.

io2 Block Express 볼륨은 0.001% 이하의 연간 장애율(AFR)로 99.999%의 볼륨 내구성을 제공하도록 설계되었습니다. 이는 1년 동안 실행 볼륨 10만 개당 1건의 볼륨 장애가 발생함을 의미합니다. io2 Block Express 볼륨은 밀리초 미만의 일관적인 지연 시간을 제공하고 gp3 볼륨보다 높은 IOPS 및 높은 처리량을 지원하는 단일 볼륨을 사용하는 것이 유리한 워크로드에 적합합니다.

io2 Block Express 볼륨은 Nitro 기반 인스턴스에 연결되었을 때 16KiB I/O 작업에 평균 500마이크로초 미만의 지연 시간을 제공하도록 설계되었습니다. io2 또한 Block Express 볼륨은 범용 볼륨에 비해 더 나은 이상 지연 시간을 제공하여 800마이크로초를 초과하는 I/O의 빈도를 10배 이상 줄여 줍니다.

주제

- 고려 사항
- 성능

고려 사항

- io2 Block Express 볼륨은 AWS 리전 및 중국 AWS GovCloud (US) 리전을 포함한 모든 리전에서 사용할 수 있습니다.
- 2025년 4월 30일을 기하여 신규 및 이전에 생성된 모든 io2 볼륨은 io2 Block Express 볼륨입니다.
- Nitro 기반 인스턴스에서는 최대 256,000 IOPS로 프로비저닝된 볼륨을 지원합니다. 다른 인스턴스 유형은 최대 64,000 IOPS로 프로비저닝된 볼륨에 연결할 수 있지만 최대 32,000 IOPS를 달성할 수 있습니다.

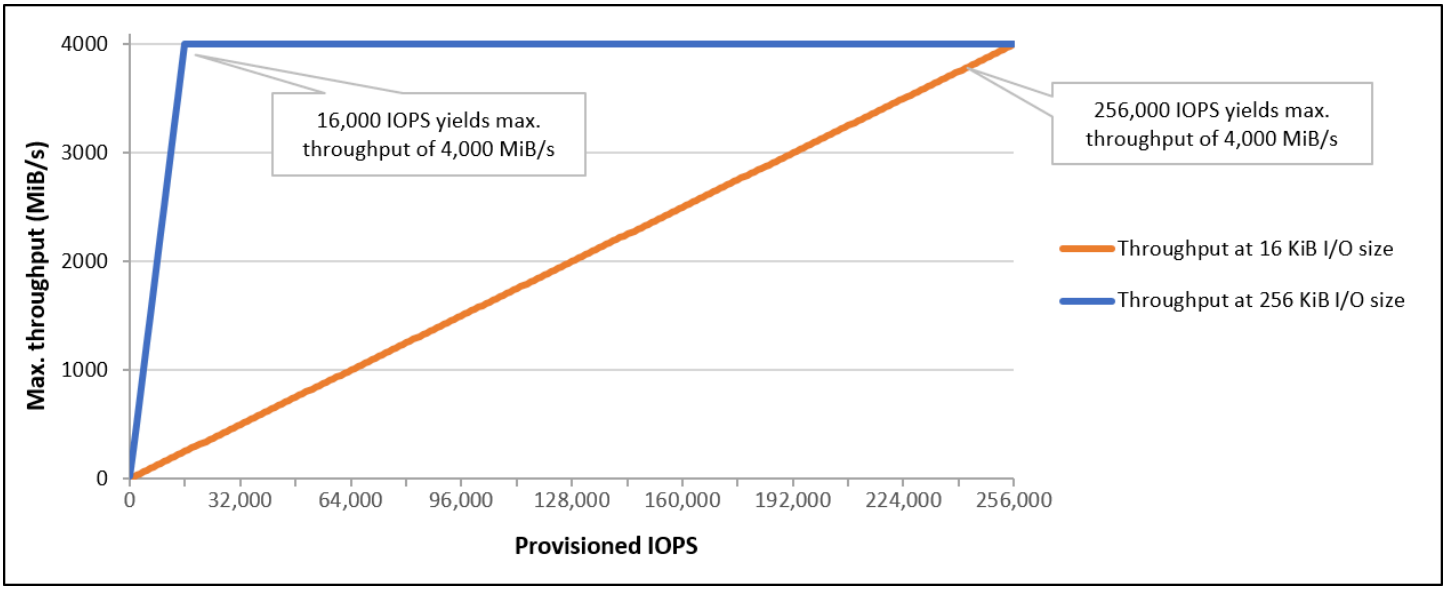
성능

io2 Block Express 볼륨은 다음과 같은 특징을 가지고 있습니다.

- 16KiB I/O 크기에 대한 500마이크로초 미만의 평균 지연 시간. 범용 볼륨에 비해 더 나은 이상 지연 시간을 제공하여 800마이크로초를 초과하는 I/O의 빈도를 10배 이상 감소.
- 최대 64TiB(65,536GiB)의 스토리지 용량.
- IOPS: GiB 비율이 1,000:1인 최대 256,000의 프로비저닝된 IOPS. 최대 IOPS는 256GiB 이상의 볼륨으로 프로비저닝될 수 있습니다(1,000 IOPS x 256GiB = 256,000 IOPS).

> **Note**
>
> Nitro 기반 인스턴스로 최대 256,000 IOPS를 달성할 수 있습니다. 다른 인스턴스에서는 최대 32,000 IOPS를 달성할 수 있습니다.

- 최대 4,000Mib/s의 볼륨 처리량. 처리량은 프로비저닝된 IOPS당 0.256Mib/s의 비율로 비례하여 확장됩니다. 최대 처리량은 16,000 IOPS 이상에서 달성할 수 있습니다.



프로비저닝된 IOPS SSD(**io1**) 볼륨

프로비저닝된 IOPS SSD(io1) 볼륨은 스토리지 성능과 일관성에 민감한 I/O 집약적 워크로드, 특히 데이터베이스 워크로드의 요구 사항을 충족하도록 설계되었습니다. 프로비저닝된 IOPS SSD 볼륨은 볼륨을 생성할 때 지정한 일관된 IOPS 속도를 사용하며 Amazon EBS는 프로비저닝된 성능의 99.9%를 제공합니다.

io1 볼륨은 0.2% 이하의 연간 장애율(AFR)로 99.8%~99.9%의 볼륨 내구성을 제공하도록 설계되었습니다. 이는 1년 동안 실행 볼륨 1,000개당 최대 2개의 볼륨 장애가 발생함을 의미합니다.

io1 볼륨은 모든 Amazon EC2 인스턴스 유형에 사용할 수 있습니다.

성능

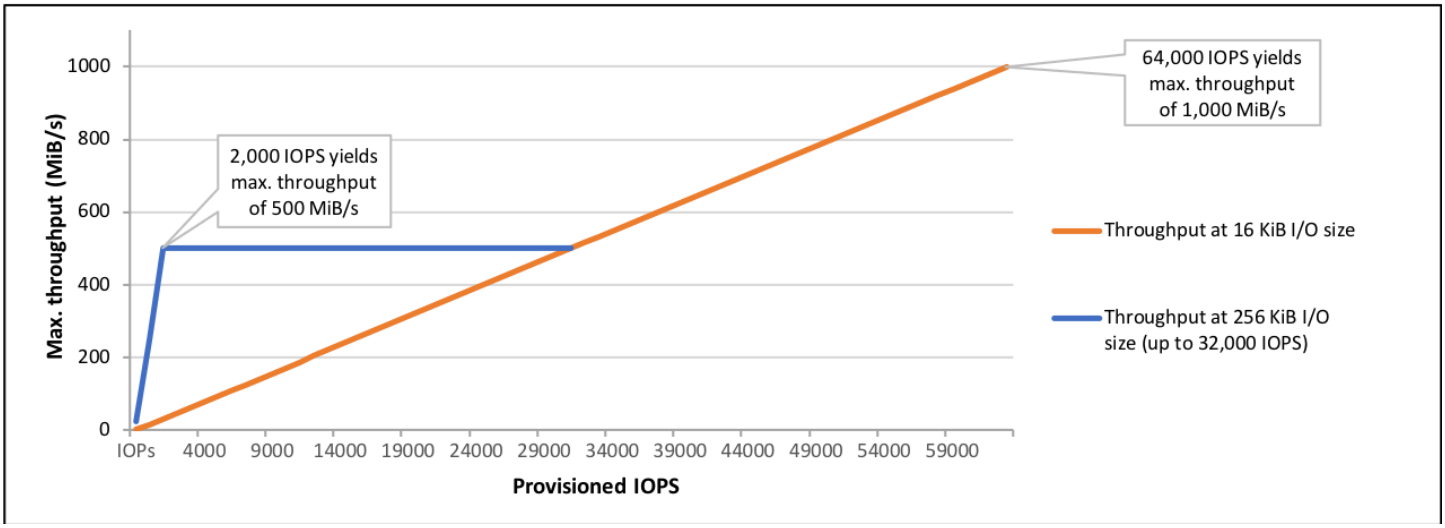
io1 볼륨의 크기는 4GiB에서 16TiB 사이가 될 수 있고 볼륨당 100 IOPS에서 최대 64,000 IOPS가 프로비저닝될 수 있습니다. 요청된 볼륨 크기(단위: GiB)에 대한 프로비저닝된 IOPS의 비율은 최대 50:1입니다. 예를 들어 100GiB io1 볼륨에서는 최대 5,000 IOPS까지 프로비저닝할 수 있습니다.

1,280GiB 이상(50 × 1,280GiB = 64,000 IOPS)의 볼륨에 대해 최대 IOPS를 프로비저닝할 수 있습니다.

- 최대 32,000 IOPS로 프로비저닝되는 io1 볼륨은 최대 256KiB의 I/O 크기를 지원하고 최대 500MiB/s의 처리량을 제공합니다. I/O 크기가 최대일 때 2,000 IOPS에서 피크 처리량에 도달합니다.
- 32,000 IOPS를 초과하여 프로비저닝된 io1 볼륨(최대 64,000 IOPS)의 처리량은 프로비저닝된 IOPS당 16KiB의 속도로 선형으로 증가합니다. 예를 들어 48,000 IOPS로 프로비저닝된 볼륨은 최대 750MiB/s의 처리량(프로비저닝된 IOPS당 16KiB x 프로비저닝된 IOPS 48,000 = 750MiB/s)을 지원할 수 있습니다.

- 1,000MiB/s의 최대 처리량을 달성하려면 64,000 IOPS(프로비저닝된 IOPS당 16KiB x 프로비저닝된 IOPS 64,000 = 1,000MiB/s)로 볼륨을 프로비저닝해야 합니다.
- Nitro 기반 인스턴스에서만 최대 64,000 IOPS를 달성할 수 있습니다. 다른 인스턴스에서는 최대 32,000 IOPS를 달성할 수 있습니다.

. 다음 그래프에 이러한 성능 특성이 예시되어 있습니다.



I/O당 지연 시간 환경은 프로비저닝된 IOPS 및 워크로드 프로파일에 따라 다릅니다. 최상의 I/O 지연 시간 환경을 위해 워크로드의 I/O 프로파일을 충족하도록 IOPS를 프로비저닝해야 합니다.

Amazon EBS 처리량 최적화 HDD 및 콜드 HDD 볼륨

Amazon EBS에서 제공하는 HDD 지원 볼륨은 다음 범주로 나뉩니다.

- 처리량 최적화 HDD - 자주 액세스하는 처리량 집약적 워크로드에 적합한 저비용 HDD입니다.
- 콜드 HDD - 자주 액세스하지 않는 워크로드용으로 설계된 가장 저렴한 HDD입니다.

주제

- 인스턴스당 처리량에 대한 제한
- 처리량 최적화 HDD 볼륨
- 콜드 HDD 볼륨
- HDD 볼륨 사용 시 성능 고려사항
- 볼륨에 대한 버스트 버킷 밸런스 모니터링

인스턴스당 처리량에 대한 제한

st1 및 sc1 볼륨의 처리량은 항상 다음 중 작은 값에 따라 결정됩니다.

- 볼륨의 처리량 제한
- 인스턴스의 처리량 제한

모든 Amazon EBS 볼륨에서와 같이, 네트워크 병목 현상을 피하려면 적절한 EBS 최적화 EC2 인스턴스를 선택하는 것이 좋습니다.

처리량 최적화 HDD 볼륨

처리량 최적화 HDD(st1) 볼륨은 IOPS가 아닌 처리량으로 성능을 정의하는 저비용 마그네틱 스토리지를 제공합니다. 이 볼륨 유형은 Amazon EMR, ETL, 데이터 웨어하우스, 로그 처리 같은 대용량 순차 워크로드에 적합합니다. 부팅 가능한 st1 볼륨은 지원되지 않습니다.

처리량 최적화 HDD(st1) 볼륨은 콜드 HDD(sc1) 볼륨과 비슷하지만 자주 액세스하는 데이터를 지원하도록 설계되었습니다.

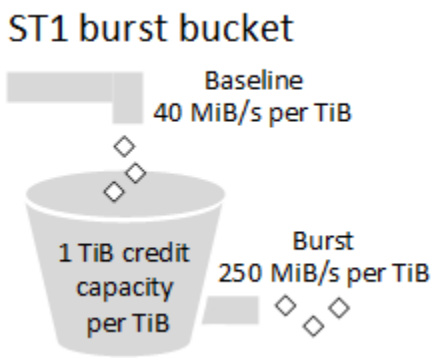
> **Note**
>
> 이 볼륨 유형은 대용량 순차 I/O와 관련된 워크로드에 최적화되어 있으며, 소량의 랜덤 I/O 워크로드를 처리하는 고객에게는 Amazon EBS 범용 SSD 볼륨 또는 Amazon EBS 프로비저닝된 IOPS SSD 볼륨 사용을 권장합니다. 자세한 내용은 HDD 기반 소량 읽기/쓰기의 비효율성 섹션을 참조하세요.

EBS 최적화 인스턴스에 연결된 처리량 최적화 HDD(st1) 볼륨은 지정된 해의 시간 중 예상 처리량 성능 99%의 90% 이상을 제공되는 일관된 성능이 제공되도록 설계되었습니다.

처리량 크레딧 및 버스트 성능

gp2처럼 st1 역시 성능 측정에 버스트 버킷 모델을 사용합니다. 볼륨 크기에 따라 볼륨의 기준 처리량, 즉 볼륨이 처리량 크레딧을 누적하는 속도가 결정됩니다. 볼륨 크기는 볼륨의 버스트 처리량, 즉 사용 가능한 크레딧을 소비할 수 있는 속도도 결정합니다. 볼륨이 클수록 기본 및 버스트 처리량이 높습니다. 볼륨에 크레딧이 많을수록 버스트 수준에서 더 오랫동안 I/O를 구동할 수 있습니다.

다음 다이어그램은 st1의 버스트 버킷 동작을 보여줍니다.



처리량 및 처리량 크레딧 한도가 적용되는 st1 볼륨의 사용 가능 처리량은 다음 수식으로 표현됩니다.

```
(Volume size) × (Credit accumulation rate per TiB) = Throughput
```

1TiB st1 볼륨의 경우 버스트 처리량은 250MiB/s로 제한되고, 버킷의 크레딧은 40MiB/s 속도로 채워지며, 최대 1TiB에 해당하는 크레딧을 보유할 수 있습니다.

최대 처리량 한도인 500MiB/s 내에서, 볼륨 크기에 비례하여 이러한 제한이 확장됩니다. 버킷이 고갈된 후 처리량은 TiB당 40MiB/s의 기준 속도로 제한됩니다.

0.125TiB~16TiB 범위의 볼륨 크기를 기준으로 기준 처리량은 5MiB/s~500MiB/s(한도)이며, 다음과 같이 12.5TiB에서 한도에 도달합니다.

```
            40 MiB/s
12.5 TiB × ---------- = 500 MiB/s
             1 TiB
```

버스트 처리량은 31MiB/s~500MiB/s(한도)이며, 다음과 같이 2TiB에서 한도에 도달합니다.

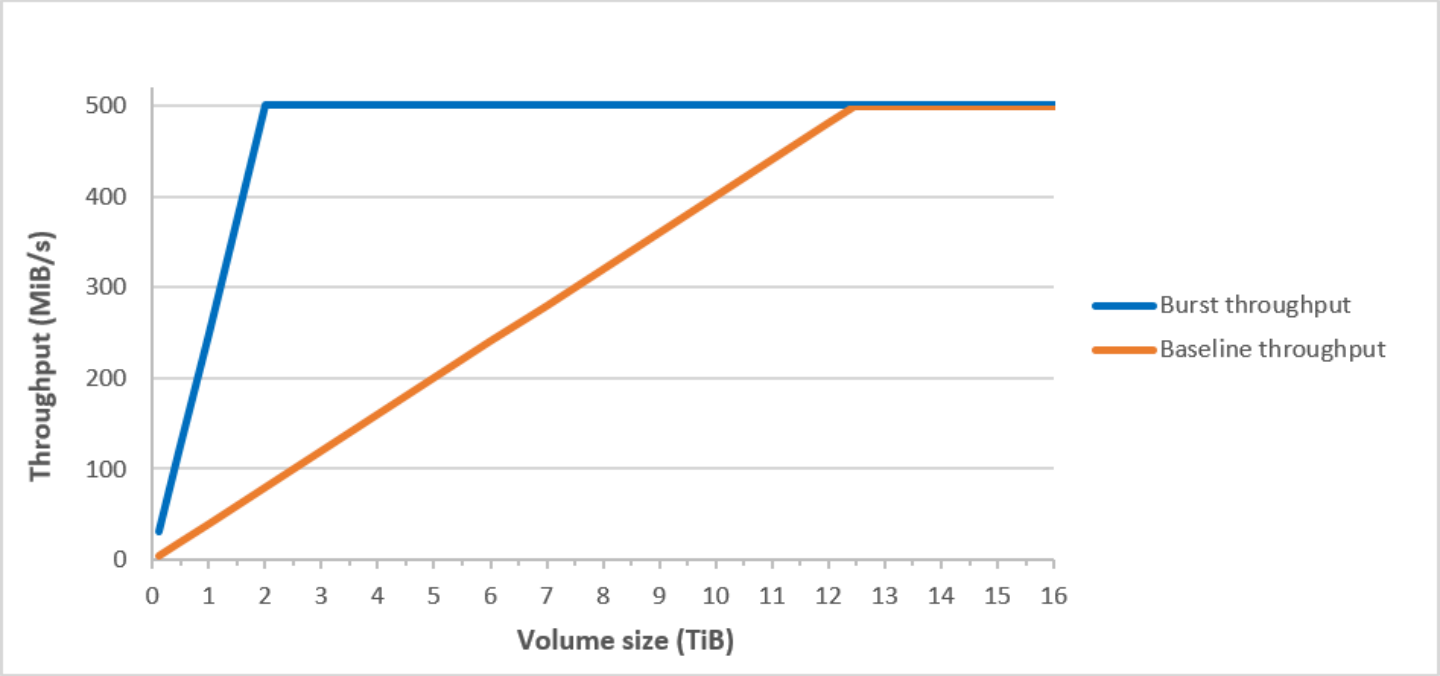
```
         250 MiB/s
2 TiB × ---------- = 500 MiB/s
          1 TiB
```

다음 표는 st1의 기준 및 버스트 처리량 값 전체를 보여줍니다.

볼륨 크기(TiB)	ST1 기준 처리량(MiB/s)	ST1 버스트 처리량(MiB/s)
0.125	5	31
0.5	20	125

볼륨 크기(TiB)	ST1 기준 처리량(MiB/s)	ST1 버스트 처리량(MiB/s)
1	40	250
2	80	500
3	120	500
4	160	500
5	200	500
6	240	500
7	280	500
8	320	500
9	360	500
10	400	500
11	440	500
12	480	500
12.5	500	500
13	500	500
14	500	500
15	500	500
16	500	500

다음 다이어그램은 표의 값을 도식화한 것입니다.



> **Note**
>
> 처리량 최적화 HDD(st1) 볼륨의 스냅샷을 생성하는 경우 스냅샷이 진행되는 동안 성능이 볼륨의 기준 값까지 떨어질 수 있습니다.

CloudWatch 지표 및 경보를 사용하여 버스트 버킷 잔고를 모니터링하는 방법은 볼륨에 대한 버스트 버킷 밸런스 모니터링 섹션을 참조하세요.

콜드 HDD 볼륨

콜드 HDD(sc1) 볼륨은 IOPS가 아닌 처리량으로 성능을 정의하는 저비용 마그네틱 스토리지를 제공합니다. 처리량 제한이 st1보다 낮은 sc1은 대용량 순차 콜드 데이터 워크로드에 적합합니다. 데이터에 자주 액세스할 필요가 없고 비용을 절약해야 한다면 저렴한 블록 스토리지로 sc1이 적합합니다. 부팅 가능한 sc1 볼륨은 지원되지 않습니다.

콜드 HDD(sc1) 볼륨은 처리량 최적화 HDD(st1) 볼륨과 비슷하지만 드물게 액세스하는 데이터를 지원하도록 설계되었습니다.

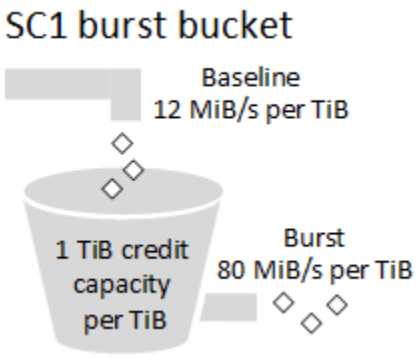
> **Note**
>
> 이 볼륨 유형은 대용량 순차 I/O와 관련된 워크로드에 최적화되어 있으며, 소량의 랜덤 I/O 워크로드를 처리하는 고객에게는 Amazon EBS 범용 SSD 볼륨 또는 Amazon EBS 프로비저닝된

IOPS SSD 볼륨 사용을 권장합니다. 자세한 내용은 HDD 기반 소량 읽기/쓰기의 비효율성 섹션을 참조하세요.

EBS 최적화 인스턴스에 연결된 콜드 HDD(sc1) 볼륨은 지정된 해의 시간 중 예상 처리량 성능 99%의 90% 이상을 제공되는 일관된 성능이 제공되도록 설계되었습니다.

처리량 크레딧 및 버스트 성능

gp2처럼 sc1 역시 성능 측정에 버스트 버킷 모델을 사용합니다. 볼륨 크기에 따라 볼륨의 기준 처리량, 즉 볼륨이 처리량 크레딧을 누적하는 속도가 결정됩니다. 볼륨 크기는 볼륨의 버스트 처리량, 즉 사용 가능한 크레딧을 소비할 수 있는 속도도 결정합니다. 볼륨이 클수록 기본 및 버스트 처리량이 높습니다. 볼륨에 크레딧이 많을수록 버스트 수준에서 더 오랫동안 I/O를 구동할 수 있습니다.



처리량 및 처리량 크레딧 한도가 적용되는 sc1 볼륨의 사용 가능 처리량은 다음 수식으로 표현됩니다.

```
(Volume size) × (Credit accumulation rate per TiB) = Throughput
```

1TiB sc1 볼륨의 경우 버스트 처리량은 80MiB/s로 제한되고, 버킷의 크레딧은 12MiB/s 속도로 채워지며, 최대 1TiB에 해당하는 크레딧을 보유할 수 있습니다.

최대 처리량 한도인 250MiB/s 내에서, 볼륨 크기에 비례하여 이러한 제한이 확장됩니다. 버킷이 고갈된 후 처리량은 TiB당 12MiB/s의 기준 속도로 제한됩니다.

0.125TiB~16TiB 범위의 볼륨 크기를 기준으로 기준 처리량은 1.5MiB/s~192MiB/s(최대)이며, 다음과 같이 16TiB에서 최대값에 도달합니다.

```
            12 MiB/s
16 TiB × ---------- = 192 MiB/s
             1 TiB
```

버스트 처리량은 10MiB/s~250MiB/s(한도)이며, 다음과 같이 3.125TiB에서 한도에 도달합니다.

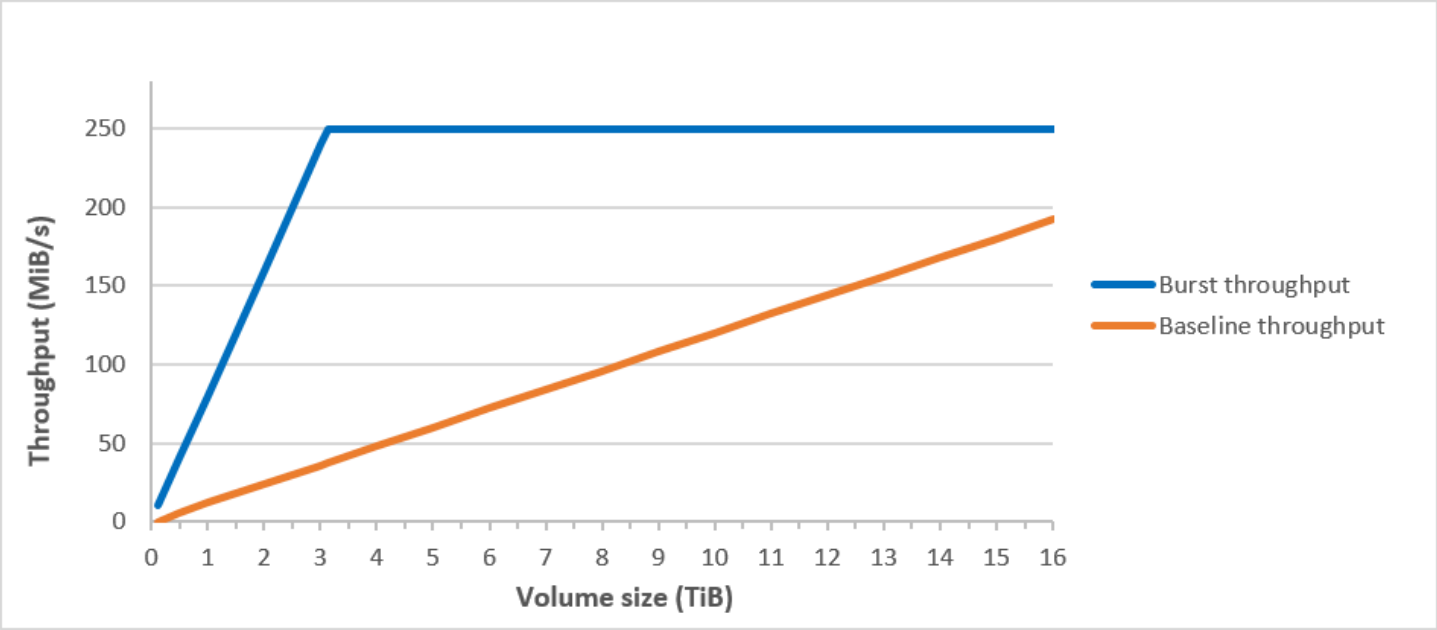
```
              80 MiB/s
3.125 TiB × ----------- = 250 MiB/s
               1 TiB
```

다음 표는 sc1의 기준 및 버스트 처리량 값 전체를 보여줍니다.

볼륨 크기(TiB)	SC1 기준 처리량(MiB/s)	SC1 버스트 처리량(MiB/s)
0.125	1.5	10
0.5	6	40
1	12	80
2	24	160
3	36	240
3.125	37.5	250
4	48	250
5	60	250
6	72	250
7	84	250
8	96	250
9	108	250
10	120	250
11	132	250
12	144	250
13	156	250

볼륨 크기(TiB)	SC1 기준 처리량(MiB/s)	SC1 버스트 처리량(MiB/s)
14	168	250
15	180	250
16	192	250

다음 다이어그램은 표의 값을 도식화한 것입니다.



> **Note**
>
> 콜드 HDD(sc1) 볼륨의 스냅샷을 생성하는 경우 스냅샷이 진행되는 동안 성능이 볼륨의 기준 값까지 떨어질 수 있습니다.

CloudWatch 지표 및 경보를 사용하여 버스트 버킷 잔고를 모니터링하는 방법은 볼륨에 대한 버스트 버킷 밸런스 모니터링 섹션을 참조하세요.

HDD 볼륨 사용 시 성능 고려사항

HDD 볼륨 사용 시 최적의 처리량을 달성하려면 다음 사항을 염두에 두고 워크로드를 계획하세요.

처리량 최적화 HDD와 콜드 HDD 비교

st1 및 sc1의 버킷 크기는 볼륨 크기에 따라 다르며, 최대 버킷에는 최대 볼륨 스캔에 충분한 토큰이 포함되어 있습니다. 그러나 st1 및 sc1 볼륨이 더 큰 경우 인스턴스당, 볼륨당 처리량 제한 때문에 볼륨 스캔을 완료하는 시간이 더 오래 걸립니다. 작은 인스턴스에 연결된 볼륨은 st1 또는 sc1 처리량이 아닌 인스턴스당 처리량에 따라 제한됩니다.

st1 및 sc1은 모두 99%의 기간 동안 90%의 버스트 처리량에 성능 일관성을 제공하도록 설계되었습니다. 매 시간 총 처리량 목표 99%를 달성하기 위해, 준수하지 않는 기간은 대략적으로 균등하게 분산됩니다.

일반적으로 스캔 시간은 이 수식으로 표현됩니다.

```
 Volume size
------------ = Scan time
 Throughput
```

예를 들어 성능 일관성 보장과 기타 최적화를 고려할 때, 5TiB 볼륨을 사용 중인 st1 고객이 전체 볼륨 스캔을 완료하는 데 걸리는 시간은 2.91~3.27시간으로 예상할 수 있습니다.

- 최적 스캔 시간

```
   5 TiB             5 TiB
----------- = ------------------ = 10,486 seconds = 2.91 hours
 500 MiB/s     0.00047684 TiB/s
```

- 최대 스캔 시간

```
  2.91 hours
-------------- = 3.27 hours
 (0.90)(0.99) <-- From expected performance of 90% of burst 99% of the time
```

마찬가지로, 5TiB 볼륨을 사용 중인 sc1 고객이 전체 볼륨 스캔을 완료하는 데 걸리는 시간은 5.83~6.54시간으로 예상됩니다.

- 최적 스캔 시간

```
   5 TiB             5 TiB
----------- = ------------------ = 20972 seconds = 5.83 hours
```

```
 250 MiB/s     0.000238418 TiB/s
```

- 최대 스캔 시간

```
  5.83 hours
-------------- = 6.54 hours
 (0.90)(0.99)
```

다음 표는 최대 버킷과 충분한 인스턴스 처리량을 가정할 때 다양한 크기의 볼륨에 이상적인 스캔 시간을 보여줍니다.

볼륨 크기(TiB)	버스팅 시 ST1 스캔 시간(단위: 시간)*	버스팅 시 SC1 스캔 시간(단위: 시간)*
1	1.17	3.64
2	1.17	3.64
3	1.75	3.64
4	2.33	4.66
5	2.91	5.83
6	3.50	6.99
7	4.08	8.16
8	4.66	9.32
9	5.24	10.49
10	5.83	11.65
11	6.41	12.82
12	6.99	13.98
13	7.57	15.15

볼륨 크기(TiB)	버스팅 시 ST1 스캔 시간(단위: 시간)*	버스팅 시 SC1 스캔 시간(단위: 시간)*
14	8.16	16.31
15	8.74	17.48
16	9.32	18.64

* 이 스캔 시간은 1MiB의 순차 I/O를 수행할 때 4 이상의 평균 대기열 깊이(가장 가까운 정수로 반올림)를 가정합니다.

따라서 빠르게 스캔을 완료해야 하거나(최대 500MiB/s) 하루 안에 여러 건의 전체 볼륨 스캔이 필요한 처리량 중심의 워크로드를 가지고 있는 경우 `st1`을 사용하십시오. 비용을 최적화해야 하고 데이터에 그다지 자주 액세스하지 않으며 250MiB/s 이상의 스캔 성능이 필요하지 않다면 `sc1`을 사용하십시오.

HDD 기반 소량 읽기/쓰기의 비효율성

`st1` 및 `sc1` 볼륨의 성능 모델은 순차 I/O, 높은 처리량의 워크로드 사용, 혼합 IOPS 및 처리량의 워크로드에 허용되는 성능 제공, 소용량 랜덤 I/O 회피에 최적화되어 있습니다.

예를 들어 1MiB 이하의 I/O 요청은 1MiB I/O 크레딧으로 간주됩니다. 그러나 순차 I/O는 1MiB I/O 블록으로 병합되고 1MiB I/O 크레딧으로 간주됩니다.

볼륨에 대한 버스트 버킷 밸런스 모니터링

`st1` 및 `sc1` 볼륨에 대해 Amazon CloudWatch에서 제공하는 Amazon EBS `BurstBalance` 지표를 사용하여 버스트 버킷 수준을 모니터링할 수 있습니다. 이 지표는 버스트 버킷에 남아 있는 `st1` 및 `sc1`의 처리량 크레딧을 보여줍니다. `BurstBalance` 지표 및 I/O 관련 기타 지표에 대한 자세한 내용은 Amazon EBS I/O 기능 및 모니터링 섹션을 참조하세요. CloudWatch를 사용하면 `BurstBalance` 값이 특정 수준으로 떨어질 때를 알려주는 경보를 설정할 수도 있습니다. 자세한 내용은 CloudWatch 경보 생성을 참조하세요.

Amazon EBS 볼륨 제약 조건

Amazon EBS 볼륨의 크기는 블록 데이터 스토리지의 물리 및 산술과 운영 체제(OS) 및 파일 시스템 디자이너의 구현 결정에 의해 제한됩니다.는 서비스의 신뢰성을 보호하기 위해 볼륨 크기에 대한 추가 제한을 AWS 부과합니다.

다음 섹션에서는 EBS 볼륨의 사용 가능한 크기를 제한하고 EBS 볼륨을 구성하기 위한 권장 사항을 제공하는 가장 중요한 요소에 대해 설명합니다.

목차

- 스토리지 용량
- 서비스 제한
- 파티셔닝 체계
- 데이터 블록 크기

스토리지 용량

다음 표에는 4,096바이트 블록 크기를 가정할 때 Amazon EBS 가장 일반적으로 사용되는 파일 시스템의 이론적 스토리지 용량과 구현된 스토리지 용량이 요약되어 있습니다.

파티셔닝 체계	최대 주소 지정 가능한 블록	이론적 최대 크기(블록 x 블록 크기)	Ext4에서 구현되는 최대 크기*	XFS에서 구현되는 최대 크기**	NTFS에서 구현되는 최대 크기	EBS에서 지원되는 최대 크기
MBR	2^{32}	2TiB	2TiB	2TiB	2TiB	2TiB
GPT	2^{64}	64ZiB	1EiB = $1{,}024^2$ TiB (RHEL7에서 인증된 50TiB)	500TiB (RHEL7에서 인증됨)	256TiB	64TiB †

* Ext4 Howto 및 Red Hat Enterprise Linux의 파일 및 시스템 크기 제한은 무엇입니까?

** Red Hat Enterprise Linux의 파일 및 시스템 크기 제한은 무엇입니까?

† io2 Block Express 볼륨은 GPT 파티션에 대해 최대 64TiB를 지원합니다. 자세한 내용은 프로비저닝된 IOPS SSD(io2) Block Express 볼륨 섹션을 참조하세요.

서비스 제한

Amazon EBS는 데이터 센터에서 대량으로 분산되는 스토리지를 가상 하드 디스크 드라이브로 추상화합니다. EC2 인스턴스에 설치된 운영 체제에서 연결된 EBS 볼륨은 512바이트 디스크 섹터가 포함된 물리적 하드 디스크 드라이브로 나타납니다. OS는 스토리지 관리 유틸리티를 통해 해당 가상 섹터에 데이터 블록(또는 클러스터)을 할당하는 작업을 관리합니다. 할당은 마스터 부트 레코드(MBR) 또는 GUID 파티션 테이블(GPT)과 같은 볼륨 파티셔닝 체계에 따라 수행되며 설치된 파일 시스템(ext4, NTFS 등)의 기능 내에서 수행됩니다.

EBS는 가상 디스크 섹터에 포함된 데이터를 인식하지 않으며, 섹터의 무결성을 보장할 뿐입니다. 즉 AWS , 작업과 OS 작업은 서로 독립적입니다. 볼륨 크기를 선택할 때는 다음과 같은 경우의 기능과 한계를 알아 두세요.

- EBS는 현재 64TiB의 최대 볼륨 크기를 지원합니다. 즉, EBS 볼륨의 크기를 64TiB까지 0000들 수 있지만, OS가 해당 용량을 모두 인식하는지 여부는 자체적인 설계 특성 및 볼륨 파티셔닝 방법에 따라 결정됩니다.
- 부트 볼륨은 MBR 또는 GPT 파티셔닝 체계를 사용해야 합니다. 인스턴스를 시작하는 AMI에 따라 부트 모드와 부트 볼륨에 사용하는 파티션 체계가 차례로 결정됩니다.

  MBR에서는 부트 볼륨의 크기가 2TiB로 제한됩니다.

  GPT에서는 GRUB2(Linux) 또는 UEFI 부트 모드(Windows)와 함께 사용하면 부트 볼륨의 크기가 64TiB까지 될 수 있습니다.

  자세한 내용은 Amazon EBS 볼륨을 사용할 수 있도록 만들기 섹션을 참조하세요.
- 2TiB(2,048GiB) 이상의 Windows 비부트 볼륨에서 전체 볼륨에 액세스하려면 GPT 파티션 테이블을 사용해야 합니다.

파티셔닝 체계

다른 영향 중에서도 특히, 파티셔닝 체계는 단일 볼륨에서 여러 논리적 데이터 블록을 고유하게 주소 지정할 수 있는 방법을 결정합니다. 자세한 내용은 데이터 블록 크기 섹션을 참조하세요. 사용 중인 일반적인 파티셔닝 체계는 마스터 부트 레코드(MBR)와 GUID 파티션 테이블(GPT)입니다. 이러한 체계 간의 중요한 차이점은 다음과 같이 요약할 수 있습니다.

MBR

MBR은 32비트 데이터 구조를 사용하여 블록 주소를 저장합니다. 따라서 각 데이터 블록은 2^{32}개의 가능한 정수 중 하나와 매핑됩니다. 주소 지정 가능한 최대 볼륨 크기는 다음 공식에 의해 지정됩니다.

$2^{32} \times$ Block size

MBR 볼륨의 블록 크기는 관례적으로 512바이트로 제한됩니다. 따라서:

$2^{32} \times 512$ bytes = 2 TiB

MBR 볼륨에 대한 이 2TiB 제한을 증가시키기 위한 엔지니어링 해결 방법은 업계에서 광범위하게 채택되는 방식과 일치하지 않습니다. 따라서 Linux와 Windows는에서 크기가 더 큰 것으로 AWS 표시되더라도 MBR 볼륨이 2TiB보다 큰 것으로 감지하지 않습니다.

GPT

GPT는 64비트 데이터 구조를 사용하여 블록 주소를 저장합니다. 따라서 각 데이터 블록은 2^{64}개의 가능한 정수 중 하나와 매핑됩니다. 주소 지정 가능한 최대 볼륨 크기는 다음 공식에 의해 지정됩니다.

$2^{64} \times$ Block size

GPT 볼륨의 블록 크기는 일반적으로 4,096바이트입니다. 따라서:

$$
\begin{aligned}
&2^{64} \times 4{,}096 \text{ bytes} \\
&\quad = 2^{64} \times 2^{12} \text{ bytes} \\
&\quad = 2^{70} \times 2^{6} \text{ bytes} \\
&\quad = 64 \text{ ZiB}
\end{aligned}
$$

실제 컴퓨터 시스템은 이러한 이론적 최대 크기와 비슷한 크기를 지원하지 않습니다. 구현되는 파일 시스템 크기는 현재 ext4의 경우 50TiB, NTFS의 경우 256TiB로 제한됩니다.

데이터 블록 크기

최신 하드 드라이브의 데이터 스토리지는 논리적 블록 주소 지정을 통해 관리됩니다. 논리적 블록 주소 지정은 운영 체제가 기본 하드웨어에 대한 많은 지식 없이 논리적 블록에서 데이터를 읽고 쓸 수 있도록 하는 추상적 계층입니다. 운영 체제는 스토리지 디바이스를 사용하여 블록을 물리적 섹터에 매핑하고 섹터 크기의 배수인 데이터 블록을 사용하여 디스크에서 데이터를 읽고 씁니다.

Amazon EBS는 다음 요소에 따라 운영 체제에 512바이트 또는 4,096바이트(4KiB) 물리적 섹터를 광고합니다.

1. Amazon EC2 인스턴스 유형
2. 운영 체제

3. NVMe 드라이버 버전

Amazon EBS는 모든 요소가 지원하는 경우에만 4-KiB 물리적 섹터를 광고합니다. 이 중 하나라도 4-KiB 물리적 섹터를 지원하지 않는 경우, Amazon EBS는 512바이트 물리적 섹터를 광고합니다.

Amazon EC2 인스턴스 유형 지원

다음 표는 Amazon EBS가 다양한 Amazon EC2 인스턴스 유형에 대해 광고하는 섹터 크기를 보여줍니다.

인스턴스 유형	Linux	Windows
모든 Xen 기반 인스턴스 유형	Amazon EBS에서는 항상 512바이트 물리적 섹터를 광고합니다.	
A1 | C5 | C5a | C5ad | C5d | C5n | C6g | C6gd | DL1 | D3 | D3en | G4ad | G4dn | G5 | G5g | I3 | I3en | Inf1 | M5 | M5a | M5ad | M5d | M5dn | M5n | M5zn | M6g | M6gd | P3dn | P4d | P4de | R5 | R5a | R5ad | R5d | R5dn | R5n | R6g | R6gd | T3 | T3a | T4g | U-12tb1 | U-18tb1 | U-24tb1 | U-3tb1 | U-6tb1 | U-9tb1 | X2gd | X2iezn | VT1 | Z1d	Amazon EBS에서는 항상 512바이트 물리적 섹터를 광고합니다.	Amazon EBS는 512바이트 또는 4-KiB 물리적 섹터를 광고합니다. [1]
기타 모든 Nitro 기반 인스턴스	Amazon EBS는 512바이트 또는 4-KiB 물리적 섹터를 광고합니다. [1]	

[1] 운영 체제 지원에 따라 달라집니다. 다음 단원을 참조하세요.

운영 체제 지원

다음 표는 운영 체제 예제와 Amazon EBS에서 광고하는 해당 물리적 섹터 크기를 제공합니다. 이 목록은 전체 목록이 아닙니다. 운영 체제에서 Amazon EBS가 광고한 물리적 섹터 크기를 확인하는 것이 좋습니다.

운영 체제	광고된 물리적 섹터 크기
• 커널 버전 4.14 이하가 포함된 Amazon Linux • RHEL 7.9 이하 • Ubuntu 20.04 이하 • Windows 7/Windows Server 2,008 이하	512B
• 커널 버전 5.3 및 이전을 사용하는 Amazon Linux • RHEL8.8 이상 • Ubuntu 22.04 이상 • Windows 8/Windows Server 2,012 이상 [1]	4KiB

[1] Windows 워크로드의 경우 최신 버전의 AWS NVMe 드라이버를 사용하고 있는지 확인합니다. Amazon EBS는 AWS NVMe 드라이버 버전 1.4.1 이상을 사용하여 44-KiB 물리적 섹터를 광고합니다.

비 기본 블록 크기

논리 데이터 블록의 업계 기본 크기는 현재 4KiB입니다. 특정 워크로드는 더 작거나 더 큰 블록 크기에서 이점을 얻을 수 있기 때문에 파일 시스템은 포맷 중 지정할 수 없는 비 기본 블록 크기를 지원합니다. 비 기본 블록 크기를 사용해야 하는 시나리오(예: 최적화)는 이 주제의 범위를 벗어나지만, 블록 크기 선택은 볼륨의 스토리지 용량에 영향을 미칩니다. 다음 표는 이론적 스토리지 용량을 블록 크기의 함수로 보여줍니다. 그러나 현재 EBS에서 부과하는 볼륨 크기(io2 Block Express의 경우 64TiB)에 대한 제한은 16KiB 데이터 블록에서 지원되는 최대 크기와 동일하다는 점에 유의하세요.

블록 크기	최대 볼륨 크기
4KiB(기본값)	16TiB
8KiB	32TiB
16KiB	64TiB
32KiB	128TiB
64KiB(최대)	256TiB

Amazon EBS 볼륨 및 NVMe

AWS Nitro 시스템에 구축된 Amazon EC2 인스턴스에서는 Amazon EBS 볼륨이 NVMe 블록 디바이스로 표시됩니다. NVMe 블록 디바이스로 노출된 Amazon EBS 볼륨의 성능과 기능을 완전히 활용하려면 EC2 인스턴스에 AWS NVMe 드라이버가 설치되어 있어야 합니다. 모든 현재 세대 AWS Windows 및 Linux AMIs NVMe 드라이버가 AWS 기본적으로 설치되어 있습니다.

AWS NVMe 드라이버가 없는 AMI를 사용하는 경우 수동으로 설치할 수 있습니다. 자세한 내용은 Amazon EC2 사용 설명서의 AWS NVMe 드라이버를 참조하세요.

Linux 인스턴스

디바이스 이름은 `/dev/nvme0n1`, `/dev/nvme1n1` 등입니다. 블록 디바이스 매핑에서 사용자가 지정하는 디바이스 이름은 NVMe 디바이스 이름(`/dev/nvme[0-26]n1`)을 이용해 바뀝니다. 블록 디바이스 드라이버는 블록 디바이스 매핑에서 볼륨에 대해 지정한 순서와는 다른 순서로 NVMe 디바이스 이름을 할당할 수 있습니다.

Windows 인스턴스

볼륨을 인스턴스에 연결할 때 해당 볼륨에 대한 디바이스 이름을 포함합니다. 이 디바이스 이름은 Amazon EC2에서 사용합니다. 인스턴스의 블록 디바이스 드라이버는 볼륨이 마운트될 때 실제 볼륨 이름을 할당합니다. 할당된 이름은 Amazon EC2에서 사용하는 이름과 다를 수 있습니다.

내용

- Amazon EBS 볼륨을 NVMe 디바이스 이름에 매핑
- Amazon EBS 볼륨에 대한 NVMe I/O 작업 시간 제한
- Amazon EBS 볼륨에 대한 NVMe Abort 명령

Amazon EBS 볼륨을 NVMe 디바이스 이름에 매핑

EBS는 단일 루트 I/O 가상화(SR-IOV)를 사용하여 NVMe 사양을 사용하는 Nitro 기반 인스턴스에서 볼륨 연결을 제공합니다. 이러한 디바이스는 운영 체제의 표준 NVMe 드라이버에 의존합니다. 이러한 드라이버는 일반적으로 인스턴스 부팅 중에 연결된 디바이스를 검색하고, 블록 디바이스 매핑에서 디바이스가 지정되는 방식이 아닌 디바이스가 응답하는 순서에 따라 디바이스 노드를 생성합니다.

Linux 인스턴스

Linux에서 NVMe 디바이스 이름은 `/dev/nvme<x>n<y>` 패턴을 따릅니다. `<x>`는 열거 순서이고, EBS의 경우 `<y>`는 1입니다. 경우에 따라 디바이스는 후속 인스턴스가 시작되는 것과 다른 순서로 검색에

응답하기도 하는데, 이로 인해 디바이스 이름이 변경됩니다. 또한 블록 디바이스 드라이버에 의해 할당된 디바이스 이름은 블록 디바이스 매핑에 지정된 이름과 다를 수 있습니다.

인스턴스 내 EBS 볼륨에 대해 다음과 같은 안정된 식별자를 사용하는 것이 좋습니다.

- Nitro 기반 인스턴스의 경우, EBS 볼륨을 연결하거나 `AttachVolume` 또는 `RunInstances` API 호출이 NVMe 컨트롤러 식별의 벤더별 데이터 필드에서 캡처되는 동안 Amazon EC2 콘솔에 지정된 블록 디바이스 매핑. 2017.09.01 버전 이후의 Amazon Linux AMI를 사용하는 경우, 이 데이터를 읽고 블록 디바이스 매핑의 심볼 링크를 생성하는 udev 규칙이 제공됩니다.
- EBS 볼륨 ID와 탑재 지점은 인스턴스 상태 변경 간에 유지됩니다. NVMe 디바이스 이름은 인스턴스 부팅 중 디바이스가 응답하는 순서에 따라 변경될 수 있습니다. 일관된 디바이스 식별을 위해 EBS 볼륨 ID와 탑재 지점을 사용하는 것이 좋습니다.
- NVMe EBS 볼륨에는 디바이스 식별에서 일련 번호로 설정된 EBS 볼륨 ID가 있습니다. `lsblk -o +SERIAL` 명령을 사용하여 일련 번호를 나열합니다.
- NVMe 디바이스 이름 형식은 EBS 볼륨이 인스턴스 시작 도중에 연결되었는지 아니면 인스턴스 시작 후에 연결되었는지에 따라 달라질 수 있습니다. 인스턴스 시작 후 연결된 볼륨의 NVMe 디바이스 이름에는 `/dev/` 접두사가 포함되지만, 인스턴스 시작 중에 연결된 볼륨의 NVMe 디바이스 이름에는 `/dev/` 접두사가 포함되지 않습니다.
  - Amazon Linux 또는 FreeBSD AMI의 경우, NVMe 디바이스 이름의 일관성을 유지하기 위해 `sudo ebsnvme-id /dev/nvme0n1 -u` 명령을 사용합니다.
  - 다른 배포판의 경우 `sudo nvme id-ctrl -V /dev/nvme0n1` 명령을 사용하여 NVMe 디바이스 이름을 확인합니다. `--vendor-specific` 명령 옵션을 포함해야 할 수 있습니다.
- 디바이스를 포맷할 때 파일 시스템 수명 기간 동안 지속되는 UUID가 생성됩니다. 이와 동시에 디바이스 레이블을 지정할 수 있습니다. 자세한 내용은 Amazon EBS 볼륨을 사용할 수 있도록 만들기 및 잘못된 볼륨에서 부팅을 참조하세요.

Amazon Linux AMI

Amazon Linux AMI 2017.09.01 이상(Amazon Linux 2 포함)에서는 다음과 같이 ebsnvme-id 명령을 실행하여 NVMe 디바이스 이름을 볼륨 ID와 디바이스 이름에 매핑할 수 있습니다.

다음 예는 인스턴스 시작 중에 연결된 볼륨에 대한 명령과 출력을 보여줍니다. NVMe 디바이스 이름에 `/dev/` 접두사가 포함되지 않은 것을 알 수 있습니다.

```
[ec2-user ~]$ sudo /sbin/ebsnvme-id /dev/nvme0n1
Volume ID: vol-01324f611e2463981
sda
```

다음 예는 인스턴스 시작 후에 연결된 볼륨에 대한 명령과 출력을 보여줍니다. NVMe 디바이스 이름에 /dev/ 접두사가 포함된 것을 알 수 있습니다.

```
[ec2-user ~]$ sudo /sbin/ebsnvme-id /dev/nvme1n1
Volume ID: vol-064784f1011136656
/dev/sdf
```

또한 Amazon Linux는 블록 디바이스 매핑의 디바이스 이름에서 NVMe 디바이스 이름으로 심볼 링크를 만듭니다(예: /dev/sdf).

FreeBSD AMI

FreeBSD 12.2-RELEASE부터 위와 같이 ebsnvme-id 명령을 실행할 수 있습니다. NVMe 디바이스의 이름 (예: nvme0)이나 디스크 디바이스(예: nvd0 또는 nda0)의 이름을 전달합니다. FreeBSD는 디스크 디바이스에 대한 심볼 링크(예: /dev/aws/disk/ebs/*volume_id*)도 생성합니다.

기타 Linux AMI

커널 버전 4.2 이상에서는 다음과 같이 nvme id-ctrl 명령을 실행하여 NVMe 디바이스 이름을 볼륨 ID에 매핑할 수 있습니다. 먼저 Linux 배포용 패키지 관리 도구를 사용하여 NVMe 명령줄 패키지 nvme-cli를 설치합니다. 다른 배포의 다운로드 및 설치 지침은 배포 관련 설명서를 참조하세요.

다음 예에서는 인스턴스 시작 중에 연결된 볼륨의 볼륨 ID와 NVMe 디바이스 이름을 가져옵니다. NVMe 디바이스 이름에 /dev/ 접두사가 포함되지 않은 것을 알 수 있습니다. NVMe 컨트롤러 벤더별 확장자(컨트롤러 식별의 바이트 384:4095)를 통해 디바이스 이름을 구할 수 있습니다.

```
[ec2-user ~]$ sudo nvme id-ctrl -V /dev/nvme0n1
NVME Identify Controller:
vid     : 0x1d0f
ssvid   : 0x1d0f
sn      : vol01234567890abcdef
mn      : Amazon Elastic Block Store
...
0000: 2f 64 65 76 2f 73 64 6a 20 20 20 20 20 20 20 20 "sda..."
```

다음 예에서는 인스턴스 시작 후에 연결된 볼륨의 볼륨 ID와 NVMe 디바이스 이름을 가져옵니다. NVMe 디바이스 이름에 /dev/ 접두사가 포함된 것을 알 수 있습니다.

```
[ec2-user ~]$ sudo nvme id-ctrl -V /dev/nvme1n1
NVME Identify Controller:
vid     : 0x1d0f
```

```
ssvid    : 0x1d0f
sn       : volabcdef01234567890
mn       : Amazon Elastic Block Store
...
0000: 2f 64 65 76 2f 73 64 6a 20 20 20 20 20 20 20 20 "/dev/sdf..."
```

lsblk 명령은 사용 가능한 디바이스와 각각의 마운트 지점(해당되는 경우)을 나열합니다. 그러면 사용할 올바른 디바이스 이름을 판단할 수 있습니다. 이 예에서 /dev/nvme0n1p1은 루트 디바이스에 마운트되어 있고 /dev/nvme1n1은 연결되었지만 마운트되어 있지 않습니다.

```
[ec2-user ~]$ lsblk
NAME          MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
nvme1n1       259:3   0  100G  0 disk
nvme0n1       259:0   0    8G  0 disk
  nvme0n1p1   259:1   0    8G  0 part /
  nvme0n1p128 259:2   0    1M  0 part
```

Windows 인스턴스

ebsnvme-id 명령을 실행하여 NVMe 디바이스 디스크 번호를 EBS 볼륨 ID와 디바이스 이름에 매핑할 수 있습니다. 기본적으로 모든 EBS NVMe 디바이스가 열거되어 있습니다. 디스크 번호를 특정 디바이스의 열거 번호로 전달할 수 있습니다. ebsnvme-id 도구는에 있는 최신 AWS 제공 Windows Server AMIs에 포함되어 있습니다C:\ProgramData\Amazon\Tools.

AWS NVMe 드라이버 패키지부터 1.5.0, 최신 버전의 ebsnvme-id 도구가 드라이버 패키지에 의해 설치됩니다. 최신 버전은 드라이버 패키지에서만 사용할 수 있습니다. ebsnvme-id 도구의 독립 실행형 다운로드 링크에서는 업데이트를 더는 받지 않습니다. 독립 실행형 링크를 통해 사용할 수 있는 마지막 버전은 ebsnvme-id.zip 링크를 사용하여 다운로드하고 ebsnvme-id.exe 액세스 권한이 있는 Amazon EC2 인스턴스에 내용을 추출할 수 있는 1.1.0입니다.

```
PS C:\ProgramData\Amazon\Tools> ebsnvme-id.exe
Disk Number: 0
Volume ID: vol-0d6d7ee9f6e471a7f
Device Name: sda1

Disk Number: 1
Volume ID: vol-03a26248ff39b57cf
Device Name: xvdd

Disk Number: 2
Volume ID: vol-038bd1c629aa125e6
```

```
Device Name: xvde

Disk Number: 3
Volume ID: vol-034f9d29ec0b64c89
Device Name: xvdb

Disk Number: 4
Volume ID: vol-03e2dbe464b66f0a1
Device Name: xvdc
```

```
PS C:\ProgramData\Amazon\Tools> ebsnvme-id.exe 4
Disk Number: 4
Volume ID: vol-03e2dbe464b66f0a1
Device Name: xvdc
```

Amazon EBS 볼륨에 대한 NVMe I/O 작업 시간 제한

대부분의 운영 체제는 I/O 작업이 NVMe 디바이스에 제출되는 시간에 제한을 두고 있습니다.

Linux 인스턴스

Linux에서 Nitro 기반 인스턴스에 연결된 EBS 볼륨은 운영 체제에서 제공되는 기본 NVMe 드라이버를 사용합니다. 대부분의 운영 체제는 I/O 작업이 NVMe 디바이스에 제출되는 시간에 제한을 두고 있습니다. 기본 제한 시간은 30초이며 `nvme_core.io_timeout` 부트 파라미터를 이용해 제한 시간을 변경할 수 있습니다. Linux 커널 4.6 이전 버전의 경우 대개 이 파라미터는 `nvme.io_timeout`입니다.

I/O 지연 시간이 이 제한 시간 파라미터의 값을 초과하면 Linux NVMe 드라이버는 I/O에 실패하고 파일 시스템 또는 애플리케이션에 오류를 반환합니다. I/O 작업에 따라 파일 시스템 또는 애플리케이션에서 오류를 다시 시도할 수 있습니다. 경우에 따라 파일 시스템이 읽기 전용으로 다시 탑재될 수 있습니다.

Xen 인스턴스에 연결된 EBS 볼륨과 비슷한 경험을 하기 위해서는 `nvme_core.io_timeout`을 가능한 최대값으로 설정하는 것이 좋습니다. 현재 커널의 경우 최대 4294967295인 것에 비해 이전 커널의 경우 최대 255입니다. Linux 버전에 따라 제한 시간이 이미 지원되는 최대값으로 설정되었을 수도 있습니다. 예를 들어 Amazon Linux AMI 2017.09.01 이상에서는 제한 시간이 기본적으로 4294967295로 설정됩니다.

제안된 최대 값보다 더 큰 값을 `/sys/module/nvme_core/parameters/io_timeout`에 쓰고 파일을 저장하려고 할 때 숫자 결과가 범위를 벗어났습니다 오류 발생 여부를 확인하여 Linux 배포에 대한 최대값을 확인할 수 있습니다.

Windows 인스턴스

Windows에서 기본 제한 시간은 60초이며 최대 허용 시간은 255초입니다. SCSI 미니포트 드라이버를 위한 레지스트리 항목에 설명된 절차를 사용하여 `TimeoutValue` 디스크 클래스 레지스트리 설정을 수정할 수 있습니다.

Amazon EBS 볼륨에 대한 NVMe Abort 명령

`Abort` 명령은 이전에 컨트롤러에 제출된 특정 명령을 종료하기 위해 실행되는 NVMe 관리 명령입니다. 일반적으로 디바이스 드라이버에서 I/O 작업 제한 시간 임계값을 초과한 스토리지 디바이스에 대해 이 명령을 실행합니다.

기본적으로 `Abort` 명령을 지원하는 Amazon EC2 인스턴스 유형은 연결된 Amazon EBS 볼륨에 `Abort` 명령이 실행될 때 컨트롤러에 이전에 제출된 특정 명령을 종료합니다. `Abort` 명령을 지원하지 않는 Amazon EC2 인스턴스는 연결된 Amazon EBS 볼륨에 `Abort` 명령이 실행될 때 아무런 조치도 취하지 않습니다.

`Abort` 명령은 다음에서 지원됩니다.

- NVMe 디바이스 버전 1.4 이상의 Amazon EBS 디바이스.
- Xen 기반 인스턴스 유형 및 다음 Nitro 기반 인스턴스 유형을 제외한 모든 Amazon EC2 인스턴스:
  - 범용: A1 | M5 | M5a | M5ad | M5d | M5dn | M5n | M5zn | M6g | M6gd | Mac1 | Mac2 | T3 | T3a | T4g
  - 컴퓨팅 최적화: C5 | c5a | C5ad | C5d | C5n | C6g | C6gd
  - 메모리 최적화: R5 | R5a | R5ad | R5d | R5dn | R5n | R6g | R6gd | U-12tb1 | U-18tb1 | U-24tb1 | U-3tb1 | U-6tb1 | U-9tb1 | X2gd | X2iezn | Z1d
  - 스토리지 최적화: D3 | D3en | I3en
  - 가속 컴퓨팅: DL1 | G4ad | G4dn | G5 | G5g | Inf1 | P3dn | P4d | P4de | VT1

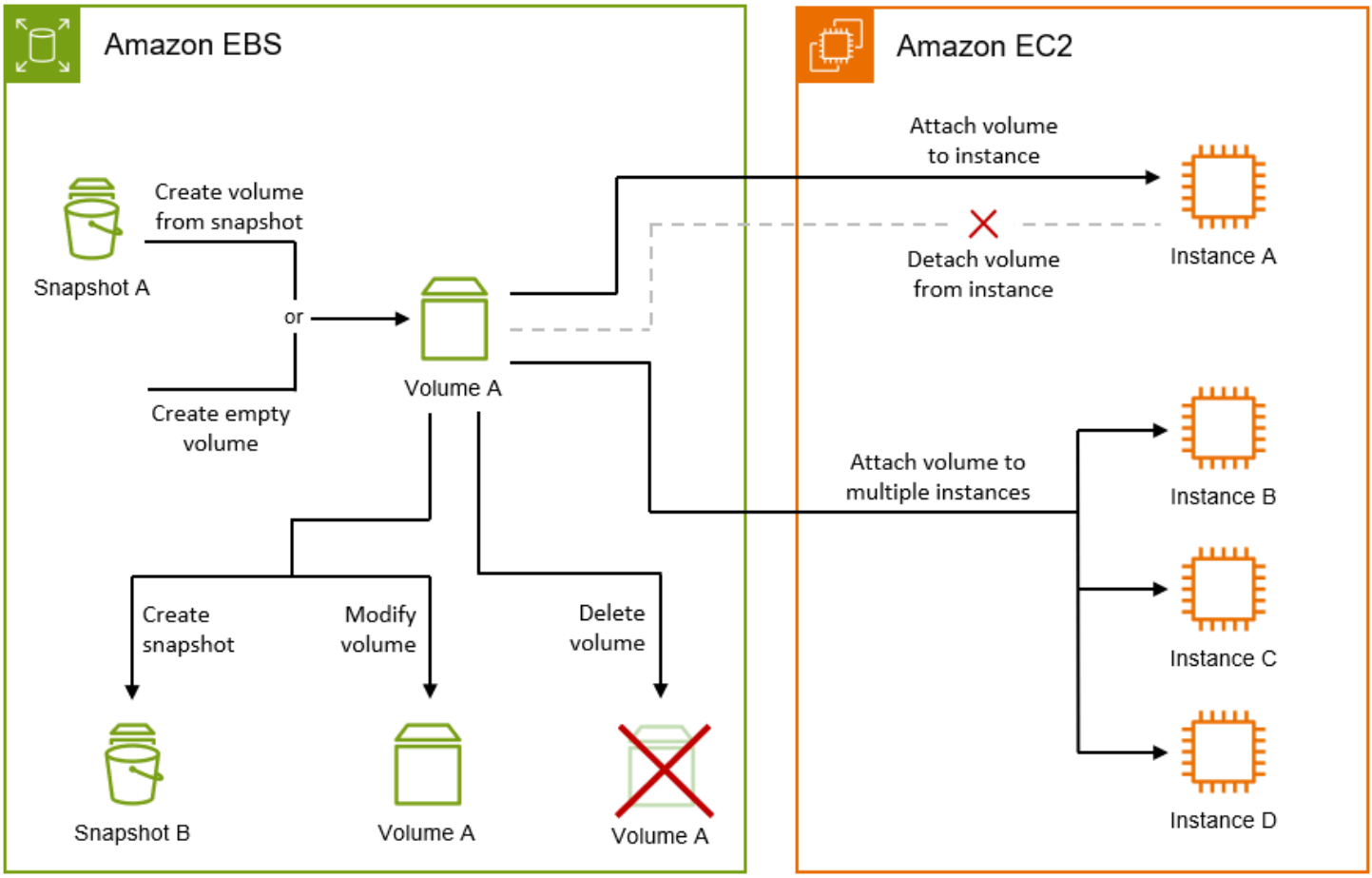
자세한 내용은 NVM Express Base Specification의 5.1 Abort command 섹션을 참조하세요.

Amazon EBS 볼륨 수명 주기

Amazon EBS 볼륨의 수명 주기는 생성 프로세스로 시작합니다. Amazon EBS 스냅샷에서 볼륨을 생성하거나 빈 볼륨을 생성할 수 있습니다. 볼륨을 사용하려면 먼저 볼륨과 동일한 가용 영역에 있는 하나 이상의 Amazon EC2 인스턴스에 연결해야 합니다. 여러 볼륨을 하나의 인스턴스에 연결할 수 있습니다. 필요하면 하나의 인스턴스에서 볼륨을 분리한 다음에 다른 인스턴스에 연결할 수 있습니다. 스토리지 요구 사항이 변경되면 언제든지 볼륨의 크기 또는 성능을 수정할 수 있습니다. Amazon EBS 스냅샷

을 생성하여 볼륨의 특정 시점 백업을 생성할 수 있습니다. 더 이상 볼륨이 필요하지 않으면 관련 스토리지 비용이 발생하지 않도록 삭제할 수 있습니다.

다음 이미지에서는 볼륨 수명 주기의 일부로 볼륨에서 수행할 수 있는 작업을 보여줍니다. 인스턴스에 연결하고 운영 체제 명령을 실행하여 수행하는 작업도 있습니다. 예를 들면 볼륨 형식 지정, 볼륨 탑재, 파티션 관리, 여유 디스크 공간 보기 등이 있습니다.



작업

- Amazon EBS 볼륨 생성
- 볼륨 공유
- Amazon EBS 볼륨 복사
- Amazon EC2 인스턴스에 Amazon EBS 볼륨 연결
- 다중 연결을 사용하여 여러 EC2 인스턴스에 EBS 볼륨 연결
- Amazon EBS 볼륨을 사용할 수 있도록 만들기
- Amazon EBS 볼륨에 대한 정보 보기
- 탄력적 볼륨 작업을 사용하여 Amazon EBS 볼륨 수정

- Amazon EC2 인스턴스에서 Amazon EBS 볼륨 분리
- Amazon EBS 볼륨 삭제

Amazon EBS 볼륨 생성

Amazon EBS 볼륨을 생성한 다음, 동일한 가용 영역에 있는 모든 EC2 인스턴스에 연결할 수 있습니다.

빈 볼륨을 생성하거나 Amazon EBS 스냅샷에서 볼륨을 생성할 수 있습니다. 스냅샷에서 생성하는 볼륨은 해당 스냅샷을 생성하는 데 사용된 볼륨과 정확히 일치합니다.

볼륨 초기화

스냅샷에서 볼륨을 생성하는 경우 스토리지 블록에 액세스하려면 먼저 Amazon S3에서 스냅샷의 스토리지 블록을 다운로드하여 볼륨에 기록해야 액세스할 수 있습니다. 이 프로세스를 볼륨 초기화라고 합니다. 이 시간 동안 볼륨에 I/O 지연 시간이 증가합니다. 모든 스토리지 블록을 다운로드하고 볼륨에 기록한 후에만 최대 볼륨 성능이 구현됩니다.

기본 볼륨 초기화 속도는 초기화 프로세스 내내 변동하므로 완료 시간을 예측할 수 없습니다.

볼륨 초기화와 관련된 성능 영향을 최소화하려면 볼륨 초기화를 위한 Amazon EBS 프로비저닝된 속도(볼륨 초기화 속도) 또는 빠른 스냅샷 복원을 사용할 수 있습니다. 자세한 내용은 Amazon EBS 볼륨 초기화 섹션을 참조하세요.

볼륨 암호화

볼륨의 암호화 상태는 계정의 기본적으로 암호화 활성화 여부와 스냅샷의 암호화 상태에 따라 달라집니다. 다음 표에는 가능한 암호화 결과가 요약되어 있습니다.

암호화 기본 제공	스냅샷 사용 여부	볼륨 암호화 결과	참고
비활성화됨	아니요	선택적 암호화	암호화를 활성화하면 사용할 KMS 키를 지정할 수 있습니다. 암호화를 활성화하지만 KMS 키를 지정하지 않으면 AWS 관리형 키 (aws/ebs)가 사용됩니다.
비활성화됨	예, 암호화되지 않음	선택적 암호화	암호화를 활성화하면 사용할 KMS 키를 지정할 수 있습니다. 암호화를 활성화하지만

암호화 기본 제공	스냅샷 사용 여부	볼륨 암호화 결과	참고
			KMS 키를 지정하지 않으면 AWS 관리형 키 (aws/ebs)가 사용됩니다.
비활성화됨	예, 암호화됨	자동 암호화	사용할 KMS 키를 지정할 수 있습니다. KMS 키를 지정하지 않으면 소스 스냅샷과 동일한 KMS 키를 사용하여 볼륨이 암호화됩니다.
활성화됨	아니요	자동 암호화	사용할 KMS 키를 지정할 수 있습니다. KMS 키를 지정하지 않으면 기본적으로 암호화에 지정된 키가 사용됩니다.
활성화됨	예, 암호화되지 않음	자동 암호화	사용할 KMS 키를 지정할 수 있습니다. KMS 키를 지정하지 않으면 기본적으로 암호화에 지정된 키가 사용됩니다.
활성화됨	예, 암호화됨	자동 암호화	사용할 KMS 키를 지정할 수 있습니다. KMS 키를 지정하지 않으면 볼륨은 소스 스냅샷(콘솔)과 동일한 키 또는 기본적으로 암호화에 지정된 키(CLI/API)를 사용하여 암호화됩니다.

추가 고려 사항

- 볼륨은 동일한 가용 영역의 인스턴스에 연결되어야 합니다.
- 볼륨은 `available` 상태로 전환된 후에만 사용할 수 있습니다.
- 콘솔을 사용하여 볼륨을 생성하는 경우 gp3이 기본 볼륨 유형입니다. 명령줄 도구, API 및 SDK의 경우 gp2가 기본 볼륨 유형입니다.
- Outpost에서 실행되는 인스턴스와 함께 볼륨을 사용하려면 인스턴스와 동일한 Outpost에서 볼륨을 생성해야 합니다.
- Windows 인스턴스에서 사용할 볼륨을 생성하고 2048GiB보다 큰 경우 GPT 파티션 테이블을 사용하도록 볼륨을 구성해야 합니다. 자세한 내용은 Amazon EBS 볼륨 제약 조건 및 2TB보다 큰 디스크에 대한 Windows 지원을 참조하세요.

- 볼륨은 Amazon EC2 인스턴스를 시작하여 간접적으로 생성될 수도 있습니다. 인스턴스를 시작하는 데 사용된 AMI 또는 인스턴스 시작 요청 자체에 Amazon EBS 볼륨에 대한 블록 디바이스 매핑이 포함될 수 있습니다. 자세한 내용은 블록 디바이스 매핑 섹션을 참조하세요.

Console

볼륨을 생성하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 볼륨을 선택한 다음 볼륨 생성을 선택합니다.
3. (Outpost 고객만 해당) Outpost ARN에 대해 볼륨을 생성할 AWS Outpost의 ARN을 입력합니다.
4. 볼륨 유형(Volume type)에서 생성할 볼륨 유형을 선택합니다. 사용 가능한 볼륨 유형에 대한 자세한 내용은 Amazon EBS 볼륨 유형 섹션을 참조하세요.
5. [크기(Size)]에 볼륨 크기를 GiB 단위로 입력합니다. 자세한 내용은 Amazon EBS 볼륨 제약 조건 섹션을 참조하세요.
6. (*io1*, *io2* 및 *gp3*에만 해당) IOPS에 볼륨이 제공해야 하는 최대 초당 입출력 작업량(IOPS) 수를 입력합니다.
7. (*gp3*에만 해당) 처리량에 볼륨에서 제공해야 하는 처리량(MiB/s)을 입력합니다.
8. 가용 영역에서 볼륨을 생성할 가용 영역을 선택합니다.
9. 스냅샷 ID에서 다음 중 하나를 수행합니다.
   - 빈 볼륨을 생성하려면 기본값(스냅샷에서 볼륨을 생성하지 않음)을 유지합니다.
   - 스냅샷에서 볼륨을 생성하려면 사용할 스냅샷을 선택합니다.
10. 볼륨 초기화 속도에 스냅샷을 선택한 경우, 선택적으로 스냅샷 블록을 Amazon S3에서 해당 볼륨으로 생성 후 다운로드할 볼륨 초기화 속도를 MiB/s 단위로 지정할 수 있습니다. 자세한 내용은 볼륨 초기화에 Amazon EBS 프로비저닝 속도 사용 섹션을 참조하세요. 기본 초기화 속도 또는 빠른 스냅샷 복원(선택한 스냅샷에 대해 활성화된 경우)을 사용하려면 속도를 지정하지 마세요.
11. (*io1* 및 *io2*에만 해당) Amazon EBS 다중 연결의 볼륨을 활성화하려면 다중 연결 활성화를 선택합니다. 자세한 내용은 다중 연결을 사용하여 여러 EC2 인스턴스에 EBS 볼륨 연결 섹션을 참조하세요.
12. 볼륨의 암호화 상태를 설정합니다.

- 계정에 기본적으로 암호화가 활성화되어 있으면 암호화는 자동이며 비활성화할 수 없습니다.
- 암호화된 스냅샷을 선택한 경우 암호화는 자동이며 비활성화할 수 없습니다.
- 계정에 기본적으로 암호화가 활성화되어 있지 않고 암호화되지 않은 스냅샷을 선택하거나 스냅샷을 선택하지 않은 경우 암호화는 선택 사항입니다.

13. (선택 사항) 볼륨에 사용자 지정 태그를 할당하려면 태그 섹션에서 태그 추가를 선택한 다음 태그 키 및 값 페어를 입력합니다.
14. 볼륨 생성을 선택합니다.
15. 볼륨을 사용하려면 볼륨이 `available` 상태에 도달할 때까지 기다린 다음 동일한 가용 영역의 Amazon EC2 인스턴스에 연결합니다. 자세한 내용은 Amazon EC2 인스턴스에 Amazon EBS 볼륨 연결 섹션을 참조하세요.

AWS CLI

볼륨을 생성하려면

create-volume 명령을 사용합니다. 다음 예제는 지정된 가용 영역에 크기가 100GiB인 빈 gp3 볼륨을 생성합니다.

```
aws ec2 create-volume \
    --volume-type gp3 \
    --size 100 \
    --availability-zone us-east-1a
```

PowerShell

볼륨을 생성하려면

New-EC2Volume cmdlet을 사용합니다. 다음 예제는 지정된 가용 영역에 크기가 100GiB인 빈 gp3 볼륨을 생성합니다.

```
New-EC2Volume `
    -VolumeType gp3 `
    -Size 100 `
    -AvailabilityZone us-east-1a
```

볼륨 공유

AWS Resource Access Manager (AWS RAM)를 사용하여 Amazon EBS 볼륨을 다른 AWS 계정과 공유할 수 있습니다. 볼륨을 공유한 후 소비 계정은 볼륨 정보를 보고 동일한 가용 영역에 공유 볼륨의 복사본을 생성할 수 있습니다. 복사본 생성에 대한 자세한 내용은 섹션을 참조하세요Amazon EBS 볼륨 복사. 사용 계정은 공유 볼륨의 스냅샷을 연결, 수정, 삭제 또는 생성할 수 없습니다.

예를 들어, 개발 계정과 프로덕션 볼륨을 공유하여 계정 수준 격리를 유지하면서 up-to-date 프로덕션 데이터로 테스트 환경을 새로 고칠 수 있습니다.

다음과 같은 볼륨을 공유할 수 있습니다.

- 암호화되지 않음
- 고객 관리형 키로 암호화됨

Amazon EBS의 기본 AWS 관리형 키로 암호화된 볼륨은 공유할 수 없습니다.

내용

- 권한
- 볼륨 공유
- 공유 볼륨 보기
- 볼륨 공유 시 고려 사항
- 공유 활동 모니터링
- 공유 볼륨 및 휴지통
- 요금

권한

공유 볼륨에 대한 리소스 공유를 생성할 때 관리형 권한을 리소스 공유와 연결해야 합니다. 관리형 권한은 계정 소비가 리소스 공유의 공유 볼륨에서 수행할 수 있는 작업을 결정합니다.는 다음과 같은 AWS 관리형 권한을 AWS RAM 제공합니다.

- AWSRAMDefaultPermissionEBSVolume(기본값) - 소비 계정에 볼륨 정보를 볼 수 있는 권한을 부여합니다.
- AWSRAMPermissionEBSVolumeCopyAccess - 소비 계정에 볼륨 정보를 보고 공유 볼륨의 복사본을 생성할 수 있는 권한을 부여합니다.

또는 ec2:Volume 리소스 유형에 대한 고객 관리형 권한을 생성할 수 있습니다. 자세한 내용은 AWS RAM 사용 설명서의 고객 관리형 권한 생성 및 사용을 참조하세요.

볼륨 공유

다음 방법 중 하나를 사용하여 Amazon EBS 볼륨을 공유합니다.

Console

볼륨을 공유하려면

1. 에서 Amazon Elastic Compute Cloud 콘솔을 엽니다https://console.aws.amazon.com/ec2/.
2. 탐색 창에서 볼륨을 선택합니다.
3. 볼륨을 선택한 다음 작업, 볼륨 공유를 선택합니다.
4. 다음 중 하나를 수행하세요.
   - 기존 리소스 공유에 볼륨을 추가하려면 리소스 공유를 선택합니다.
   - 새 리소스 공유를 생성하려면 새 리소스 공유 생성을 선택합니다. 그러면 공유를 구성하는 AWS RAM 콘솔이 열립니다.
5. 보안 주체의 경우 볼륨을 공유할 계정을 추가합니다. 개별 계정 IDs, 조직 단위(OUs) 또는 전체 조직을 추가할 수 있습니다.
6. 관리형 권한에서 다음 중 하나를 선택합니다.
   - AWSRAMDefaultPermissionEBSVolume(기본값) - 소비 계정은 볼륨 메타데이터만 볼 수 있습니다.
   - AWSRAMPermissionEBSVolumeCopyAccess - 소비 계정은 볼륨 메타데이터를 보고 공유 볼륨의 복사본을 생성할 수 있습니다.
   - ec2:Volume 리소스 유형에 대해 생성한 고객 관리형 권한입니다.
7. 리소스 공유 생성을 선택합니다.

AWS CLI

create-resource-share 명령을 사용합니다. 볼륨 ARN, 소비 계정 및 권한 ARN을 지정합니다. 볼륨 ARN을 찾으려면 describe-volumes를 호출합니다. 응답에는 각 볼륨의 VolumeArn 필드가 포함됩니다.

다음 예제에서는 계정 222222222222vol-1234567890abcdef0과 볼륨을 공유하는 리소스 공유를 생성하고 AWSRAMPermissionEBSVolumeCopyAccess 관리형 권한을 할당합니다.

```
aws ram create-resource-share \
--name my-volume-share \
--resource-arns arn:aws:ec2:us-east-1:111111111111:volume/vol-1234567890abcdef0 \
--principals 222222222222 \
--permission-arns arn:aws:ram::aws:permission/AWSRAMPermissionEBSVolumeCopyAccess \
--region us-east-1
```

기본 권한(AWSRAMDefaultPermissionEBSVolume, 설명만)으로 공유하려면 --permission-arns 파라미터를 생략합니다.

```
aws ram create-resource-share \
--name my-volume-share \
--resource-arns arn:aws:ec2:us-east-1:111111111111:volume/vol-1234567890abcdef0 \
--principals 222222222222 \
--region us-east-1
```

리소스 공유에 대한 자세한 내용은 AWS RAM 사용 설명서의 리소스 공유 생성을 참조하세요.

공유 볼륨 보기

리소스 공유를 생성하면 지정된 계정이 초대를 받습니다. 에서 조직 외부의 계정은 공유 볼륨에 액세스하려면 초대를 수락해야 AWS Organizations 합니다. 자세한 내용은 AWS RAM 사용 설명서의 공유 리소스 초대 수락 및 거부를 참조하세요.

다음 방법 중 하나를 사용하여 계정과 공유된 볼륨을 볼 수 있습니다.

Console

공유 볼륨을 보려면

1. 에서 Amazon Elastic Compute Cloud 콘솔을 엽니다https://console.aws.amazon.com/ec2/.
2. 탐색 창에서 볼륨을 선택합니다.
3. 다음 중 하나를 수행하세요.
   - 내 계정과 공유된 모든 볼륨을 표시하려면 나와 공유 사전 설정 필터를 선택합니다.
   - 필터 추가, 소유자 ID를 선택한 다음 공유 계정 ID를 입력합니다.

공유 볼륨은 소유자 ID 열에 공유 계정 ID를 표시합니다.

AWS CLI

owner-id 필터와 함께 describe-volumes 명령을 사용하여 특정 계정에서 공유한 볼륨을 표시합니다.

```
aws ec2 describe-volumes \
--filters Name=owner-id,Values=111111111111 \
--region us-east-1
```

공유 볼륨에는 응답에 공유 계정 ID를 보여주는 OwnerId 필드가 포함됩니다.

볼륨 공유 시 고려 사항

볼륨을 다른 계정과 공유할 때는 다음 사항을 고려하세요.

- 암호화되지 않은 볼륨과 고객 관리형 키로 암호화된 볼륨을 공유할 수 있습니다. Amazon EBS의 기본 AWS 관리형 키로 암호화된 볼륨은 공유할 수 없습니다.
- 볼륨을 공유하려면 볼륨 소유자의 계정에 기본 Amazon EBS 암호화 키에 대한 kms:DescribeKey 권한이 있어야 합니다. 이는 암호화되지 않은 볼륨을 포함한 모든 볼륨에 필요합니다. 서비스가 공유 프로세스 중에 볼륨의 암호화 상태를 검증하기 때문입니다.
- 고객 관리형 키로 암호화된 볼륨을 공유하려면 KMS 키도 소비 계정과 공유해야 합니다. 자세한 내용은 AWS Key Management Service 개발자 가이드의 다른 계정의 사용자가 CMK를 사용하도록 허용을 참조하세요.
- 암호화된 공유 볼륨의 복사본을 생성하려면 소비 계정에 KMS 키kms:Decrypt에 kms:CreateGrant, kms:GenerateDataKey, kms:ReEncrypt*, kms:GenerateDataKeyWithoutPlaintext및가 필요합니다.
- 사용 계정은 공유 볼륨의 스냅샷을 연결, 수정, 삭제 또는 생성할 수 없습니다. 볼륨 메타데이터만 보고 복사본을 생성할 수 있습니다.
- 볼륨이 공유되는 모든 계정에서 공유 볼륨에 대해 한 번에 하나의 복사 작업만 진행할 수 있습니다. 추가 복사 요청은 현재 작업이 완료될 때까지 오류를 반환합니다.
- 가용 영역 이름(예: us-east-1a)은 서로 다른 AWS 계정의 서로 다른 물리적 위치에 매핑됩니다. 가용 영역 IDs(예: use1-az1)를 사용하여 계정 간에 동일한 물리적 위치를 식별합니다. 자세한 내용은 AWS RAM 사용 설명서의 AWS 리소스의 가용 영역 IDs를 참조하세요.
- 리소스 공유에서 볼륨을 제거해도 진행 중인 복사 작업은 취소되지 않습니다.
- 소스 볼륨을 삭제해도 진행 중인 복사 작업은 취소되지 않습니다.

공유 활동 모니터링

AWS CloudTrail API 로깅

모든 볼륨 공유 및 복사 API 호출이 로그인됩니다 AWS CloudTrail. `CopyVolumes` API 호출은 복사를 시작하는 계정과 볼륨 소유자의 계정 모두에 대해 로깅됩니다. 자세한 내용은 를 사용하여 Amazon Elastic Compute Cloud API 호출 로깅을 참조하세요 AWS CloudTrail.

Amazon EventBridge 알림

Amazon EBS는 교차 계정 복사 작업이 완료되면 Amazon EventBridge 이벤트를 전송합니다. 볼륨 소유자와 소비 계정 모두 이벤트를 수신합니다. 볼륨 복사 이벤트는 이벤트 이름 `EBS Volume Notification`와 함께 세부 정보 유형을 사용합니다`sharedVolumeCopy`. 이벤트에는 볼륨 ID, 완료 상태(`completed` 또는 `failed`), 타임스탬프가 포함됩니다.

알림을 설정하려면 `EBS Volume Notification` 이벤트를 `sharedVolumeCopy` 이벤트 이름과 일치시키는 Amazon EventBridge 규칙을 생성합니다. 자세한 내용은 Amazon EBS에 대한 Amazon EventBridge 이벤트를 참조하세요.

공유 볼륨 및 휴지통

볼륨에 적용되는 휴지통 보존 규칙이 있는 경우 공유 볼륨을 삭제하면 휴지통으로 이동합니다. 볼륨은 보존 기간 동안 AWS RAM 리소스 공유에 유지됩니다. 휴지통에서 볼륨을 복원하면 삭제 전에 볼륨이 속한 동일한 리소스 공유를 통해 액세스할 수 있습니다.

휴지통에 있는 볼륨에 대한 다음 리소스 공유 동작에 유의하세요.

- 볼륨이 휴지통에 있는 동안 리소스 공유에서 계정을 제거한 경우 볼륨을 복원해도 해당 계정 액세스 권한이 다시 부여되지 않습니다.
- 볼륨이 휴지통에 있는 동안 리소스 공유에 계정을 추가한 경우 해당 계정은 복원 후 볼륨에 액세스할 수 있습니다.

휴지통에서 볼륨을 복원한 후 AWS RAM 콘솔에서 리소스 공유를 검토하여 의도한 계정만 액세스할 수 있는지 확인합니다.

요금

를 통한 볼륨 공유에는 비용이 들지 않습니다 AWS RAM. 복사본을 생성하는 계정은 볼륨 복사 요금과 새 볼륨에 대한 일반 Amazon EBS 볼륨 요금을 지불합니다. 자세한 내용은 Amazon EBS 요금을 참조하세요.

Amazon EBS 볼륨 복사

볼륨 복사는 소스 볼륨의 충돌 일관성이 있는 point-in-time 복사입니다. 여기에는 볼륨 복사 초기화가 시작되는 시점에 소스 볼륨에 기록된 모든 데이터 블록이 포함됩니다. 볼륨 복사본은 고유한 볼륨 ID를 가집니다. 볼륨 복사본은 즉각적으로 생성되며 `available` 상태에 도달하면 Amazon EC2 인스턴스에 연결될 수 있습니다. 볼륨 복사본을 사용하면 테스트 및 개발 환경을 위한 프로덕션 데이터를 빠르게 복사할 수 있습니다.

다른 AWS 계정에서 공유한 볼륨에서 복사본을 생성할 수도 있습니다. 다른 계정에서 볼륨을 복사하려면 AWS Resource Access Manager ()를 통해 소스 볼륨을 공유해야 합니다AWS RAM. 자세한 내용은 볼륨 공유 섹션을 참조하세요.

Initialization(초기화)

볼륨 복사본은 생성 후에 초기화됩니다. 초기화 중, 데이터 블록은 소스 볼륨에서 복사되어 백그라운드의 볼륨 복사본에 기록됩니다. 볼륨은 초기화 완료 시까지 `initializing` 상태로 유지됩니다.

초기화 중의 성능

복사 작업은 소스 볼륨의 성능에 영향을 주지 않습니다. 복사 프로세스 중에는 소스 볼륨을 정상적으로 계속 사용할 수 있습니다. 소스 볼륨의 데이터가 복사될 때까지 기다리지 않고도 복사된 볼륨에 즉시 액세스할 수 있습니다. 볼륨 복사본을 사용하면 10밀리초 미만의 지연 시간으로 데이터에 즉시 액세스할 수 있습니다. 그러나 실제 지연 시간은 볼륨 유형에 따라 다를 수 있습니다. 초기화 중에 볼륨 복사는 다음 세 가지 값 중 가장 낮은 값과 동일한 기준 성능을 제공합니다.

- 초당 3,000개의 입/출력 작업(IOPS) 및 125MiB/s
- 소스 볼륨의 프로비저닝된 성능
- 볼륨 복사본의 프로비저닝된 성능

볼륨 복사본은 다음 기준이 충족될 때 기준 성능을 초과할 수 있습니다.

1. 소스 볼륨과 볼륨 복사은 모두 3,000 IOPS 및 125MiB/s 이상으로 프로비저닝됩니다.
2. 소스 볼륨에 사용되지 않는 성능 용량이 있습니다(구동 성능이 프로비저닝된 성능보다 작음).

예를 들어 소스 볼륨이 10,000 IOPS로 프로비저닝되었으며 워크로드가 현재 5,000 IOPS만 구동하고 볼륨 복사본이 10,000 IOPS로 프로비저닝된 경우, 볼륨 복사본은 소스 볼륨의 미사용 5,000 IOPS를 사용하여 초기화 중에 3,000 IOPS 기준 성능보다 높은 성능을 달성할 수 있습니다.

초기화 기간

볼륨 복사본의 초기화에 걸리는 시간은 볼륨 복사본을 생성할 때 소스 볼륨에 기록된 블록 데이터의 크기에 따라 다릅니다. 볼륨 복사본은 다음 일반 지침에 따라 최선의 작업을 기반으로 초기화됩니다. 데이터 블록의 첫 1TiB의 경우 볼륨 초기화에 최대 6시간이 소요됩니다. 이후의 각 1TiB 데이터 블록은 최대 16TiB까지 초기화에 TiB당 1.2시간이 소요됩니다. 16TiB보다 큰 데이터의 경우 초기화에 24시간이 소요됩니다.

초기화 진행 상황 모니터링

초기화 진행 상황은 describe-volume-status AWS CLI 명령 또는 Amazon EventBridge를 사용하여 모니터링할 수 있습니다. 자세한 정보는 Amazon EBS 볼륨 초기화 상태 모니터링을 참조하세요.

암호화(Encryption)

볼륨을 복사할 때 복사의 암호화는 소스 볼륨의 암호화 상태, 계정의 기본 설정 암호화, 복사 요청에서 지정하는 `Encrypted` 및 `KmsKeyId` 파라미터에 따라 달라집니다.

암호화되지 않은 소스 볼륨

소스 볼륨이 암호화되지 않고 계정에 대해 기본적으로 암호화가 비활성화된 경우 복사본은 암호화되지 않습니다. `Encrypted` 파라미터를 로 설정하여 복사본을 암호화할 수 있습니다`true`. 를 지정하지 않으면 KMS key계정의 기본 Amazon EBS 암호화 키가 사용됩니다.

소스 볼륨이 암호화되지 않고 계정에 암호화가 기본적으로 활성화된 경우 복사본이 자동으로 암호화됩니다. 를 지정하지 않으면 KMS key계정의 기본 Amazon EBS 암호화 키가 사용됩니다.

암호화된 소스 볼륨

소스 볼륨이 암호화된 경우 사본은 항상 암호화됩니다. 암호화된 소스 볼륨에서는 암호화되지 않은 복사본을 생성할 수 없습니다.

동일 계정 복사본의 경우 복사본은 기본적으로 소스 볼륨과 동일한 KMS 키를 사용합니다. 다른 고객 관리형 키를 지정하여 사본을 다시 암호화할 수 있습니다.

교차 계정 복사본의 경우 복사본은 기본적으로 계정의 기본 Amazon EBS 암호화 키를 사용합니다. `KmsKeyId` 파라미터에서 다른 고객 관리형 키를 지정할 수 있습니다. 소스 계정의 KMS 키를 사용하여 해당 키가 계정과 공유된 경우 볼륨 복사본을 암호화할 수 있습니다.

교차 계정 암호화 요구 사항

고객 관리형 키로 암호화된 공유 볼륨을 복사하려면 볼륨 소유자가 KMS 키를 계정과 공유해야 합니다. 자세한 내용은 AWS Key Management Service 개발자 안내서의 다른 계정의 사용자가를 사용하도록 허용 KMS key을 참조하세요.

Amazon EBS의 기본 AWS 관리형 키로 암호화된 볼륨은 계정 간에 공유하거나 복사할 수 없습니다.

암호화 결과

암호화되지 않은 소스 볼륨

암호화 기본 제공	를 지정합니다. KMS key	결과
비활성화됨	아니요	복사가 암호화되지 않음
비활성화됨	예	지정된 키로 암호화된 복사
활성화됨	아니요	계정의 기본 Amazon EBS 암호화 키로 암호화된 복사
활성화됨	예	지정된 키로 암호화된 복사

암호화된 소스 볼륨(동일 계정)

를 지정합니다. KMS key	결과
아니요	소스 볼륨과 동일한 키로 암호화된 복사
예	지정된 키로 다시 암호화된 복사본

암호화된 소스 볼륨(교차 계정)

를 지정합니다. KMS key	결과
아니요	계정의 기본 Amazon EBS 암호화 키로 암호화된 복사
예	지정된 키로 암호화된 복사

교차 계정 복사본은 기본적으로 소스 계정의 KMS 키를 재사용하지 않습니다. 소스 계정의 키를 사용하려면 해당 키를 계정과 공유해야 합니다.

고려 사항

- 볼륨 복사본은 소스 볼륨의 충돌 일치 point-in-time 상태를 캡처합니다. 여기에는 복사 요청 시 볼륨에 기록된 데이터만 포함됩니다. 애플리케이션 또는 운영 체제에서 캐시한 데이터는 포함되지 않습니다. 애플리케이션에 일관되게 적용되는 복사본을 얻으려면 복사본을 생성하기 전에 볼륨에 대한 쓰기를 일시 중지하거나 애플리케이션을 중지합니다. 데이터베이스의 경우 적절한 메커니즘을 사용하여 더티 페이지를 플러시하고 I/O를 고정시킵니다(예: Linux`fsfreeze`의 경우 , Windows의 경우 VSS). 복사 요청이 반환된 후 쓰기를 재개할 수 있습니다.
- 소스 볼륨의 볼륨 복사본은 한 번에 하나만 생성할 수 있습니다. 이전 볼륨 복사본이 완전히 초기화된 후에만 동일한 소스 볼륨에 대한 후속 복사본을 생성할 수 있습니다.
- 리전당 최대 5개의 볼륨 복사를 동시에 진행할 수 있습니다. 이 할당량을 초과하는 경우 `CopyVolumesLimitExceeded` 오류가 발생합니다. 필요한 경우 할당량 증가를 요청할 수 있습니다.
- 볼륨 복사본은 소스 볼륨과 같은 가용 영역에 생성되어야 합니다.
- 볼륨 복사본의 크기는 소스 볼륨의 크기보다 크거나 같아야 합니다.
- 볼륨 복사본이 생성 또는 초기화되는 동안에는 복사할 수 없습니다.
- 볼륨 복사본을 생성하려면 소스 볼륨이 `available` 또는 `in-use` 상태이고 볼륨 수정이 `completed` 또는 `optimizing` 상태여야 합니다.
- 볼륨 복사본에는 일반 Amazon EBS 볼륨과 동일한 계정 및 리전 스토리지 및 IOPS 할당량이 적용됩니다. 자세한 내용은 Amazon EBS 할당량을 참조하세요.
- 복사 작업이 진행되는 동안 소스 볼륨을 삭제하는 경우, 복사 작업은 여전히 완료됩니다.
- 소스 볼륨에 할당된 태그는 볼륨 복사에 할당되지 않습니다.
- Outposts 또는 Wavelength Zone에 있는 볼륨에 대해서는 복사본을 생성할 수 없습니다.

요금

소스 볼륨의 복사본을 생성하면 복사 작업에 대한 요금이 부과됩니다. 복사 작업 요금은 복사 생성 시 소스 볼륨에 기록된 데이터 블록의 크기를 기준으로 합니다. 새로 생성된 볼륨은 다른 Amazon EBS 볼륨과 동일한 방식으로 청구됩니다. 자세한 내용은 Amazon EBS 요금을 참조하세요.

볼륨 복사

Amazon EBS 볼륨을 수정하려면 다음 방법 중 하나를 사용합니다.

Console

볼륨을 복사하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 볼륨을 선택합니다.
3. 복사할 볼륨을 선택하고 작업, 볼륨 수정을 선택합니다.
4. 볼륨 유형에서 복사할 볼륨 유형을 선택합니다. 기본 볼륨 유형은 gp3입니다.
5. 크기에 볼륨 복사본 크기를 GiB 단위로 입력합니다. 크기는 소스 볼륨의 크기와 같거나 커야 합니다.
6. (*io1*, *io2* 및 *gp3*에만 해당) IOPS에 볼륨 복사본의 최대 초당 입출력 작업량(IOPS) 수를 입력합니다.
7. (*gp3*에만 해당) 처리량에 볼륨 복사본의 처리량을 MiB/s 단위로 입력합니다.
8. (*io1* 및 *io2*에만 해당) Amazon EBS 다중 연결에 대한 볼륨 복사본을 활성화하려면 다중 연결 활성화를 선택합니다.
9. 암호화의 경우 기본값은 소스 볼륨과 계정 설정에 따라 달라집니다. 동일 계정 복사본의 경우 복사본은 소스 볼륨의 암호화 키를 상속합니다. 교차 계정 복사본의 경우 계정의 기본 Amazon EBS 암호화 키가 사용됩니다. 다른 고객 관리형 키를 선택할 수 있습니다.
10. (선택 사항) 볼륨 복사본에 사용자 지정 태그를 할당하려면 태그 섹션에서 태그 추가를 선택한 다음 태그 키 및 값 페어를 입력합니다.
11. 볼륨 복사를 선택합니다.
12. 복사된 볼륨은 `creating` 상태로 시작되었다가 잠시 후에 `available` 상태로 전환됩니다. 그 다음, 동일한 가용 영역에 있는 모든 Amazon EC2 인스턴스에 연결할 수 있습니다.

AWS CLI

볼륨을 복사하려면

copy-volumes 명령을 사용합니다.

다음 예제는 `vol-01234567890abcdef`에 대한 `gp3` 볼륨 유형, `100`GiB 크기 및 `250`MiB/s 처리량의 볼륨 복사본을 생성합니다.

```
aws ec2 copy-volumes \
--source-volume-id vol-01234567890abcdef \
```

```
--volume-type gp3 \
--size 100 \
--throughput 250
```

복사 중에 암호화되지 않은 볼륨을 암호화하려면

copy-volumes 명령을 --encrypted 및 --kms-key-id 파라미터와 함께 사용합니다.

```
aws ec2 copy-volumes \
--source-volume-id vol-01234567890abcdef \
--encrypted \
--kms-key-id alias/my-key
```

다른 고객 관리형 키로 다시 암호화하려면

파라미터와 함께 copy-volumes 명령을 사용하여 다른 키를 --kms-key-id 지정합니다.

```
aws ec2 copy-volumes \
--source-volume-id vol-01234567890abcdef \
--kms-key-id arn:aws:kms:us-east-1:222222222222:key/abcd1234-a123-456a-a12b-
a123b4cd56ef
```

PowerShell

볼륨을 복사하려면

Copy-EC2Volume cmdlet을 사용합니다.

다음 예제는 vol-01234567890abcdef에 대한 gp3 볼륨 유형, 100GiB 크기 및 250MiB/s 처리량의 볼륨 복사본을 생성합니다.

```
Copy-EC2Volume `
-SourceVolumeId vol-01234567890abcdef `
-VolumeType gp3 `
-Size 100 `
-Throughput 250
```

복사 중에 암호화되지 않은 볼륨을 암호화하려면

Copy-EC2Volume cmdlet을 -Encrypted 및 -KmsKeyId 파라미터와 함께 사용합니다.

```
Copy-EC2Volume `
-SourceVolumeId vol-01234567890abcdef `
-Encrypted $true `
-KmsKeyId "alias/my-key"
```

다른 고객 관리형 키로 다시 암호화하려면

Copy-EC2Volume cmdlet을 -KmsKeyId 파라미터와 함께 사용하여 다른 키를 지정합니다.

```
Copy-EC2Volume `
-SourceVolumeId vol-01234567890abcdef `
-KmsKeyId "arn:aws:kms:us-east-1:222222222222:key/abcd1234-a123-456a-a12b-
a123b4cd56ef"
```

Amazon EC2 인스턴스에 Amazon EBS 볼륨 연결

사용 가능한 EBS 볼륨을 해당 볼륨과 동일한 가용 영역에 있는 하나 이상의 인스턴스에 연결할 수 있습니다.

시작 시 인스턴스에 EBS 볼륨을 추가하는 방법에 대한 내용은 인스턴스 블록 디바이스 매핑을 참조하세요.

고려 사항

- 인스턴스에 연결할 수 있는 Amazon EBS 볼륨의 최대 수는 인스턴스 유형에 따라 달라집니다. 인스턴스의 볼륨 연결 한도를 초과하면 AttachmentLimitExceeded 오류와 함께 연결 요청이 실패합니다. 자세한 내용은 인스턴스 볼륨 제한을 참조하세요.
- 동일한 가용 영역에 있는 인스턴스에만 볼륨을 연결할 수 있습니다.
- 다중 연결 사용 볼륨은 최대 16개의 인스턴스에 연결할 수 있습니다. 자세한 내용은 다중 연결을 사용하여 여러 EC2 인스턴스에 EBS 볼륨 연결 섹션을 참조하세요.
- 볼륨에 AWS Marketplace 제품 코드가 있는 경우:
  - 중지된 인스턴스에만 연결할 수 있습니다.
  - 볼륨에 있는 AWS Marketplace 코드를 구독해야 합니다.
  - 유형 및 운영 체제와 같은 인스턴스의 구성은 해당 특정 AWS Marketplace 코드를 지원해야 합니다. 예를 들어, Windows 인스턴스의 볼륨을 Linux 인스턴스로 연결할 수 없습니다.
  - AWS Marketplace 코드는 볼륨에서 인스턴스로 복사됩니다.

- 지정한 이 디바이스 이름은 Amazon EC2에서 사용합니다. 블록 디바이스 드라이버는 지정한 디바이스와 다른 디바이스 이름으로 디바이스를 탑재할 수 있습니다. 자세한 내용은 Amazon EC2 인스턴스의 볼륨에 대한 디바이스 이름을 참조하세요.
- 일부 경우에는 /dev/xvda 또는 /dev/sda에 연결된 볼륨이 아닌 다른 볼륨이 Linux 인스턴스의 루트 볼륨이 될 수 있습니다. 이 상황은 다른 인스턴스의 루트 볼륨이나 루트 볼륨의 스냅샷에서 생성된 볼륨을 기존 루트 볼륨의 인스턴스에 연결하는 경우에 발생할 수 있습니다. 자세한 내용은 잘못된 볼륨에서 부팅을 참조하세요.
- 일부 인스턴스 유형은 둘 이상의 EBS 카드를 지원합니다. EBS 카드 인덱스를 지정하여 연결할 볼륨의 EBS 카드를 선택할 수 있습니다. 여러 EBS 카드를 지원하는 인스턴스는 EBS 카드를 참조하세요.
  - 루트 볼륨은 EBS 카드 인덱스에 연결되어야 합니다0.
  - 여러 EBS 카드를 지원하는 인스턴스의 경우 EBS 카드 인덱스를 지정하지 않으면 볼륨이 EBS 카드 인덱스에 연결됩니다0.
  - 고성능 워크로드에 맞게 EC2 인스턴스를 구성할 때는 EBS 카드의 성능 제한에 도달하지 않도록 성능 요구 사항에 따라 EBS 카드 간에 EBS 볼륨의 균형을 조정해야 합니다.
  - 인스턴스 유형에 대한 볼륨 연결 제한은 각 EBS 카드에 균등하게 분산됩니다. 예를 들어 2개의 EBS 카드가 있는 볼륨 연결을 지원하는 EC2 인스턴스에서 각 EBS 카드는 최대 개의 64 볼륨 연결을 지원할 수 있습니다. 128 EBS 카드 연결 한도를 초과하면 `CardAttachmentLimitExceeded` 오류가 발생하여 요청이 실패합니다.

Console

인스턴스에 EBS 볼륨을 연결하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 볼륨을 선택합니다.
3. 연결할 볼륨을 선택하고 작업(Actions), 볼륨 연결(Attach volume)을 선택합니다.
4. 인스턴스(Instance)에서 인스턴스 ID를 입력하거나 옵션 목록에서 인스턴스를 선택합니다.
5. 디바이스 이름에서 다음 중 하나를 수행합니다.
   - 루트 볼륨의 경우 목록의 루트 볼륨용으로 예약됨 섹션에서 필요한 디바이스 이름을 선택합니다. 일반적으로 Linux 인스턴스의 경우 AMI에 따라 /dev/sda1 또는 /dev/xvda이고 Windows 인스턴스의 경우 /dev/sda1입니다.
   - 데이터 볼륨의 경우 목록의 데이터 볼륨에 권장 섹션에서 사용 가능한 디바이스 이름을 선택합니다.

- 사용자 지정 디바이스 이름을 사용하려면 사용자 지정 디바이스 이름 지정을 선택한 다음 사용할 디바이스 이름을 입력합니다.

6. 볼륨 연결(Attach Volume)을 선택합니다.
7. 인스턴스에 연결하고 볼륨을 탑재합니다. 자세한 내용은 Amazon EBS 볼륨을 사용할 수 있도록 만들기 섹션을 참조하세요.

AWS CLI

인스턴스에 EBS 볼륨을 연결하려면

attach-volume 명령을 사용합니다. 다음 예제는 지정된 디바이스 이름을 사용하여 지정된 볼륨을 지정된 인스턴스에 연결합니다.

```
aws ec2 attach-volume \
    --volume-id vol-01234567890abcdef \
    --instance-id i-1234567890abcdef0 \
    --device /dev/sdf
```

PowerShell

인스턴스에 EBS 볼륨을 연결하려면

Add-EC2Volume cmdlet을 사용합니다. 다음 예제는 지정된 디바이스 이름을 사용하여 지정된 볼륨을 지정된 인스턴스에 연결합니다.

```
Add-EC2Volume `
    -VolumeId vol-01234567890abcdef `
    -InstanceId i-1234567890abcdef0 `
    -Device /dev/sdf
```

다중 연결을 사용하여 여러 EC2 인스턴스에 EBS 볼륨 연결

Amazon EBS 다중 연결을 사용하면 단일 프로비저닝된 IOPS SSD(io1 또는 io2) 볼륨을 동일한 가용 영역에 있는 여러 인스턴스에 연결할 수 있습니다. 여러 다중 연결 지원 볼륨을 인스턴스 또는 인스턴스 집합에 연결할 수 있습니다. 볼륨이 연결된 각 인스턴스는 공유된 볼륨에 대한 전체 읽기 및 쓰기 권한을 가집니다. 다중 연결을 사용하면 동시 쓰기 작업을 관리하는 애플리케이션에서 더 쉽게 더 높은 애플리케이션 가용성을 얻을 수 있습니다.

요금 및 결제

Amazon EBS 다중 연결 사용에 따르는 추가 비용은 없습니다. 프로비저닝된 IOPS SSD(io1 및 io2) 볼륨에 적용되는 표준 요금이 청구됩니다. 자세한 내용은 Amazon EBS 요금을 참조하세요.

내용

- 고려 사항 및 제한
- 다중 연결 Amazon EBS 볼륨의 성능
- Amazon EBS 볼륨에 대한 다중 연결 활성화
- Amazon EBS 볼륨에 대한 다중 연결 비활성화
- 다중 연결 지원 Amazon EBS 볼륨에서 NVMe 예약 사용

고려 사항 및 제한

- 다중 연결 지원 볼륨은 동일한 가용 영역에 있는 Nitro System에 구축된 최대 16개의 인스턴스에 연결할 수 있습니다.
- Linux 인스턴스에서는 다중 연결 사용 io1 및 io2 볼륨을 지원합니다. Windows 인스턴스에서는 다중 연결 사용 io2 볼륨만 지원합니다.
- 인스턴스에 연결할 수 있는 Amazon EBS 볼륨의 최대 수는 인스턴스 유형 및 인스턴스 크기에 따라 달라집니다. 자세한 내용은 인스턴스 볼륨 제한을 참조하세요.
- 다중 연결은 프로비저닝된 IOPS SSD(io1 및 io2) 볼륨에서만 지원됩니다.
- io1 볼륨에 대한 다중 연결은를 지원하는 모든 리전에서 사용할 수 있습니다io1.

  io2에 대한 다중 연결은 io2를 지원하는 모든 리전에서 사용할 수 있습니다.

  > **Note**
  >
  > 저렴한 비용으로 성능, 일관성 및 내구성을 높이려면 io2 볼륨을 사용하는 것이 좋습니다.

- XFS 및 EXT4와 같은 표준 파일 시스템은 EC2 인스턴스와 같은 여러 서버에서 동시에 액세스하도록 설계되지 않았습니다. 클러스터링된 파일 시스템을 사용하여 데이터가 프로덕션 워크로드에 대해 복원력과 신뢰성을 갖도록 해야 합니다.
- 다중 연결 활성화 io1 및 io2 볼륨은 I/O 펜싱을 지원합니다. I/O 차단 프로토콜은 데이터 일관성을 유지하기 위해 공유된 스토리지 환경에서 쓰기 액세스를 제어합니다. 애플리케이션은 데이터 일관성을 유지하기 위해 연결된 인스턴스에 대해 쓰기 순서를 제공해야 합니다. 자세한 내용은 다중 연결 지원 Amazon EBS 볼륨에서 NVMe 예약 사용 섹션을 참조하세요.
- 다중 연결 지원 볼륨은 부팅 볼륨으로 만들 수 없습니다.

- 다중 연결 지원 볼륨은 인스턴스당 하나의 블록 디바이스 매핑에 연결할 수 있습니다.
- 인스턴스 시작 중에는 Amazon EC2 콘솔 또는 RunInstances API를 사용하여 다중 연결을 활성화할 수 없습니다.
- Amazon EBS 인프라 계층에서 문제가 있는 다중 연결 지원 볼륨은 연결된 모든 인스턴스에서 사용할 수 없습니다. Amazon EC2 또는 네트워킹 계층의 문제가 있는 경우 연결된 인스턴스 일부만 영향을 받을 수 있습니다.
- 다음 표에는 다중 연결 사용 `io1` 및 `io2` 볼륨을 생성한 후의 볼륨 수정 지원 정보가 나와 있습니다.

	io2 볼륨	**io1** 볼륨
볼륨 유형 수정	✓ **	✓ **
볼륨 크기 수정	✓	✓
프로비저닝된 IOPS 수정	✓	✓
다중 연결 활성화	✓ *	✓ *
다중 연결 비활성화	✓ *	✓ *

* 볼륨이 인스턴스에 연결되어 있는 동안에는 다중 연결 기능을 활성화하거나 비활성화할 수 없습니다.

** 다중 연결 지원 볼륨`io2`에 대한 볼륨 유형 수정은 볼륨이 연결된 경우를 포함하여 `io1` 및 사이에서 어느 방향이든 지원됩니다. 다른 볼륨 유형 변경 사항은 지원되지 않습니다.

- 다중 연결 지원 볼륨은 마지막으로 연결된 인스턴스가 종료되고 종료 시 볼륨을 삭제하도록 해당 인스턴스가 구성된 경우 인스턴스 종료 시 삭제됩니다. 볼륨이 볼륨 블록 디바이스 매핑에서 종료 시 삭제 설정이 다른 여러 인스턴스에 연결되어 있는 경우 마지막으로 연결된 인스턴스의 블록 디바이스 매핑 설정에 따라 종료 시 삭제 동작이 결정됩니다.

  종료 시 삭제 동작을 예측 가능하도록 하려면 볼륨이 연결된 모든 인스턴스에 대해 종료 시 삭제를 활성화 또는 비활성화합니다. 자세한 내용은 인스턴스가 종료될 때 데이터 보존을 참조하세요.
- Amazon EBS 볼륨에 대한 CloudWatch 지표를 사용하여 다중 연결 지원 볼륨을 모니터링할 수 있습니다. 연결된 모든 인스턴스에서 데이터가 집계됩니다. 연결된 개별 인스턴스에 대한 지표를 모니터

링할 수는 없습니다. 자세한 내용은 Amazon EBS에 대한 Amazon CloudWatch 지표 섹션을 참조하세요.

다중 연결 Amazon EBS 볼륨의 성능

연결된 각 인스턴스는 최대 IOPS 성능을 볼륨의 최대 프로비저닝된 성능까지 구동할 수 있습니다. 그러나 연결된 모든 인스턴스의 전체 성능은 볼륨의 최대 프로비저닝된 성능을 초과할 수 없습니다. 연결된 인스턴스의 IOPS에 대한 수요가 볼륨의 프로비저닝된 IOPS보다 높으면 볼륨이 프로비저닝된 성능을 초과하지 않습니다.

예를 들어 `io2` 프로비저닝된 IOPS를 사용하여 `80,000` 다중 연결 지원 볼륨을 생성하고 최대 `40,000` IOPS를 지원하는 `m7g.large` 인스턴스와 최대 `60,000` IOPS를 지원하는 `r7g.12xlarge` 인스턴스에 연결한다고 가정합니다. 각 인스턴스는 최대 IOPS가 볼륨의 프로비저닝된 IOPS인 `80,000`보다 작기 때문에 최대 IOPS를 구동할 수 있습니다. 그러나 두 인스턴스 모두 볼륨에 대한 I/O를 동시에 구동하는 경우 결합된 IOPS는 볼륨의 프로비저닝된 성능인 `80,000` IOPS를 초과할 수 없습니다.

일관된 성능을 얻으려면 다중 연결 지원 볼륨의 섹터 전체에 대해 연결된 인스턴스에서 구동되는 I/O의 균형을 유지하는 것이 가장 좋습니다.

Amazon EC2 인스턴스 유형의 IOPS 성능에 대한 자세한 내용은 Amazon EC2 사용 설명서의 Amazon EBS 최적화 인스턴스 유형을 참조하세요.

Amazon EBS 볼륨에 대한 다중 연결 활성화

다중 연결 지원 볼륨은 다른 Amazon EBS 볼륨을 관리하는 것과 거의 동일한 방식으로 관리할 수 있습니다. 그러나 다중 연결 기능을 사용하려면 볼륨에 대해 이 기능을 활성화해야 합니다.

새 볼륨을 만들 때 다중 연결은 기본적으로 비활성화되어 있습니다. 볼륨을 생성할 때 다중 연결을 활성화할 수 있습니다.

생성 후 `io1` 및 `io2` 볼륨에 대해 다중 연결을 활성화할 수도 있지만 인스턴스에 연결되지 않은 경우에만 활성화할 수 있습니다.

볼륨에 대한 다중 연결을 활성화한 후에는 다른 EBS 볼륨을 연결하는 것과 동일한 방식으로 볼륨을 인스턴스에 연결할 수 있습니다. 자세한 내용은 Amazon EC2 인스턴스에 Amazon EBS 볼륨 연결 섹션을 참조하세요.

Console

볼륨 생성 중에 다중 연결을 활성화하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 볼륨을 선택합니다.
3. 볼륨 생성을 선택합니다.
4. 볼륨 유형에서 프로비저닝된 IOPS SSD(**io1**) 또는 프로비저닝된 IOPS SSD(**io2**)를 선택합니다.
5. 크기 및 IOPS의 경우 필요한 볼륨 크기와 프로비저닝할 IOPS 수를 선택합니다.
6. 가용 영역의 경우 인스턴스가 있는 가용 영역과 동일한 가용 영역을 선택합니다.
7. Amazon EBS 다중 연결(Amazon EBS Multi-Attach)에서 다중 연결 활성화(Enable Multi-Attach)를 선택합니다.
8. (선택 사항) 스냅샷 ID(Snapshot ID)에서 볼륨을 생성할 스냅샷을 선택합니다.
9. 볼륨의 암호화 상태를 설정합니다.

   선택한 스냅샷이 암호화되거나 계정에 기본적으로 암호화가 활성화되어 있으면 암호화가 자동으로 사용되며 비활성화할 수 없습니다. 볼륨 암호화에 사용할 KMS 키를 선택할 수 있습니다.

   선택한 스냅샷이 암호화되지 않았으며 계정이 기본적으로 암호화를 사용하도록 설정되어 있지 않은 경우 암호화는 선택 사항입니다. 볼륨을 암호화하려면 암호화(Encryption)에서 이 볼륨 암호화(Encrypt this volume)를 선택한 다음 볼륨 암호화에 사용할 KMS 키를 선택합니다.

   암호화된 볼륨은 Amazon EBS 암호화를 지원하는 인스턴스에만 연결할 수 있습니다. 자세한 내용은 Amazon EBS 암호화 섹션을 참조하세요.
10. (선택 사항) 볼륨에 사용자 정의 태그를 할당하려면 태그(Tags) 섹션에서 태그 추가(Add tag)를 선택한 다음 태그 키 및 값 페어를 입력합니다.
11. 볼륨 생성을 선택합니다.

생성 후 다중 연결을 활성화하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 볼륨을 선택합니다.
3. 볼륨을 선택한 후 작업(Actions), 볼륨 수정(Modify volume)을 선택합니다.
4. Amazon EBS 다중 연결(Amazon EBS Multi-Attach)에서 다중 연결 활성화(Enable Multi-Attach)를 선택합니다.

5. 수정을 선택합니다.

AWS CLI

볼륨 생성 중에 다중 연결을 활성화하려면

`--multi-attach-enabled` 옵션과 함께 create-volume 명령을 사용합니다.

```
aws ec2 create-volume \
    --volume-type io2 \
    --multi-attach-enabled \
    --size 100 \
    --iops 2000 \
    --region us-west-2 \
    --availability-zone us-west-2b
```

생성 후 다중 연결을 활성화하려면

`--multi-attach-enabled` 옵션과 함께 modify-volume 명령을 사용합니다.

```
aws ec2 modify-volume \
    --volume-id vol-01234567890abcdef \
    --multi-attach-enabled
```

PowerShell

볼륨 생성 중에 다중 연결을 활성화하려면

`-MultiAttachEnabled` 파라미터와 함께 New-EC2Volume cmdlet을 사용합니다.

```
New-EC2Volume `
    -VolumeType io2 `
    -MultiAttachEnabled $true `
    -Size 100 `
    -Iops 2000 `
    -Region us-west-2 `
    -AvailabilityZone us-west-2b
```

생성 후 다중 연결을 활성화하려면

`-MultiAttachEnabled` 파라미터와 함께 Edit-EC2Volume cmdlet을 사용합니다.

```
Edit-EC2Volume `
    -VolumeId vol-01234567890abcdef `
    -MultiAttachEnabled $true
```

Amazon EBS 볼륨에 대한 다중 연결 비활성화

인스턴스에 연결되지 않은 경우에만 `io1` 또는 `io2` 볼륨에 대한 다중 연결을 비활성화할 수 있습니다.

Console

생성 후 다중 연결을 비활성화하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 볼륨을 선택합니다.
3. 볼륨을 선택한 후 작업(Actions), 볼륨 수정(Modify volume)을 선택합니다.
4. Amazon EBS 다중 연결(Amazon EBS Multi-Attach)에서 다중 연결 활성화(Enable Multi-Attach) 선택을 취소합니다.
5. 수정을 선택합니다.

AWS CLI

생성 후 다중 연결을 비활성화하려면

`-no-multi-attach-enabled` 옵션과 함께 modify-volume 명령을 사용합니다.

```
aws ec2 modify-volume \
    --volume-id vol-01234567890abcdef \
    --no-multi-attach-enabled
```

PowerShell

생성 후 다중 연결을 비활성화하려면

`-MultiAttachEnabled` 파라미터와 함께 Edit-EC2Volume cmdlet을 사용합니다.

```
Edit-EC2Volume `
    -VolumeId vol-01234567890abcdef `
```

```
    -MultiAttachEnabled $false
```

다중 연결 지원 Amazon EBS 볼륨에서 NVMe 예약 사용

다중 연결 활성화 `io1` 및 `io2` 볼륨은 업계 표준 스토리지 펜싱 프로토콜 세트인 NVMe 예약을 지원합니다. 이러한 프로토콜을 사용하면 여러 인스턴스에서 공유 볼륨으로의 액세스를 제어하고 조정하는 예약을 생성하고 관리할 수 있습니다. 공유 스토리지 애플리케이션은 예약을 사용하여 데이터를 일관되게 유지합니다.

내용

- 요구 사항
- NVMe 예약에 대한 지원 활성화
- 지원되는 NVMe 예약 명령
- 요금

요구 사항

NVMe 예약은 다중 연결 활성화 `io1` 및 `io2` 볼륨에서 지원됩니다. 다중 연결 지원 볼륨은 Nitro System에 구축된 인스턴스에 연결할 수 있습니다.

NVMe 예약은 다음 운영 체제에서 지원됩니다.

- SUSE Linux Enterprise 12 SP3 이상
- RHEL 8.3 이상
- Amazon Linux 2 이상
- Windows Server 2016 이상

> **Note**
>
> 2023.09.13 이후 지원되는 Windows Server AMI의 경우 필수 NVMe 드라이버가 포함되어 있습니다. 이전 AMI의 경우 NVMe 드라이버 버전 1.5.0 이상으로 업데이트해야 합니다. 자세한 내용은 AWS NVMe 드라이버를 참조하세요.

EC2Launch v2를 사용하여 디스크를 초기화하는 경우 버전 2.0.1521 이상으로 업그레이드해야 합니다. 자세한 내용은 EC2Launch v2 에이전트 사용을 참조하세요.

NVMe 예약에 대한 지원 활성화

NVMe 예약에 대한 지원은 2023년 9월 18일 이후에 생성된 모든 다중 연결 지원 io2 볼륨과 2026년 10월 1일 이후에 생성된 모든 다중 연결 지원 io1 볼륨에 대해 기본적으로 활성화됩니다.

2023년 9월 18일 이전에 생성된 기존 io2 볼륨 또는 2026년 10월 1일 이전에 생성된 기존 io1 볼륨에 대한 NVMe 예약을 지원하려면 볼륨에서 모든 인스턴스를 분리한 다음 필요한 인스턴스를 다시 연결해야 합니다. 모든 인스턴스를 분리한 후 이루어진 모든 연결에는 NVMe 예약이 활성화됩니다.

지원되는 NVMe 예약 명령

Amazon EBS는 다음과 같은 NVMe 예약 명령을 지원합니다.

예약 등록

예약 키를 등록, 등록 취소 또는 교체합니다. 등록 키는 인스턴스를 식별하고 인증하는 데 사용됩니다. 예약 키를 볼륨에 등록하면 인스턴스와 볼륨 간의 연결이 생성됩니다. 인스턴스를 볼륨에 등록해야 해당 인스턴스가 예약을 획득할 수 있습니다.

예약 획득

볼륨에 대한 예약을 획득하고, 네임스페이스에 보관된 예약을 선점하고, 볼륨에 대한 예약을 중단합니다. 획득할 수 있는 예약 유형은 다음과 같습니다.

- 독점 예약 작성
- 독점 액세스 예약
- 독점 작성 - 등록자 전용 예약
- 독점 액세스 - 등록자 전용 예약
- 독점 작성 - 모든 등록자 예약
- 독점 액세스 - 모든 등록자 예약

예약 릴리스

볼륨에 대해 보류된 예약을 해제하거나 취소합니다.

예약 보고서

볼륨의 등록 및 예약 상태를 설명합니다.

요금

다중 연결 활성화 및 사용에 따르는 추가 비용은 없습니다.

Amazon EBS 볼륨을 사용할 수 있도록 만들기

Amazon EBS 볼륨은 인스턴스에 연결하면 블록 디바이스로 표시됩니다. 볼륨을 원하는 파일 시스템으로 포맷한 다음 마운트합니다. EBS 볼륨을 사용할 수 있게 만들면 다른 볼륨과 동일한 방식으로 액세스할 수 있습니다. 이 파일 시스템에 작성된 모든 데이터가 EBS 볼륨에 작성되고 해당 디바이스를 사용하는 애플리케이션도 그대로 적용됩니다.

다른 볼륨을 생성할 때 기준으로 사용하거나 백업을 목적으로 EBS 볼륨의 스냅샷을 생성할 수 있습니다. 자세한 내용은 Amazon EBS 스냅샷 섹션을 참조하세요.

사용 준비 중인 EBS 볼륨이 2TiB보다 크면 GPT 파티션 체계를 사용하여 전체 볼륨에 액세스해야 합니다. 자세한 내용은 Amazon EBS 볼륨 제약 조건 섹션을 참조하세요.

Linux 인스턴스

연결된 볼륨 포맷 및 탑재

루트 디바이스용 EBS 볼륨이 있는 EC2 인스턴스가 있으며, /dev/xvda, 방금 /dev/sdf를 이용해 인스턴스를 빈 EBS 볼륨에 연결했다고 가정합시다. 다음 절차에 따라, 새로 연결한 볼륨을 사용할 수 있게 만드세요.

EBS 볼륨을 Linux에서 포맷 및 탑재

1. SSH로 인스턴스에 연결합니다. 자세한 내용은 Linux 인스턴스에 연결을 참조하세요.
2. 디바이스는 블록 디바이스 매핑에 지정한 것과는 다른 디바이스 이름으로 인스턴스에 연결할 수 있습니다. 자세한 내용은 Linux 인스턴스의 디바이스 이름을 참조하세요. lsblk 명령을 사용하면 사용 가능한 디스크 디바이스 및 마운트 포인트(해당하는 경우)가 표시되어 사용 가능한 올바른 디바이스 이름을 결정하는 데 도움을 받을 수 있습니다. lsblk 명령의 출력에서는 전체 디바이스 경로 중 맨 앞에 /dev/가 생략됩니다.

   다음은 EBS 볼륨이 NVMe 블록 디바이스로 표시되는 Nitro System에 구축된 인스턴스의 예시 출력입니다. 루트 디바이스는 nvme0n1p1 및 nvme0n1p128이라는 두 개의 파티션이 있는 /dev/nvme0n1입니다. 연결된 볼륨은 파티션이 없고 아직 탑재되지 않은 /dev/nvme1n1입니다.

```
[ec2-user ~]$ lsblk
NAME          MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
nvme1n1       259:0    0  10G  0 disk
nvme0n1       259:1    0   8G  0 disk
-nvme0n1p1    259:2    0   8G  0 part /
-nvme0n1p128  259:3    0   1M  0 part
```

다음은 T2 인스턴스의 예시 출력입니다. 루트 디바이스는 xvda1이라는 파티션이 하나 있는 /dev/xvda입니다. 연결된 볼륨은 파티션이 없고 아직 탑재되지 않은 /dev/xvdf입니다.

```
[ec2-user ~]$ lsblk
NAME    MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
xvda    202:0    0    8G  0 disk
-xvda1  202:1    0    8G  0 part /
xvdf    202:80   0   10G  0 disk
```

3. 볼륨에 파일 시스템이 있는지 확인합니다. 새 볼륨은 원시 블록 디바이스이므로 볼륨을 탑재하고 사용하기 전에 해당 볼륨에서 파일 시스템을 생성해야 합니다. 스냅샷에서 생성된 볼륨에는 이미 파일 시스템이 있을 수 있습니다. 기존 파일 시스템 위에 새 파일 시스템을 생성하면 해당 작업으로 데이터가 덮어쓰기됩니다.

   볼륨에 파일 시스템이 있는지 여부를 확인하려면 다음 방법 중 하나 또는 모두를 사용하세요.

   - file -s 명령을 사용하면 파일 시스템 유형 등의 특정 디바이스 정보를 확인할 수 있습니다. 다음 예시 출력에서와 같이 출력에 data만 표시된다면, 디바이스에는 파일 시스템이 없습니다.

     ```
     [ec2-user ~]$ sudo file -s /dev/xvdf
     /dev/xvdf: data
     ```

     디바이스에 파일 시스템이 있다면, 명령은 파일 시스템 유형에 관한 정보를 표시합니다. 예를 들어 다음 출력은 XFS 파일 시스템이 있는 루트 디바이스를 표시합니다.

     ```
     [ec2-user ~]$ sudo file -s /dev/xvda1
     /dev/xvda1: SGI XFS filesystem data (blksz 4096, inosz 512, v2 dirs)
     ```

   - lsblk -f 명령을 사용하여 인스턴스에 연결된 모든 디바이스 관련 정보를 가져옵니다.

     ```
     [ec2-user ~]$ sudo lsblk -f
     ```

     예를 들어, 다음 출력은 인스턴스에 연결된 3개의 디바이스(nvme1n1, nvme0n1 및 nvme2n1)를 보여줍니다. 첫 번째 열에는 디바이스와 해당 파티션이 나열됩니다. FSTYPE 열에는 각 디바이스의 파일 시스템 유형이 표시됩니다. 특정 디바이스에 대한 열이 비어 있으면 디바이스에 파일 시스템이 없음을 의미합니다. 이 예에서 디바이스 nvme1n1과 nvme0n1 디바이스의 파티션 nvme0n1p1은 모두 XFS 파일 시스템을 사용하여 포맷되어 있지만, 디바이스 nvme2n1과 디바이스 nvme0n1의 파티션 nvme0n1p128에는 파일 시스템이 없습니다.

```
NAME  FSTYPE LABEL UUID      MOUNTPOINT
nvme1n1         xfs  7f939f28-6dcc-4315-8c42-6806080b94dd
nvme0n1
##nvme0n1p1 xfs     / 90e29211-2de8-4967-b0fb-16f51a6e464c         /
##nvme0n1p128
nvme2n1
```

이러한 명령의 출력에 디바이스에 파일 시스템이 없다고 표시된 경우 생성해야 합니다.

4. (선택 사항) 이전 단계에서 디바이스에 파일 시스템이 있음을 발견했다면, 이 단계는 생략하세요. 빈 볼륨이 있다면 mkfs -t 명령을 이용해 볼륨에서 파일 시스템을 생성하세요.

> **Warning**
>
> 이미 데이터가 있는 볼륨(예: 스냅샷에서 생성된 볼륨)을 탑재하는 경우, 이 명령을 사용하지 마세요. 아니면 볼륨을 포맷하여 기존 데이터를 삭제합니다.

```
[ec2-user ~]$ sudo mkfs -t xfs /dev/xvdf
```

`mkfs.xfs`이 발견되지 않는 오류가 발생하는 경우 다음 명령을 사용해 XFS 도구를 설치하고 이전 명령을 반복합니다.

```
[ec2-user ~]$ sudo yum install xfsprogs
```

5. mkdir 명령을 사용하여 볼륨에서 사용할 탑재 지점 디렉터리를 생성합니다. 마운트 포인트는 파일 시스템 트리에 볼륨이 위치하고 볼륨을 마운트한 후 파일을 읽고 쓰는 위치입니다. 다음은 `/data`라는 이름의 디렉터리를 생성하는 예제입니다.

```
[ec2-user ~]$ sudo mkdir /data
```

6. 이전 단계에서 생성한 탑재 지점 디렉터리에서 볼륨이나 파티션을 탑재합니다.

   볼륨에 파티션이 없는 경우 다음 명령을 사용하여 전체 볼륨을 탑재할 디바이스 이름을 지정합니다.

```
[ec2-user ~]$ sudo mount /dev/xvdf /data
```

볼륨에 파티션이 있는 경우 다음 명령을 사용하여 파티션을 탑재할 파티션 이름을 지정합니다.

```
[ec2-user ~]$ sudo mount /dev/xvdf1 /data
```

7. 새 볼륨 마운트의 파일 권한을 검토하여 사용자 및 애플리케이션이 볼륨에 기록할 수 있는지 확인합니다. 파일 권한에 대한 자세한 내용은 Linux Documentation Project에서 File security 단원을 참조하십시오.
8. 탑재 지점은 인스턴스를 재부팅하면 자동으로 보존되지 않습니다. 재부팅 후에도 이 EBS 볼륨을 자동으로 탑재하고 싶다면, 다음 절차를 참조하세요.

재부팅 후에도 연결된 볼륨을 자동으로 탑재

시스템을 재부팅할 때마다 연결된 EBS 볼륨을 탑재하려면, 디바이스에 대한 항목을 /etc/fstab 파일에 추가합니다.

/dev/xvdf에 있는 /etc/fstab 같은 디바이스 이름을 사용할 수 있습니다. 하지만 디바이스의 128비트 UUID(Universally Unique Identifier)를 사용할 것을 권장합니다. 디바이스 이름은 바꿀 수 있지만, UUID는 파티션 수명이 다할 때까지 유지됩니다. UUID를 사용하면 하드웨어 재구성 후 시스템을 부팅할 수 없게 되는 경우가 줄어듭니다. 자세한 내용은 Amazon EBS 볼륨을 NVMe 디바이스 이름에 매핑 섹션을 참조하세요.

재부팅 후 연결된 볼륨을 자동으로 탑재하는 방법

1. (선택 사항) 수정 도중 실수로 이 파일이 손상되거나 삭제되는 경우에 대비하여 /etc/fstab 파일의 백업을 생성합니다.

   ```
   [ec2-user ~]$ sudo cp /etc/fstab /etc/fstab.orig
   ```

2. blkid 명령을 사용하여 디바이스의 UUID를 찾습니다. 재부팅 후 탑재할 장치의 UUID를 기록해 둡니다. 다음 단계에서 필요합니다.

   예를 들어 다음 명령은 인스턴스에 2개의 디바이스가 탑재되어 있음을 보여주며 두 디바이스 모두의 UUID를 보여줍니다.

   ```
   [ec2-user ~]$ sudo blkid
   /dev/xvda1: LABEL="/" UUID="ca774df7-756d-4261-a3f1-76038323e572" TYPE="xfs"
    PARTLABEL="Linux" PARTUUID="02dcd367-e87c-4f2e-9a72-a3cf8f299c10"
   /dev/xvdf: UUID="aebf131c-6957-451e-8d34-ec978d9581ae" TYPE="xfs"
   ```

Ubuntu 18.04의 경우 lsblk 명령을 사용합니다.

```
[ec2-user ~]$ sudo lsblk -o +UUID
```

3. nano 또는 vim과 같은 텍스트 편집기를 사용하여 /etc/fstab 파일을 엽니다.

```
[ec2-user ~]$ sudo vim /etc/fstab
```

4. 다음 항목을 /etc/fstab에 추가해 디바이스를 지정된 탑재 지점에 탑재합니다. 필드는 blkid가 반환하는 UUID 값(또는 Ubuntu 18.04의 경우 lsblk), 탑재 지점, 파일 시스템, 권장하는 파일 시스템 탑재 옵션입니다. 필수 필드에 대한 자세한 내용을 보려면 man fstab를 실행하여 fstab 매뉴얼을 엽니다.

   다음 예시에서는 UUID를 사용하여 디바이스를 탑재 지점aebf131c-6957-451e-8d34-ec978d9581ae에 탑재/data하고 xfs 파일 시스템을 사용합니다. 또한 defaults 및 nofail 플래그를 사용합니다. 파일 시스템이 덤프되지 않도록 0를 지정하고 루트가 아닌 디바이스임을 나타내2도록를 지정합니다.

```
UUID=aebf131c-6957-451e-8d34-ec978d9581ae  /data  xfs  defaults,nofail  0  2
```

> **Note**
>
> (볼륨을 다른 인스턴스로 옮긴 후 등의 상황에서) 이 볼륨을 연결하지 않고 인스턴스를 부팅했다면, nofail 탑재 옵션을 이용해 볼륨 탑재 시 오류가 있더라도 인스턴스를 부팅할 수 있습니다. 16.04 이전의 Ubuntu 버전을 포함하는 Debian 계열 시스템에서는 nobootwait 탑재 옵션도 추가해야 합니다.

5. 항목이 제대로 작동하는지 확인하기 위해, 다음 명령을 실행해 디바이스 탑재를 해제하고 /etc/fstab에서 모든 파일 시스템을 탑재합니다. 오류가 없다면 /etc/fstab 파일에 문제가 없다는 뜻이며, 파일 시스템은 재부팅 후 자동으로 탑재됩니다.

```
[ec2-user ~]$ sudo umount /data
[ec2-user ~]$ sudo mount -a
```

   오류 메시지가 표시된다면, 파일의 오류를 처리하세요.

> **Warning**
>
> /etc/fstab 파일에서 오류가 발생하면 시스템이 부팅되지 않을 수 있습니다. /etc/fstab 파일에서 오류가 발생한 시스템을 종료하지 마십시오.

/etc/fstab의 오류 수정 방법을 모르며 이 절차의 첫 번째 단계에서 백업 파일을 만들었다면, 다음 명령을 이용해 백업 파일에서 복원을 진행할 수 있습니다.

```
[ec2-user ~]$ sudo mv /etc/fstab.orig /etc/fstab
```

Windows 인스턴스

다음과 같은 방법 중 하나를 사용하여 Windows 인스턴스에서 볼륨을 사용할 수 있도록 만듭니다.

PowerShell

원시 파티션이 있는 모든 EBS 볼륨을 Windows PowerShell에서 사용

1. 원격 데스크톱을 사용하여 Windows 인스턴스에 로그인합니다. 자세한 내용은 Windows 인스턴스에 연결을 참조하세요.
2. 작업 표시줄에서 시작 메뉴를 열고 Windows PowerShell을 선택합니다.
3. 열린 PowerShell 프롬프트에서 제공된 일련의 Windows PowerShell 명령을 사용합니다. 이 스크립트는 기본값으로 다음 작업을 수행합니다.

   1. ShellHWDetection 서비스를 중지합니다.
   2. 파티션 스타일이 원시 디스크인 디스크를 열거합니다.
   3. 디스크 및 파티션 유형이 지원할 최대 크기에 걸쳐 있는 새 파티션을 만듭니다.
   4. 사용 가능한 드라이브 문자를 배정합니다.
   5. 지정된 파일 시스템 레이블을 사용하여 파일 시스템을 NTFS로 포맷합니다.
   6. ShellHWDetection 서비스를 다시 시작합니다.

```
Stop-Service -Name ShellHWDetection
```

```
Get-Disk | Where PartitionStyle -eq 'raw' | Initialize-Disk -PartitionStyle MBR
 -PassThru | New-Partition -AssignDriveLetter -UseMaximumSize | Format-Volume -
FileSystem NTFS -NewFileSystemLabel "Volume Label" -Confirm:$false
Start-Service -Name ShellHWDetection
```

DiskPart command line tool

DiskPart 명령줄 도구를 통해 사용 가능한 EBS 볼륨을 만들기

1. 원격 데스크톱을 사용하여 Windows 인스턴스에 로그인합니다. 자세한 내용은 Windows 인스턴스에 연결을 참조하세요.
2. 사용할 디스크 번호를 확인합니다.
   1. 시작 메뉴를 열고 Windows PowerShell을 선택합니다.
   2. 사용 가능한 디스크 목록을 검색하는 `Get-Disk` Cmdlet을 사용합니다.
   3. 명령 출력에서 사용 가능한 디스크에 해당하는 숫자(Number)를 기록합니다.
3. DiskPart 명령을 실행할 스크립트 파일을 작성합니다.
   1. 시작 메뉴를 열고 파일 탐색기(File Explorer)를 선택합니다.
   2. 스크립트 파일을 저장할 디렉터리(예: C:\)로 이동합니다.
   3. 폴더 내의 빈 공간을 선택하여(마우스 오른쪽 버튼 클릭) 대화 상자를 열고 새로 만들기 위에 커서를 놓고 컨텍스트 메뉴에 액세스한 다음 텍스트 문서를 선택합니다.
   4. 텍스트 파일을 `diskpart.txt`로 지정합니다.
4. 스크립트 파일에 다음 명령을 추가합니다. 디스크 번호, 파티션 유형, 볼륨 레이블 및 드라이브 문자를 수정해야 할 수 있습니다. 이 스크립트는 기본값으로 다음 작업을 수행합니다.
   1. 수정할 디스크 1을 선택합니다.
   2. 기본 부트 레코드(MBR) 파티션 구조를 사용하도록 볼륨을 구성합니다.
   3. 볼륨을 NTFS 볼륨으로 포맷합니다.
   4. 볼륨 레이블을 설정합니다.
   5. 볼륨에 드라이브 문자를 할당합니다.

> **Warning**
>
> 이미 데이터가 있는 볼륨을 마운트하는 경우 볼륨을 재포맷하지 않아야 기존 데이터가 삭제되지 않습니다.

```
select disk 1
attributes disk clear readonly
online disk noerr
convert mbr
create partition primary
format quick fs=ntfs label="volume_label"
assign letter="drive_letter"
```

자세한 내용은 DiskPart 구문 및 파라미터를 참조하세요.

5. 명령 프롬프트를 열고 스크립트가 있는 폴더로 이동하고 다음 명령을 실행하여 지정된 디스크에서 볼륨을 사용할 수 있도록 합니다.

```
C:\> diskpart /s diskpart.txt
```

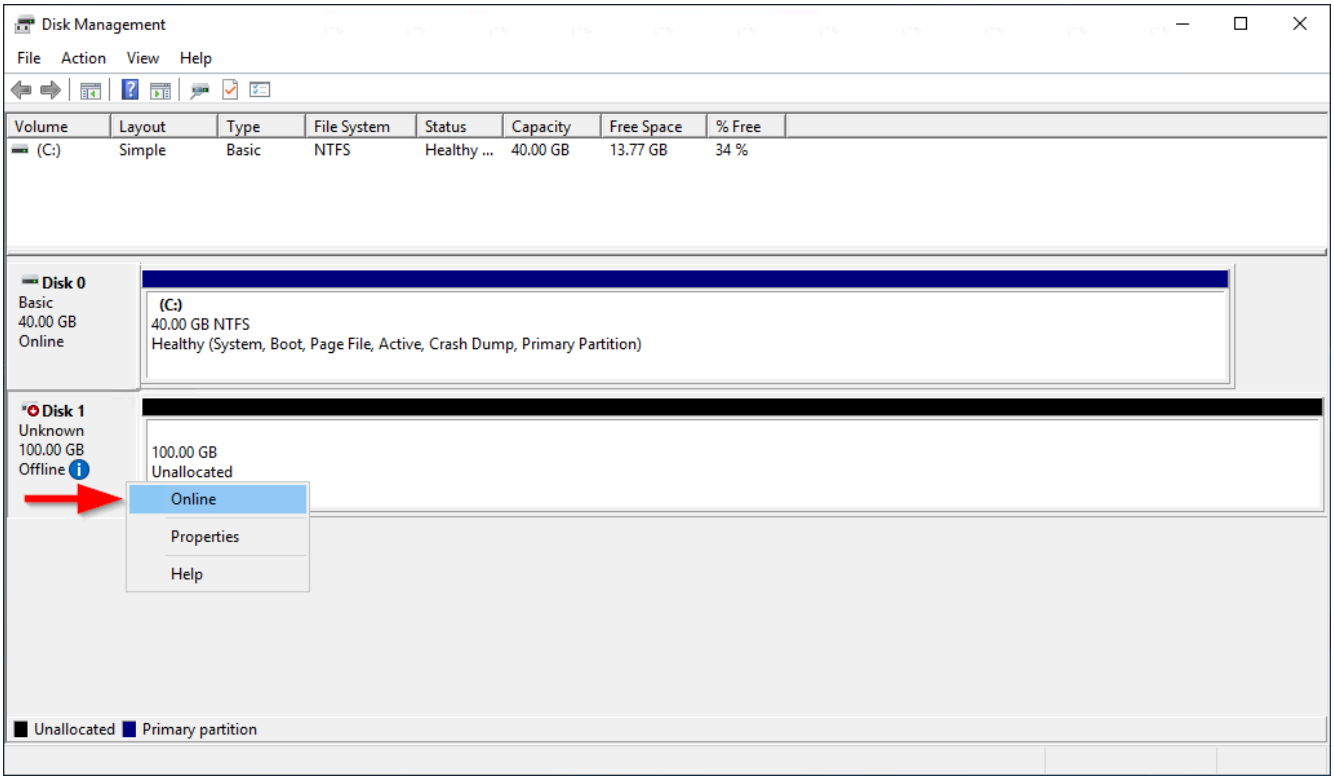
Disk Management utility

디스크 관리 유틸리티를 통해 사용 가능한 EBS 볼륨 만들기

1. 원격 데스크톱을 사용하여 Windows 인스턴스에 로그인합니다. 자세한 내용은 Windows 인스턴스에 연결을 참조하세요.
2. 디스크 관리 유틸리티를 시작합니다. 작업 표시줄에서 마우스 오른쪽 버튼을 클릭하여 Windows 로고에 대한 컨텍스트 메뉴를 열고 디스크 관리(Disk Management)를 선택합니다.

> **Note**
>
> Windows Server 2008에서는 시작(Start), 관리 도구(Administrative Tools), 컴퓨터 관리(Computer Management), 디스크 관리(Disk Management)를 선택합니다.

3. 볼륨을 온라인 상태로 전환합니다. 아래쪽의 왼쪽 창에서 마우스 오른쪽 버튼을 클릭하여 EBS 볼륨용 디스크에 대한 컨텍스트 메뉴를 엽니다. 온라인을 선택합니다.

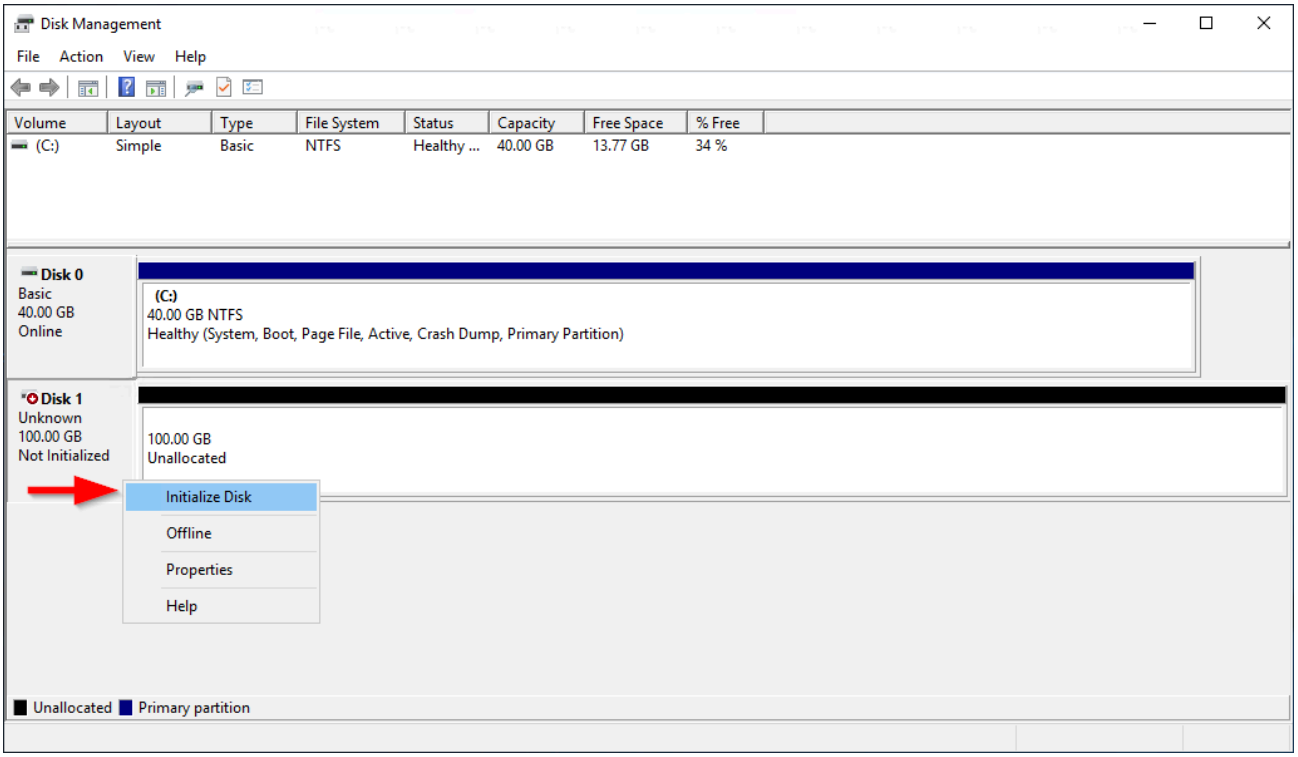


4. (조건) 디스크가 초기화되지 않은 경우 디스크를 초기화해야 사용할 수 있습니다. 디스크가 이미 초기화된 경우 이 단계를 건너뜁니다.

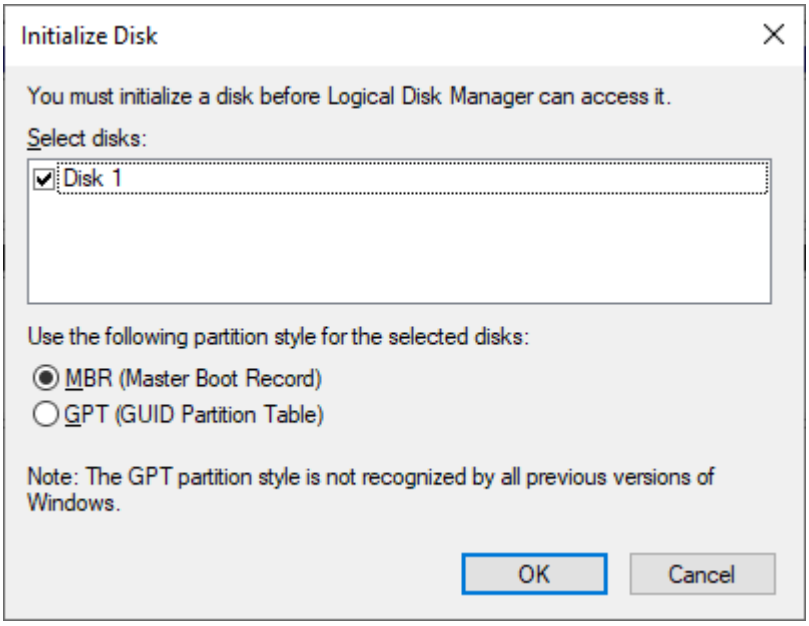
> **Warning**
>
> 이미 데이터가 있는 볼륨을 마운트하는 경우(예: 퍼블릭 데이터 세트 또는 스냅샷에서 생성된 볼륨), 볼륨을 다시 포맷하지 말고 기존 데이터를 삭제하세요.

디스크가 초기화되어 있지 않으면 다음과 같이 초기화를 수행하세요.

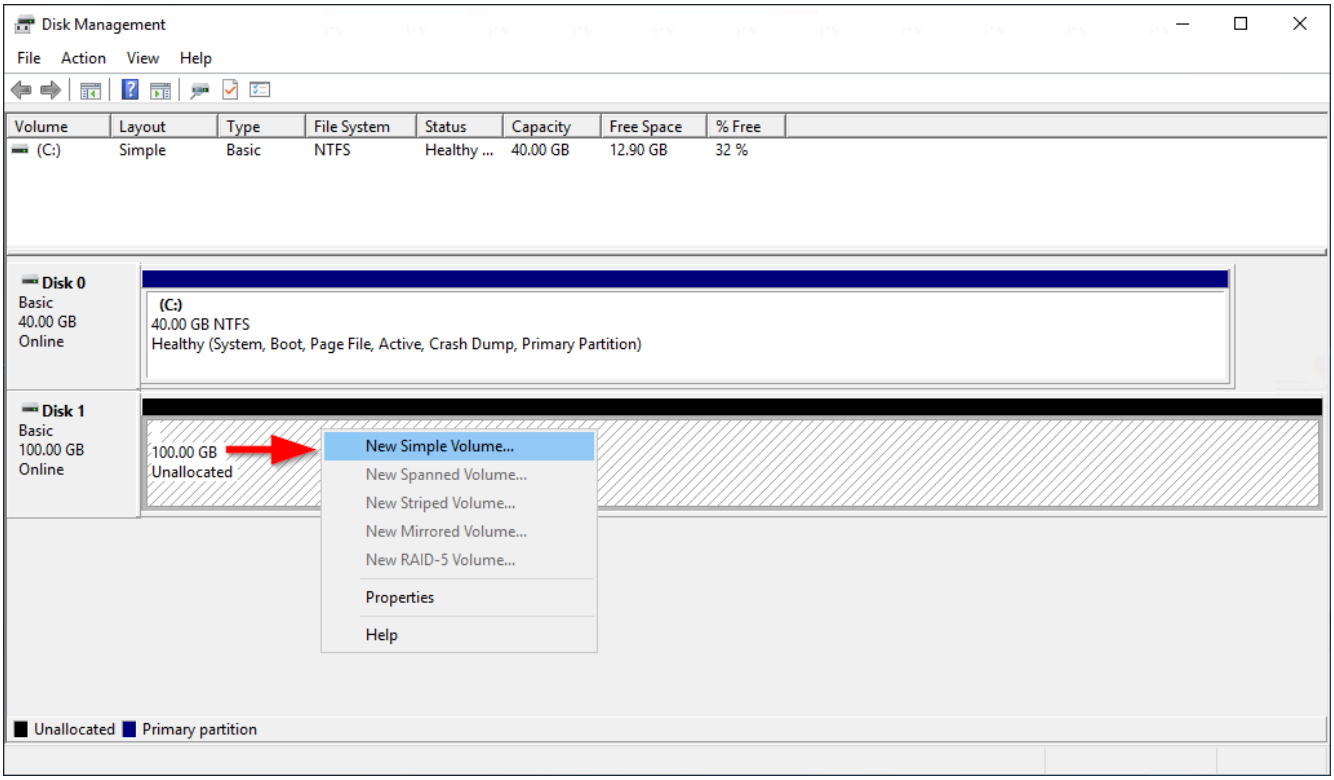
1. 왼쪽 창에서 마우스 오른쪽 버튼을 클릭하여 디스크에 대한 컨텍스트 메뉴를 열고 디스크 초기화(Initialize Disk)를 선택합니다.



2. 디스크 초기화(Initialize Disk) 대화 상자에서 파티션 스타일을 선택하고 확인(OK)을 선택합니다.



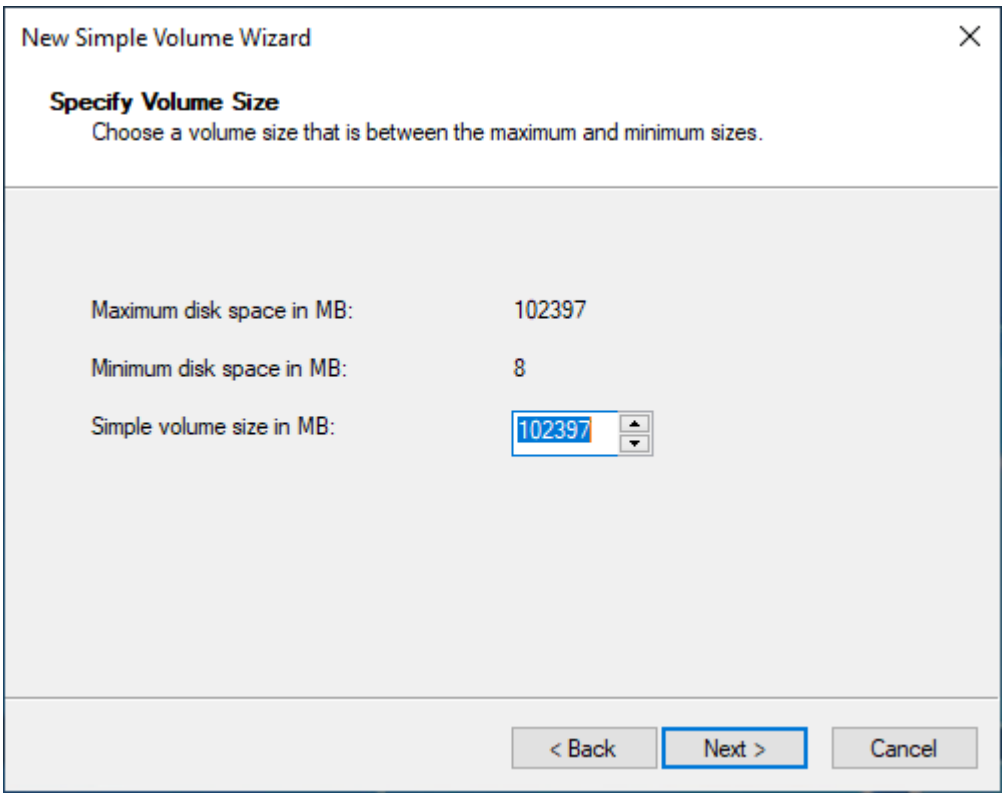
5. 오른쪽 창에서 마우스 오른쪽 버튼을 클릭하여 디스크에 대한 컨텍스트 메뉴를 열고 새 단순 볼륨(New Simple Volume)을 선택합니다.



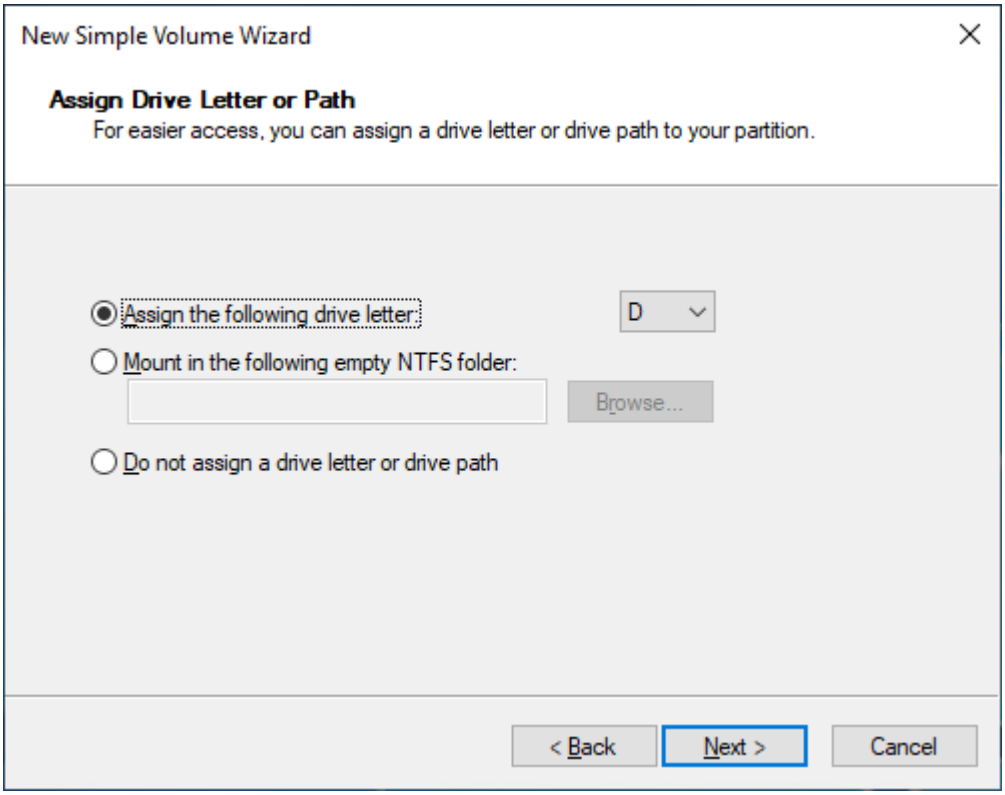
6. 새로운 단순 볼륨 마법사(New Simple Volume Wizard)에서 다음(Next)을 선택합니다.



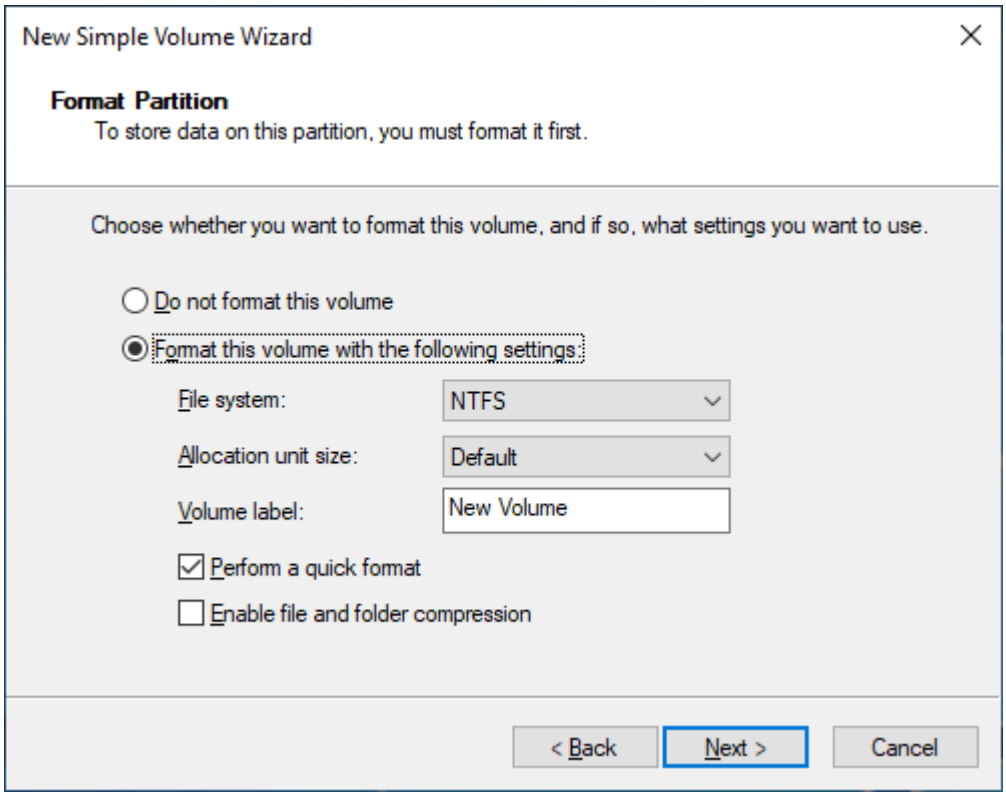
7. 기본 최대값을 변경하려면 MB 크기의 단순 볼륨(Simple volume size in MB)을 지정한 후 다음(Next)을 선택합니다.



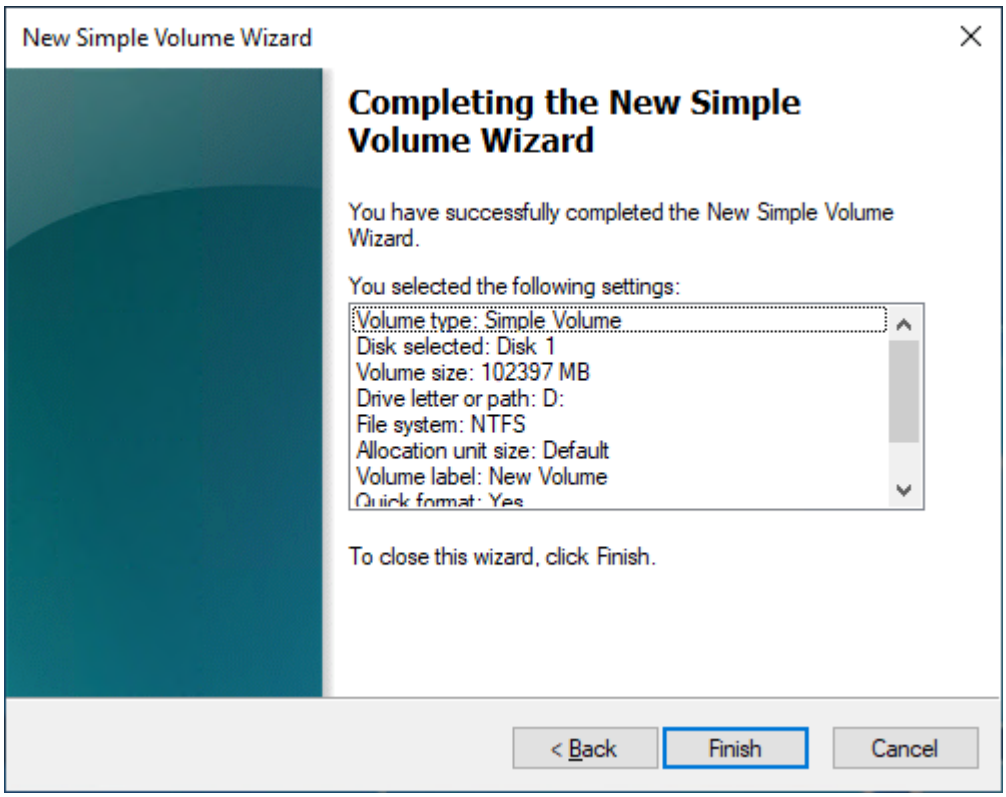
8. 필요한 경우 다음 드라이브 문자 할당(Assign the following drive letter) 드롭다운에서 권장 드라이브 문자를 지정한 후 다음(Next)을 선택합니다.



9. 볼륨 레이블(Volume Label)을 지정하고 필요에 따라 기본 설정을 조정한 후 다음(Next)을 선택합니다.



10. 설정을 검토한 다음 완료(Finish)를 선택하여 수정 사항을 적용하고 새 단순 볼륨 만들기 마법사를 닫습니다.



Amazon EBS 볼륨에 대한 정보 보기

EBS 볼륨에 대한 설명이 포함된 정보를 볼 수 있습니다. 예를 들어 특정 리전에 있는 모든 볼륨에 대한 정보를 보거나 크기, 볼륨 유형, 볼륨 암호화 여부, 볼륨 암호화에 사용된 KMS 키, 볼륨이 연결된 특정 인스턴스 등 단일 볼륨에 대한 자세한 정보를 볼 수 있습니다.

사용 가능한 디스크 공간 등 EBS 볼륨에 대한 추가 정보를 인스턴스의 운영 체제에서 가져올 수 있습니다.

내용

- 볼륨 정보 보기
- 볼륨 상태
- 볼륨 지표 보기
- 여유 디스크 공간 보기

볼륨 정보 보기

EBS 볼륨에 대한 정보를 볼 수 있습니다.

Console

볼륨에 대한 정보를 보려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 볼륨을 선택합니다.
3. 목록을 줄이기 위해 태그 및 볼륨 속성을 사용하여 볼륨을 필터링할 수 있습니다. 필터 필드를 선택하고 태그 또는 볼륨 속성을 선택한 다음 필터 값을 선택합니다.
4. 볼륨에 대한 자세한 정보를 확인하려면 해당 ID를 선택합니다.

인스턴스에 연결된 EBS 볼륨을 보려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. 인스턴스를 선택합니다.
4. 스토리지(Storage) 탭의 블록 디바이스(Block devices) 섹션에는 인스턴스에 연결된 볼륨을 나열됩니다. 특정 볼륨에 대한 정보를 보려면 볼륨 ID(Volume ID) 열에서 해당 ID를 선택합니다.

Amazon EC2 Global View

Amazon EC2 Global View를 사용하여 AWS 계정이 활성화된 모든 리전의 볼륨을 볼 수 있습니다.

모든 리전의 EBS 볼륨에 대한 요약을 보려면

1. https://console.aws.amazon.com/ec2globalview/home에서 Amazon EC2 Global View 콘솔을 엽니다.
2. 리전 탐색기 탭의 요약에서 볼륨 수와 리전 수를 포함한 볼륨 리소스 수를 확인합니다. 밑줄이 그어진 텍스트를 선택하여 볼륨 수가 리전에 어떻게 분산되는지 확인합니다.
3. 전역 검색 탭에서 클라이언트 필터 리소스 유형 = 볼륨을 선택합니다. 리전 또는 태그를 지정하여 결과를 추가로 필터링할 수 있습니다.

AWS CLI

EBS 볼륨에 대한 정보를 보려면

describe-volumes 명령을 사용합니다. 다음 예제는 현재 리전의 볼륨 수를 계산합니다.

```
aws ec2 describe-volumes --query "length(Volumes[*])"
```

다음 예제는 지정된 인스턴스에 연결된 볼륨을 나열합니다.

```
aws ec2 describe-volumes \
    --filters "Name=attachment.instance-id,Values=i-1234567890abcdef0" \
    --query Volumes[*].VolumeId \
    --output text
```

다음 예제는 지정된 볼륨을 설명합니다.

```
aws ec2 describe-volumes --volume-ids vol-01234567890abcdef
```

다음은 예제 출력입니다.

```
{
    "Volumes": [
        {
            "Iops": 3000,
            "VolumeType": "gp3",
            "MultiAttachEnabled": false,
            "Throughput": 125,
            "Operator": {
                "Managed": false
            },
            "VolumeId": "vol-01234567890abcdef",
            "Size": 8,
            "SnapshotId": "snap-0abcdef1234567890",
            "AvailabilityZone": "us-west-2b",
            "State": "in-use",
            "CreateTime": "2024-05-17T23:23:00.400000+00:00",
            "Attachments": [
                {
                    "DeleteOnTermination": true,
                    "VolumeId": "vol-01234567890abcdef",
                    "InstanceId": "i-1234567890abcdef0",
                    "Device": "/dev/xvda",
                    "State": "attached",
                    "AttachTime": "2024-05-17T23:23:00+00:00"
                }
            ],
            "Encrypted": false
```

```
        }
    ]
}
```

PowerShell

EBS 볼륨에 대한 정보를 보려면

Get-EC2Volume cmdlet을 사용합니다. 다음 예제는 현재 리전의 볼륨 수를 계산합니다.

```
(Get-EC2Volume).Count
```

다음 예제는 지정된 인스턴스에 연결된 볼륨을 나열합니다.

```
(Get-EC2Volume `
    -Filters @{Name="attachment.instance-id";Values="i-1234567890abcdef0"}).VolumeId
```

다음 예제는 지정된 볼륨을 설명합니다.

```
Get-EC2Volume -VolumeId vol-01234567890abcdef
```

다음은 예제 출력입니다.

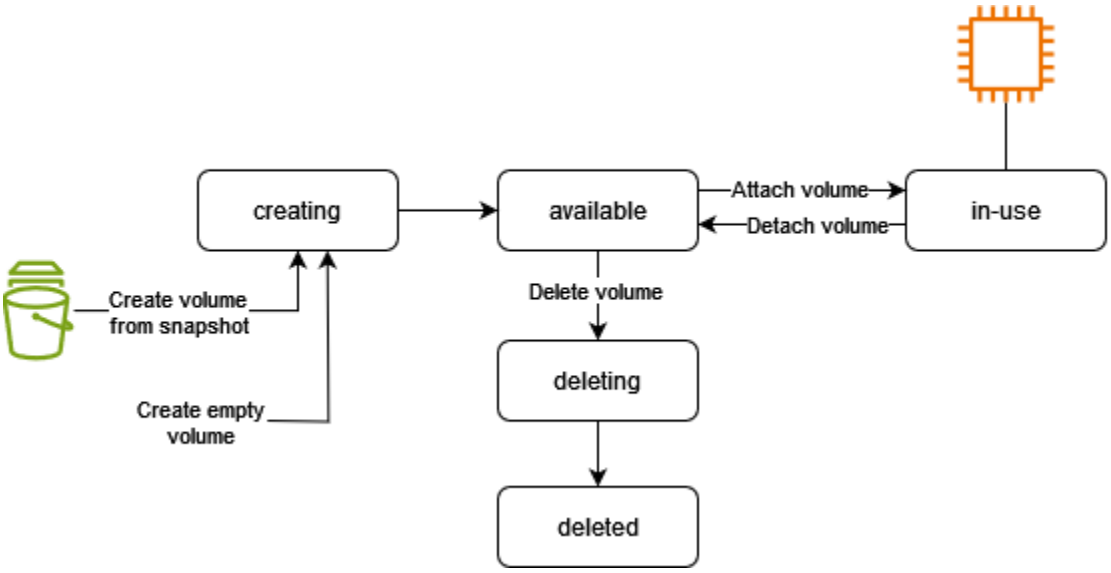
```
Attachments        : {i-1234567890abcdef0}
AvailabilityZone   : us-west-2b
CreateTime         : 5/17/2024 11:23:00 PM
Encrypted          : False
FastRestored       : False
Iops               : 3000
KmsKeyId           :
MultiAttachEnabled : False
Operator           : Amazon.EC2.Model.OperatorResponse
OutpostArn         :
Size               : 8
SnapshotId         : snap-0abcdef1234567890
SseType            :
State              : in-use
Tags               : {}
Throughput         : 125
VolumeId           : vol-01234567890abcdef
VolumeType         : gp3
```

볼륨 상태

볼륨 상태는 Amazon EBS 볼륨의 가용성을 설명합니다. 콘솔의 볼륨 페이지에 있는 상태 열에서 또는 describe-volumes 명령을 사용하여 볼륨 상태를 볼 수 있습니다. https://docs.aws.amazon.com/cli/latest/reference/ec2/describe-volumes.html AWS CLI

Amazon EBS 볼륨은 생성되는 순간부터 삭제될 때까지 다양한 상태로 전환됩니다.

다음 그림에서는 볼륨 상태 간 전환을 보여줍니다. Amazon EBS 스냅샷에서 볼륨을 생성하거나 빈 볼륨을 생성할 수 있습니다. 볼륨은 생성하면 `creating` 상태로 전환됩니다. 볼륨이 사용할 준비가 되면 `available` 상태로 전환됩니다. 사용 가능한 볼륨을 볼륨과 동일한 가용 영역에 있는 인스턴스에 연결할 수 있습니다. 볼륨을 다른 인스턴스에 연결하거나 삭제하려면 먼저 볼륨을 분리해야 합니다. 더 이상 필요하지 않은 볼륨은 삭제할 수 있습니다.



다음 표에는 볼륨 상태가 요약되어 있습니다.

State	설명
`creating`	볼륨이 생성되고 있습니다.
`available`	볼륨이 인스턴스에 연결되어 있지 않습니다.
`in-use`	볼륨이 인스턴스에 연결되어 있습니다.
`deleting`	볼륨이 삭제 중입니다.
`deleted`	볼륨이 삭제되었습니다.

State	설명
error	EBS 볼륨과 관련된 기본 하드웨어에 장애가 발생하여 볼륨과 연결된 데이터를 복구할 수 없습니다. 볼륨을 복원하거나 볼륨의 데이터를 복원하는 방법에 대한 자세한 내용은 내 EBS 볼륨이 왜 "오류" 상태입니까?를 참조하세요.

볼륨 지표 보기

Amazon CloudWatch에서 EBS 볼륨에 대한 추가 정보를 얻을 수 있습니다. 자세한 내용은 Amazon EBS에 대한 Amazon CloudWatch 지표 섹션을 참조하세요.

여유 디스크 공간 보기

사용 가능한 디스크 공간 등 EBS 볼륨에 대한 추가 정보를 인스턴스의 운영 체제에서 가져올 수 있습니다.

Linux 인스턴스

df -hT 명령을 사용하여 디바이스 이름을 지정합니다.

```
[ec2-user ~]$ df -hT /dev/xvda1
Filesystem     Type      Size  Used Avail Use% Mounted on
/dev/xvda1     xfs       8.0G  1.2G  6.9G  15% /
```

Windows 인스턴스

파일 탐색기를 열어 이 PC를 선택하면 여유 디스크 공간을 볼 수 있습니다.

다음과 같이 dir 명령을 사용하여 출력된 내용의 마지막 행에서 여유 디스크 공간을 확인할 수도 있습니다.

```
C:\> dir C:
 Volume in drive C has no label.
 Volume Serial Number is 68C3-8081

 Directory of C:\

03/25/2018  02:10 AM    <DIR>          .
03/25/2018  02:10 AM    <DIR>          ..
03/25/2018  03:47 AM    <DIR>          Contacts
```

```
03/25/2018  03:47 AM    <DIR>          Desktop
03/25/2018  03:47 AM    <DIR>          Documents
03/25/2018  03:47 AM    <DIR>          Downloads
03/25/2018  03:47 AM    <DIR>          Favorites
03/25/2018  03:47 AM    <DIR>          Links
03/25/2018  03:47 AM    <DIR>          Music
03/25/2018  03:47 AM    <DIR>          Pictures
03/25/2018  03:47 AM    <DIR>          Saved Games
03/25/2018  03:47 AM    <DIR>          Searches
03/25/2018  03:47 AM    <DIR>          Videos
               0 File(s)              0 bytes
              13 Dir(s)  18,113,662,976 bytes free
```

다음과 같이 `fsutil` 명령을 사용해 여유 디스크 공간을 확인할 수도 있습니다.

```
C:\> fsutil volume diskfree C:
Total # of free bytes        : 18113204224
Total # of bytes             : 32210153472
Total # of avail free bytes  : 18113204224
```

> **Tip**
>
> CloudWatch 에이전트를 사용하여 인스턴스에 연결하지 않고도 Amazon EC2 인스턴스에서 디스크 공간 사용량 지표를 수집할 수도 있습니다. 자세한 내용을 알아보려면 Amazon CloudWatch 사용 설명서의 CloudWatch 에이전트 구성 파일 생성과 CloudWatch 에이전트 설치를 참조하세요. 여러 인스턴스의 디스크 공간 사용량을 모니터링해야 하는 경우, Systems Manager를 사용하여 해당 인스턴스에 CloudWatch 에이전트를 설치하고 구성할 수 있습니다. 자세한 내용을 알아보려면 Systems Manager를 사용하여 CloudWatch 에이전트 설치(Installing the CloudWatch agent using Systems Manager)를 참조하세요.

탄력적 볼륨 작업을 사용하여 Amazon EBS 볼륨 수정

Amazon EBS Elastic Volumes를 통해 볼륨 크기를 늘리거나 볼륨 유형을 변경하거나 EBS 볼륨의 성능을 조정할 수 있습니다. 인스턴스가 탄력적 볼륨을 지원하는 경우에는 볼륨을 분리하거나 인스턴스를 재시작하지 않고도 이것이 가능합니다. 따라서 변경 사항이 적용되는 동안 애플리케이션을 계속 사용할 수 있습니다.

볼륨 구성 수정은 무료입니다. 볼륨 수정이 시작된 후 새 볼륨 구성에 대한 요금이 청구됩니다. 자세한 내용은 Amazon EBS 요금 페이지를 참조하세요.

내용

- 고려 사항
- 제한 사항
- Amazon EBS 볼륨 수정 요구 사항
- Amazon EBS 볼륨 수정 요청
- Amazon EBS 볼륨 수정 진행 상황 모니터링
- Amazon EBS 볼륨 크기 조정 후 파일 시스템 확장

고려 사항

- 볼륨 수정을 시작한 후에는 해당 수정이 `completed` 상태에 도달할 때까지 기다려야 동일한 볼륨에 대해 다른 수정을 시작할 수 있습니다. 볼륨이 `in-use` 또는 `available` 상태이고 해당 볼륨에 대한 모든 이전 수정이 인 경우 롤링 24시간 내에 볼륨을 최대 4회 수정할 수 있습니다`completed`. 이 제한을 초과하면 다음 수정을 수행할 수 있는 시기를 나타내는 오류 메시지가 표시됩니다.
- 볼륨 수정은 최대한 수행되며 요청된 볼륨 구성에 따라 완료하는 데 몇 분에서 몇 시간이 걸릴 수 있습니다. 일반적으로 1-TiB 볼륨을 수정하는 데 최대 6시간이 걸릴 수 있습니다. 그러나 시간이 항상 볼륨 크기에 따라 선형적으로 조정되는 것은 아닙니다. 볼륨이 클수록 시간이 줄어들고 볼륨이 작을수록 시간이 더 걸릴 수 있습니다.
- 크기 증가는 볼륨 수정이 `optimizing` 상태에 도달하면 적용되며, 일반적으로 몇 초 정도 걸립니다.
- 완전히 초기화되지 않은 볼륨의 경우 수정 시간이 늘어납니다. 자세한 내용은 볼륨 생성 후 수동 초기화 섹션을 참조하세요.
- IOPS 또는 처리량 성능을 지정gp3하지 않고 볼륨 유형을에서 gp2로 변경하면 Amazon EBS가 자동으로 성능을 프로비저닝합니다. 소스 gp2 볼륨과 동등한 성능 또는 기준 gp3 성능 중 더 높은 성능을 사용합니다.

  예를 들어 처리량이 250MiB/s이고 IOPS가 1,500인 500GiB gp2 볼륨을 IOPS나 처리량 성능을 지정하지 않고 gp3으로 수정할 경우, Amazon EBS에서 IOPS 3,000(기준 gp3 IOPS) 및 250MiB/s(소스 gp2 볼륨 처리량과 일치)으로 gp3 볼륨을 자동 프로비저닝합니다.
- EBS 볼륨을 수정하려고 시도할 때 오류 메시지가 표시되는 경우 또는 이전 세대 인스턴스 유형에 연결된 EBS 볼륨을 수정하는 경우 다음 중 한 가지 조치를 취하세요.
  - 루트가 아닌 볼륨의 경우, 인스턴스에서 볼륨을 분리하고 수정 사항을 적용한 다음 볼륨을 다시 연결합니다.
  - 루트 볼륨의 경우, 인스턴스를 중단하고 수정 사항을 적용한 다음 인스턴스를 다시 시작합니다.

제한 사항

- 볼륨 수정 요청을 제출한 후에는 취소할 수 없습니다.
- 볼륨 크기를 높여야 합니다. 볼륨 크기는 줄일 수 없습니다. 그러나 더 작은 볼륨을 생성한 다음에 rsync(Linux 인스턴스) 또는 robocopy(Windows 인스턴스) 같은 애플리케이션 수준 도구를 사용하여 해당 볼륨으로 데이터를 마이그레이션할 수 있습니다.
- 볼륨 수정 시 요청할 수 있는 최대 집계 스토리지에는 제한이 있습니다. 자세한 내용은 Amazon Web Services 일반 참조의 Amazon EBS 서비스 할당량을 참조하세요.
- 새 볼륨 크기는 볼륨의 파일 시스템 및 파티셔닝 체계에서 지원되는 용량을 초과할 수 없습니다. 자세한 내용은 Amazon EBS 볼륨 제약 조건 섹션을 참조하세요.
- 볼륨 유형을 변경하지 않는 경우 볼륨 크기와 성능 수정은 현재 볼륨 유형의 제한 범위 내에서 이루어져야 합니다. 볼륨 유형을 변경하는 경우 볼륨 크기와 성능 수정은 대상 볼륨 유형의 제한 범위 내에서 이루어져야 합니다. 자세한 내용은 Amazon EBS 볼륨 유형 섹션을 참조하세요.
- Nitro 기반 인스턴스에서는 최대 256,000 IOPS로 프로비저닝된 볼륨을 지원합니다. 다른 인스턴스 유형은 최대 64,000 IOPS로 프로비저닝된 볼륨에 연결할 수 있지만 최대 32,000 IOPS를 달성할 수 있습니다.
- 다중 연결 지원 볼륨의 경우 볼륨이 연결된 경우를 포함하여 볼륨 유형을 `io1` `io2`와 사이에서 어느 방향으로든 변경할 수 있습니다. 다른 볼륨 유형 변경 사항은 지원되지 않습니다.

  `io2` 다중 연결 활성화 볼륨의 볼륨 유형을 `io1` ~로 변경하면 볼륨의 NVMe 예약 활성화 상태가 대상 볼륨 유형으로 전달됩니다.
- 루트 볼륨 유형 `io1`, `io2`, `gp2`, `gp3` 또는 `standard`는 인스턴스에서 분리하더라도 `st1` 또는 `sc1` 볼륨으로 수정할 수 없습니다.
- 볼륨이 2016년 11월 3일 23:40 UTC 이전에 연결된 경우에는 탄력적 볼륨 지원을 초기화해야 합니다. 자세한 내용은 탄력적 볼륨 지원 초기화를 참조하십시오.
- `m3.medium` 인스턴스는 볼륨 수정을 완전하게 지원하지만 `m3.large`, `m3.xlarge` 및 `m3.2xlarge` 인스턴스는 일부 볼륨 수정 기능을 지원하지 않을 수 있습니다.

Amazon EBS 볼륨 수정 요구 사항

Amazon EBS 볼륨을 수정할 때 다음과 같은 요구 사항과 제한 사항이 적용됩니다. EBS 볼륨에 대한 일반 요구 사항에 대한 자세한 내용은 Amazon EBS 볼륨 제약 조건을 참조하세요.

주제

- 지원되는 인스턴스 유형

- 운영 체제

지원되는 인스턴스 유형

탄력적 볼륨을 지원하는 인스턴스는 다음과 같습니다.

- 모든 현재 세대 인스턴스
- 이전 세대 인스턴스: C1, C3, C4, G2, I2, M1, M3, M4, R3, R4

인스턴스 유형에서 탄력적 볼륨을 지원하지 않는 경우에는 탄력적 볼륨이 지원되지 않는 경우의 EBS 볼륨 수정 섹션을 참조하세요.

운영 체제

다음과 같은 운영 체제 요구 사항이 적용됩니다.

Linux

Linux AMI에서 부팅 볼륨 2TiB(2,048GiB) 이상을 사용하려면 GUID 파티션 테이블(GPT)과 GRUB 2가 필요합니다. 현재 여러 Linux AMI에서도 부팅 볼륨 크기를 최대 2TiB까지만 지원하는 MBR 파티셔닝 체계를 사용하고 있습니다. 인스턴스가 2TiB 이상의 부팅 볼륨에서 부팅되지 않는 경우, 사용 중인 AMI의 부팅 볼륨 크기가 2TiB 미만으로 제한된 상태일 수 있습니다. 부팅 볼륨이 아닌 볼륨에는 이 Linux 인스턴스에 대한 제한이 적용되지 않습니다.

부팅 볼륨을 2TiB 이상으로 크기를 조정하기 전에 인스턴스에서 다음 명령을 실행하여 볼륨이 MBR 또는 GPT 파티셔닝을 사용하는지 확인할 수 있습니다.

```
[ec2-user ~]$ sudo gdisk -l /dev/xvda
```

GPT 파티셔닝을 사용하는 Amazon Linux 인스턴스는 다음 정보를 반환합니다.

```
GPT fdisk (gdisk) version 0.8.10

  Partition table scan:
    MBR: protective
    BSD: not present
    APM: not present
    GPT: present

  Found valid GPT with protective MBR; using GPT.
```

MBR 파티셔닝을 사용하는 SUSE 인스턴스는 다음 정보를 반환합니다.

```
GPT fdisk (gdisk) version 0.8.8

  Partition table scan:
    MBR: MBR only
    BSD: not present
    APM: not present
    GPT: not present
```

Windows

기본적으로 Windows는 마스터 부트 레코드(MBR) 파티션 테이블을 사용하여 볼륨을 초기화합니다. MBR는 2TiB(2,048GiB) 미만의 볼륨만 지원하기 때문에 Windows에서는 MBR 볼륨 크기를 이 한도를 넘는 크기로 변경할 수 없습니다. 이렇게 하면 Windows 디스크 관리 유틸리티에서 볼륨 확장 옵션이 비활성화됩니다. AWS Management Console 또는를 사용하여 크기 제한을 초과하는 MBR 분할 볼륨을 AWS CLI 생성하는 경우 Windows는 추가 공간을 감지하거나 사용할 수 없습니다.

이 제한을 해결하려면 GUID 파티션 테이블(GPT)을 사용하여 더 큰 볼륨을 새로 만든 후 원래 MBR 볼륨의 데이터를 복사합니다.

GPT 볼륨을 생성하려면

1. EC2 인스턴스의 가용 영역에서 원하는 크기의 비어 있는 새 볼륨을 생성하고 이 볼륨을 인스턴스에 연결합니다.

   > **Note**
   >
   > 새 볼륨은 스냅샷에서 복원한 볼륨이 아니어야 합니다.

2. Windows 시스템에 로그인하고 디스크 관리(diskmgmt.exe)를 엽니다.
3. 새 디스크를 마우스 오른쪽 버튼으로 클릭하여 컨텍스트 메뉴를 열고 온라인을 선택합니다.
4. 디스크 초기화 창에서 새 디스크를 선택하고 GPT(GUID 파티션 테이블), 확인을 선택합니다.
5. 초기화가 완료되면 robocopy 또는 teracopy 등의 도구를 사용하여 원래 볼륨의 데이터를 새 볼륨으로 복사합니다.
6. 디스크 관리에서 드라이브 문자를 적절한 값으로 변경하고 기존 볼륨을 오프라인으로 전환합니다.
7. Amazon EC2 콘솔에서 기존 볼륨을 인스턴스에서 분리하고, 인스턴스를 재부팅하여 인스턴스가 제대로 작동하는지 확인한 다음, 기존 볼륨을 삭제합니다.

Amazon EBS 볼륨 수정 요청

탄력적 볼륨을 사용하면 Amazon EBS 볼륨을 분리하지 않고도 크기를 늘리고, 성능을 높이거나 낮추고, 볼륨 유형을 동적으로 변경할 수 있습니다.

프로세스 개요

1. (선택 사항) 중요한 데이터가 포함된 볼륨을 수정하려면 먼저 변경 내용을 롤백해야 할 경우를 대비하여 볼륨의 스냅샷을 생성하는 것이 바람직합니다. 자세한 내용은 Amazon EBS 스냅샷 생성 섹션을 참조하세요.
2. 볼륨 수정을 요청합니다.
3. 볼륨 수정의 진행 상황을 모니터링합니다. 자세한 내용은 Amazon EBS 볼륨 수정 진행 상황 모니터링 섹션을 참조하세요.
4. 볼륨 크기가 수정된 경우 볼륨의 파일 시스템을 확장하여 스토리지 용량 증가를 활용합니다. 자세한 내용은 Amazon EBS 볼륨 크기 조정 후 파일 시스템 확장 섹션을 참조하세요.

목차

- 탄력적 볼륨을 사용하여 EBS 볼륨 수정
- 탄력적 볼륨이 지원되지 않는 경우의 EBS 볼륨 수정
- 탄력적 볼륨 지원 초기화(필요한 경우)

탄력적 볼륨을 사용하여 EBS 볼륨 수정

시작하기 전에 다음을 수행하세요.

- 고려 사항
- 제한 사항
- 요구 사항

Console

EBS 볼륨을 수정하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 볼륨을 선택합니다.
3. 수정할 볼륨을 선택하고 작업(Actions), 볼륨 수정(Modify volume)을 선택합니다.

4. 볼륨 수정(Modify volume) 화면에 볼륨 ID와 유형, 크기, IOPS 및 처리량을 포함한 볼륨의 현재 구성이 표시됩니다. 다음과 같이 새로운 구성 값을 설정합니다.

   - 유형을 수정하려면 볼륨 유형(Volume type)의 값을 선택합니다.
   - 크기를 수정하려면 [크기(Size)]에 대한 새 값을 입력합니다.
   - (gp3, io1 및 io2에만 해당) IOPS를 수정하려면 IOPS에 대한 새 값을 입력합니다.
   - (gp3에만 해당) 처리량을 수정하려면 처리량(Throughput)에 대한 새 값을 입력합니다.

5. 볼륨 설정 변경을 완료했으면 수정을 선택합니다. 확인 메시지가 나타나면 수정(Modify)을 선택합니다.
6. 볼륨 크기를 늘린 경우 추가 저장 용량을 사용하려면 볼륨의 파티션도 확장해야 합니다. 자세한 내용은 Amazon EBS 볼륨 크기 조정 후 파일 시스템 확장 섹션을 참조하세요.
7. (Windows 인스턴스만 해당) AWS NVMe 드라이버가 없는 인스턴스에서 NVMe 볼륨의 크기를 늘리는 경우 Windows가 새 볼륨 크기를 볼 수 있도록 인스턴스를 재부팅해야 합니다. AWS NVMe 드라이버 설치에 대한 자세한 내용은 AWS NVMe 드라이버를 참조하세요.

AWS CLI

EBS 볼륨을 수정하려면

modify-volume 명령을 사용합니다. 예를 들어 크기가 100GiB이고 유형이 gp2인 볼륨을 가지고 있는 경우, 다음 예제가 IOPS가 10,000이고 크기가 200GiB이며 유형이 io1인 볼륨에 대한 구성을 변경합니다.

```
aws ec2 modify-volume \
    --volume-id vol-01234567890abcdef \
    --volume-type io1 \
    --iops 10000 \
    --size 200
```

다음은 예제 출력입니다.

```
{
    "VolumeModification": {
        "TargetSize": 200,
        "TargetVolumeType": "io1",
        "ModificationState": "modifying",
        "VolumeId": "vol-01234567890abcdef",
        "TargetIops": 10000,
```

```
        "StartTime": "2022-01-19T22:21:02.959Z",
        "Progress": 0,
        "OriginalVolumeType": "gp2",
        "OriginalIops": 300,
        "OriginalSize": 100
    }
}
```

볼륨 크기를 늘린 경우 추가 저장 용량을 사용하려면 볼륨의 파티션도 확장해야 합니다. 자세한 내용은 Amazon EBS 볼륨 크기 조정 후 파일 시스템 확장 섹션을 참조하세요.

PowerShell

EBS 볼륨을 수정하려면

Edit-EC2Volume cmdlet을 사용합니다. 예를 들어 크기가 100GiB이고 유형이 gp2인 볼륨을 가지고 있는 경우, 다음 예제가 IOPS가 10,000이고 크기가 200GiB이며 유형이 io1인 볼륨에 대한 구성을 변경합니다.

```
Edit-EC2Volume `
    -VolumeId vol-01234567890abcdef `
    -VolumeType io1 `
    -Iops 10000 `
    -Size 200
```

볼륨 크기를 늘린 경우 추가 저장 용량을 사용하려면 볼륨의 파티션도 확장해야 합니다. 자세한 내용은 Amazon EBS 볼륨 크기 조정 후 파일 시스템 확장 섹션을 참조하세요.

탄력적 볼륨이 지원되지 않는 경우의 EBS 볼륨 수정

지원되는 인스턴스 유형을 사용하고 있는 경우에는 탄력적 볼륨을 이용해 Amazon EBS 볼륨을 분리하지 않고도 크기, 성능 및 볼륨 유형을 동적으로 수정할 수 있습니다.

탄력적 볼륨을 사용할 수는 없지만 루트(부트) 볼륨을 수정해야 하는 경우에는 인스턴스를 중지하고 볼륨을 수정한 후 인스턴스를 다시 시작해야 합니다.

인스턴스가 시작된 후 파일 시스템의 크기를 확인하여 인스턴스가 더 큰 볼륨 공간을 인식하는지 파악할 수 있습니다. Linux에서는 df -h 명령을 사용하여 파일 시스템의 크기를 확인합니다.

```
[ec2-user ~]$ df -h
Filesystem            Size  Used Avail Use% Mounted on
```

```
/dev/xvda1            7.9G  943M  6.9G  12% /
tmpfs                 1.9G     0  1.9G   0% /dev/shm
```

새로 확장된 볼륨이 크기에 반영되지 않을 경우 인스턴스에서 새 공간을 사용할 수 있도록 디바이스의 파일 시스템을 확장해야 합니다. 자세한 내용은 Amazon EBS 볼륨 크기 조정 후 파일 시스템 확장 섹션을 참조하세요.

볼륨을 Windows 인스턴스에서 사용하려면 온라인 상태로 전환해야 할 수 있습니다. 자세한 내용은 Amazon EBS 볼륨을 사용할 수 있도록 만들기 섹션을 참조하세요. 볼륨을 다시 포맷할 필요는 없습니다.

탄력적 볼륨 지원 초기화(필요한 경우)

2016년 11월 3일 23:40 UTC 이전에 인스턴스에 연결된 볼륨을 수정하기 전에 다음 중 한 가지 조치를 취하여 볼륨 수정 지원을 초기화해야 합니다.

- 볼륨을 분리한 후 다시 연결합니다.
- 인스턴스 중지 및 시작

Console

인스턴스가 준비가 되었는지를 확인하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. 열 표시/숨기기(Show/Hide Columns) 아이콘(기어 모양)을 선택합니다. 시작 시간 속성 열을 선택한 다음 확인을 선택합니다.
4. 시작 시간(Launch Time) 열을 기준으로 인스턴스의 목록을 정렬합니다. 컷오프 날짜 이전에 시작된 각 인스턴스에 대해 스토리지 탭을 선택하고 연결 시간 열에서 볼륨이 연결된 시간을 확인합니다.

AWS CLI

인스턴스가 준비가 되었는지를 확인하려면

다음 describe-instances 명령을 사용하여 2016년 11월 3일 23:40 UTC 전에 볼륨이 연결되었는지 여부를 확인합니다.

```
aws ec2 describe-instances \
```

```
    --query "Reservations[*].Instances[*].
[InstanceId,LaunchTime<='2016-11-01',BlockDeviceMappings[*]
[Ebs.AttachTime<='2016-11-01']]" \
    --output text
```

각 인스턴스의 출력 첫 줄에는 해당 ID와 컷오프 날짜 이전에 시작되었는지 여부(True 또는 False)가 표시됩니다. 첫 줄 다음에는 각 EBS 볼륨이 컷오프 날짜 이전에 연결되었는지 여부를 보여주는 (True 또는 False) 줄이 하나 이상 뒤따라 표시됩니다. 다음 예제 출력에서, 첫 번째 인스턴스가 컷오프 날짜 이전에 시작되었으며 해당 루트 볼륨이 컷오프 날짜 이전에 연결되었으므로 첫 번째 인스턴스에 대한 볼륨 수정을 초기화해야 합니다. 다른 인스턴스는 컷오프 날짜 이후에 시작되었으므로 사용 준비가 되었습니다.

```
i-e905622e              True
True
i-719f99a8              False
True
i-006b02c1b78381e57     False
False
False
i-e3d172ed              False
True
```

PowerShell

인스턴스가 준비가 되었는지 확인하려면

Get-EC2Instance cmdlet을 사용하여 볼륨이 2016년 11월 3일 23:40 UTC 이전에 연결되었는지 여부를 확인합니다.

```
(Get-EC2Instance `
    -InstanceId i-1234567890abcdef0).Instances.BlockDeviceMappings | `
     Format-Table @{Name="VolumeId";Expression={$_.Ebs.VolumeId}}, `
                  @{Name="AttachTime";Expression={$_.Ebs.AttachTime}}
```

다음은 예제 출력입니다.

```
VolumeId              AttachTime
--------              ----------
vol-0b243c8d927752d2b 3/23/2020 12:21:14 AM
vol-043eadbeb4a8387c3 9/5/2020 7:39:22 PM
vol-0c3f0c4e55c082753 4/23/2019 4:07:40 PM
```

Amazon EBS 볼륨 수정 진행 상황 모니터링

수정 시 EBS 볼륨은 상태 시퀀스를 통과합니다. 볼륨은 `modifying` 상태가 된 다음 `optimizing` 상태가 되고, 마지막으로 `completed` 상태가 됩니다. 그러면 볼륨을 더 수정할 준비가 완료됩니다.

볼륨이 `optimizing` 상태에 있는 동안 볼륨 성능은 소스 및 대상 구성 사양 사이에 있습니다. 일시적인 볼륨 성능은 소스 볼륨 성능 이상입니다. IOPS를 다운로드하면 일시적인 볼륨 성능은 대상 볼륨 성능 이상입니다.

볼륨 수정 변경 사항은 다음과 같이 적용됩니다.

- 크기 증가는 볼륨 수정이 `optimizing` 상태에 도달하면 적용되며, 일반적으로 몇 초 정도 걸립니다.
- 요청된 볼륨 구성에 따라 성능(IOPS 및 처리량) 변경을 완료하는 데 몇 분에서 몇 시간이 걸릴 수 있습니다. 일반적으로 완전히 사용되는 1-TiB 볼륨은 새 성능 구성으로 마이그레이션하는 데 약 6시간이 걸릴 수 있습니다. 경우에 따라 볼륨이 완전히 초기화되지 않은 경우와 같이 새 성능 구성이 적용되는 데 24시간 이상 걸릴 수 있습니다.

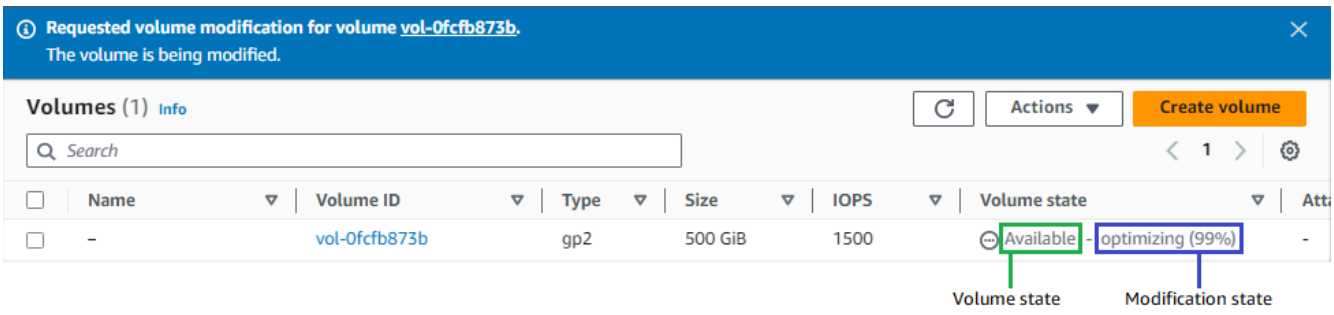
가능한 볼륨 상태는 `creating`, `available`, `in-use`, `deleting`, `deleted` 및 `error`입니다.

가능한 수정 상태는 `modifying`, `optimizing` 및 `completed`입니다.

Console

수정 진행 상태를 모니터링하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 볼륨을 선택합니다.
3. 볼륨을 선택합니다.
4. 세부 정보 탭의 볼륨 상태 열과 볼륨 상태 필드에는 *## ## - ## ##(## ###)* 형식의 정보가 들어 있습니다. 다음 이미지는 볼륨 및 볼륨 수정 상태를 보여줍니다.



수정이 완료되면 볼륨 상태만 표시됩니다. 수정 상태 및 진행 상황이 더 이상 표시되지 않습니다.

또는 Amazon EventBridge를 사용하여 볼륨 수정 이벤트에 대한 알림 규칙을 만들 수 있습니다. 자세한 내용은 Amazon EventBridge 시작하기를 참조하세요.

AWS CLI

수정 진행 상태를 모니터링하려면

describe-volumes-modifications 명령을 사용하여 하나 이상의 볼륨 수정 진행 상황을 모니터링합니다. 다음 예제에서는 두 볼륨의 볼륨 수정을 설명합니다.

```
aws ec2 describe-volumes-modifications \
    --volume-ids vol-1111111111111111 vol-2222222222222222
```

다음 예제 출력에서 볼륨 수정의 여전히 modifying 상태입니다. 진행률은 백분율로 보고됩니다.

```
{
    "VolumesModifications": [
        {
            "TargetSize": 200,
            "TargetVolumeType": "io1",
            "ModificationState": "modifying",
            "VolumeId": "vol-1111111111111111",
            "TargetIops": 10000,
            "StartTime": "2017-01-19T22:21:02.959Z",
            "Progress": 0,
            "OriginalVolumeType": "gp2",
            "OriginalIops": 300,
            "OriginalSize": 100
        },
        {
            "TargetSize": 2000,
            "TargetVolumeType": "sc1",
            "ModificationState": "modifying",
            "VolumeId": "vol-2222222222222222",
            "StartTime": "2017-01-19T22:23:22.158Z",
            "Progress": 0,
            "OriginalVolumeType": "gp2",
            "OriginalIops": 300,
```

```
            "OriginalSize": 1000
        }
    ]
}
```

다음 예에서는 수정 상태가 optimizing 또는 completed인 모든 볼륨을 설명하고, 2017년 2월 1일 이후 시작된 수정만 표시하도록 결과를 필터링 및 형식 지정합니다.

```
aws ec2 describe-volumes-modifications \
    --filters Name=modification-state,Values="optimizing","completed" \
    --query "VolumesModifications[?StartTime>='2017-02-01'].
{ID:VolumeId,STATE:ModificationState}"
```

다음은 두 볼륨에 대한 정보가 포함된 출력 예제입니다.

```
[
    {
        "STATE": "optimizing",
        "ID": "vol-06397e7a0eEXAMPLE"
    },
    {
        "STATE": "completed",
        "ID": "vol-ba74e18c2aEXAMPLE"
    }
]
```

PowerShell

수정 진행 상태를 모니터링하려면

Get-EC2VolumeModification cmdlet을 사용합니다. 다음 예제에서는 두 볼륨의 볼륨 수정을 설명합니다.

```
Get-EC2VolumeModification `
    -VolumeId vol-1111111111111111 vol-2222222222222222
```

> **Note**
>
> 드물게 일시적인 AWS 장애로 인해 `failed` 상태가 발생할 수 있습니다. 이는 볼륨 상태를 나타내는 것이 아니라 단지 볼륨 수정이 실패했음을 나타내는 것입니다. 이 경우 볼륨 수정을 다시 시도합니다.

Amazon EBS 볼륨 크기 조정 후 파일 시스템 확장

EBS 볼륨 크기 증가 후 파일 시스템을 새롭게 더 큰 크기로 확장하려면 파티션과 파일 시스템을 확장해야 합니다. 볼륨이 `optimizing` 상태가 되자마자 이 작업을 수행할 수 있습니다.

시작하기 전 준비 사항

- 변경 사항을 롤백해야 하는 경우에 대비하여 볼륨의 스냅샷을 생성합니다. 자세한 내용은 Amazon EBS 스냅샷 생성 섹션을 참조하세요.
- 볼륨 수정이 성공했으며 `optimizing` 또는 `completed` 상태인지 확인합니다. 자세한 내용은 Amazon EBS 볼륨 수정 진행 상황 모니터링 섹션을 참조하세요.
- 볼륨이 인스턴스에 연결되어 있고 형식이 지정되고 마운트되어 있는지 확인합니다. 자세한 내용은 연결된 볼륨 포맷 및 탑재 섹션을 참조하세요.
- (Linux 인스턴스에만 해당) Amazon EBS 볼륨에서 논리 볼륨을 사용하는 경우, 논리 볼륨 관리자(LVM)를 사용하여 논리 볼륨을 확장해야 합니다. 이 작업을 수행하는 방법에 대한 지침을 알아보려면 문서 LVM을 사용하여 Amazon EBS 볼륨의 파티션에 논리 볼륨을 생성하려면 어떻게 해야 합니까?의 LV 확장 섹션을 참조하세요.

Linux 인스턴스

> **Note**
>
> 다음과 같은 지침에서는 Linux용 XFS 및 Ext4 파일 시스템을 확장하는 프로세스를 안내합니다. 다른 파일 시스템을 확장하는 방법에 대한 자세한 내용은 해당 설명서를 참조하세요.

볼륨에 파티션이 있으면 파티션을 먼저 확장해야 Linux에서 파일 시스템을 확장할 수 있습니다.

EBS 볼륨의 파일 시스템 확장

크기가 조정된 볼륨의 파일 시스템을 확장하려면 다음 절차를 따르세요.

디바이스 및 파티션 이름 지정은 Xen 인스턴스와 Nitro System에 구축된 인스턴스에 따라 다릅니다. 인스턴스가 Xen 기반인지 Nitro 기반인지 여부를 확인하려면 Amazon EC2 하이퍼바이저 유형을 참조하세요.

EBS 볼륨의 파일 시스템을 확장하려면

1. 인스턴스에 연결합니다.
2. 필요한 경우 파티션 크기를 조정합니다. 그렇게 하려면 다음을 수행하세요.

   a. 볼륨에 파티션이 있는지 확인합니다. lsblk 명령을 사용합니다.

      Nitro instance example

      다음 예제 출력에서 루트 볼륨(nvme0n1)에는 2개의 파티션(nvme0n1p1 및 nvme0n1p128)이 있는 반면 추가 볼륨(nvme1n1)에는 파티션이 없습니다.

```
[ec2-user ~]$ sudo lsblk
NAME          MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
nvme1n1       259:0    0  30G  0 disk /data
nvme0n1       259:1    0  16G  0 disk
##nvme0n1p1   259:2    0   8G  0 part /
##nvme0n1p128 259:3    0   1M  0 part
```

      Xen instance example

      다음 예제 출력에서 루트 볼륨(xvda)에는 1개의 파티션(xvda1)이 있는 반면 추가 볼륨(xvdf)에는 파티션이 없습니다.

```
[ec2-user ~]$ sudo lsblk
NAME    MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
xvda    202:0    0  16G  0 disk
##xvda1 202:1    0   8G  0 part /
xvdf    202:80   0  24G  0 disk
```

   - 볼륨에 파티션이 있으면 다음 단계(2b)를 계속합니다.
   - 볼륨에 파티션이 없으면 2b, 2c 및 2d 단계를 건너뛰고 3단계를 계속 진행합니다.

> **문제 해결 도움말**
>
> 명령 출력에 볼륨이 표시되지 않는 경우 볼륨이 인스턴스에 연결되어 있고 형식이 지정되고 마운트되어 있는지 확인합니다.

b. 파티션을 확장해야 하는지 여부를 확인합니다. 이전 단계의 lsblk 명령 출력에서 파티션 크기와 볼륨 크기를 비교합니다.

- 파티션 크기가 볼륨 크기보다 작은 경우 다음 단계(2c)를 계속 진행합니다.
- 파티션 크기가 볼륨 크기와 같은 경우, 파티션을 확장할 필요가 없습니다. 2c단계와 2d단계를 건너뛰고 3단계로 계속 진행하세요.

> **문제 해결 도움말**
>
> 볼륨이 여전히 원래 크기를 반영하는 경우 볼륨 수정이 성공했는지 확인합니다.

c. 파티션을 확장합니다. growpart 명령을 사용하여 디바이스 이름과 파티션 번호를 지정합니다.

Nitro instance example

파티션 번호는 p 뒤에 오는 숫자입니다. 예를 들어 nvme0n1p1의 경우 파티션 번호는 1입니다. nvme0n1p128의 경우 파티션 번호는 128입니다.

nvme0n1p1이라는 파티션을 확장하려면 다음 명령을 사용합니다.

> **Important**
>
> 디바이스 이름(nvme0n1)과 파티션 번호(1) 사이의 공백에 유의하세요.

```
[ec2-user ~]$ sudo growpart /dev/nvme0n1 1
```

Xen instance example

파티션 번호는 디바이스 이름 뒤의 번호입니다. 예를 들어 xvda1의 경우 파티션 번호는 1입니다. xvda128의 경우 파티션 번호는 128입니다.

xvda1이라는 파티션을 확장하려면 다음 명령을 사용합니다.

> **Important**
>
> 디바이스 이름(xvda)과 파티션 번호(1) 사이의 공백에 유의하세요.

```
[ec2-user ~]$ sudo growpart /dev/xvda 1
```

> **문제 해결 팁**
>
> - `mkdir: cannot create directory '/tmp/growpart.31171': No space left on device FAILED: failed to make temp dir`: 크기 조정을 수행하는 데 필요한 임시 디렉터리를 생성하기 위해 볼륨에 여유 디스크 공간이 부족함을 나타냅니다. 디스크 공간을 확보한 다음 다시 시도합니다.
> - `must supply partition-number`: 잘못된 파티션을 지정했음을 나타냅니다. lsblk 명령을 사용하여 파티션 이름을 확인하고 디바이스 이름과 파티션 번호 사이에 공백을 입력해야 합니다.
> - `NOCHANGE: partition 1 is size 16773087. it cannot be grown`: 파티션이 이미 전체 볼륨을 확장하고 확장할 수 없음을 나타냅니다. 볼륨 수정이 성공했는지 확인합니다.

d. 파티션이 확장되었는지 확인합니다. lsblk 명령을 사용합니다. 이제 파티션 크기가 볼륨 크기와 같아야 합니다.

Nitro instance example

다음 예제 출력은 볼륨(nvme0n1)과 파티션(nvme0n1p1)이 동일한 크기(16 GB)임을 보여줍니다.

```
[ec2-user ~]$ sudo lsblk
NAME          MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
nvme1n1       259:0    0  30G  0 disk /data
nvme0n1       259:1    0  16G  0 disk
##nvme0n1p1   259:2    0  16G  0 part /
```

```
##nvme0n1p128 259:3    0   1M  0 part
```

Xen instance example

다음 예제 출력은 볼륨(xvda)과 파티션(xvda1)이 동일한 크기(16 GB)임을 보여줍니다.

```
[ec2-user ~]$ sudo lsblk
NAME    MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
xvda    202:0    0  16G  0 disk
##xvda1 202:1    0  16G  0 part /
xvdf    202:80   0  24G  0 disk
```

3. 파일 시스템을 확장합니다.

   a. 확장해야 하는 파일 시스템의 이름, 크기, 유형 및 탑재 지점을 가져옵니다. df -hT 또는 lsblk -f 명령을 사용합니다.

   Nitro instance example

   df -hT 명령에 대한 다음 예제 출력은 /dev/nvme0n1p1 파일 시스템의 크기가 8GB이고 유형이 xfs이며 탑재 지점이 /임을 보여줍니다.

   ```
   [ec2-user ~]$ df -hT
   Filesystem      Type  Size  Used Avail Use% Mounted on
   /dev/nvme0n1p1  xfs   8.0G  1.6G  6.5G  20% /
   /dev/nvme1n1    xfs   8.0G   33M  8.0G   1% /data
   ...
   ```

   Xen instance example

   df -hT 명령에 대한 다음 예제 출력은 /dev/xvda1 파일 시스템의 크기가 8GB이고 유형이 ext4이며 탑재 지점이 /임을 보여줍니다.

   ```
   [ec2-user ~]$ df -hT
   Filesystem      Type   Size    Used   Avail   Use%   Mounted on
   /dev/xvda1      ext4   8.0G    1.9G   6.2G    24%    /
   /dev/xvdf1      xfs    24.0G   45M    8.0G    1%     /data
   ...
   ```

   - 파일 시스템 크기가 볼륨 크기보다 작은 경우 다음 단계(3b)를 계속합니다.

- 파일 시스템 크기가 볼륨 크기와 같은 경우, 확장할 필요가 없습니다. 이 경우에는 나머지 단계를 건너뜁니다. 파티션 및 파일 시스템은 이미 새 볼륨 크기로 확장되었습니다.

b. 파일 시스템을 확장하는 명령은 파일 시스템 유형에 따라 다릅니다. 이전 단계에서 기록한 파일 시스템 유형에 따라 다음 올바른 명령을 선택합니다.

- [XFS 파일 시스템] xfs_growfs 명령을 사용하여 이전 단계에서 기록한 파일 시스템의 탑재 지점을 지정합니다.

  Nitro and Xen instance example

  예를 들어 /에 탑재된 파일 시스템을 확장하려면 다음 명령을 사용합니다.

  ```
  [ec2-user ~]$ sudo xfs_growfs -d /
  ```

  문제 해결 팁

  - `xfs_growfs: /data is not a mounted XFS filesystem`: 잘못된 탑재 지점을 지정했거나 파일 시스템이 XFS가 아님을 나타냅니다. 탑재 지점과 파일 시스템 유형을 확인하려면 df -hT 명령을 사용합니다.
  - `data size unchanged, skipping`: 파일 시스템이 이미 전체 볼륨을 확장하고 있음을 나타냅니다. 볼륨에 파티션이 없는 경우 볼륨 수정이 성공했는지 확인합니다. 볼륨에 파티션이 있는 경우 2단계에 설명된 대로 파티션이 확장되었는지 확인합니다.

- [Ext4 파일 시스템] resize2fs 명령을 사용하여 이전 단계에서 기록한 파일 시스템의 이름을 지정합니다.

  Nitro instance example

  예를 들어 /dev/nvme0n1p1이라는 이름으로 탑재된 파일 시스템을 확장하려면 다음 명령을 사용합니다.

  ```
  [ec2-user ~]$ sudo resize2fs /dev/nvme0n1p1
  ```

  Xen instance example

  예를 들어 /dev/xvda1이라는 이름으로 탑재된 파일 시스템을 확장하려면 다음 명령을 사용합니다.

```
[ec2-user ~]$ sudo resize2fs /dev/xvda1
```

문제 해결 팁

- `resize2fs: Bad magic number in super-block while trying to open /dev/xvda1`: 파일 시스템이 Ext4가 아님을 나타냅니다. 탑재 지점과 파일 시스템 유형을 확인하려면 df -hT 명령을 사용합니다.
- `open: No such file or directory while opening /dev/xvdb1`: 잘못된 파티션을 지정했음을 나타냅니다. 파티션을 확인하려면 df -hT 명령을 사용합니다.
- `The filesystem is already 3932160 blocks long. Nothing to do!`: 파일 시스템이 이미 전체 볼륨을 확장하고 있음을 나타냅니다. 볼륨에 파티션이 없는 경우 볼륨 수정이 성공했는지 확인합니다. 볼륨에 파티션이 있는 경우 2단계에 설명된 대로 파티션이 확장되었는지 확인합니다.

- [기타 파일 시스템] 사용 중인 파일 시스템의 설명서에서 지침을 참조하세요.

c. 파일 시스템이 확장되었는지 확인합니다. df -hT 명령을 사용하여 파일 시스템 크기가 볼륨 크기와 같은지 확인합니다.

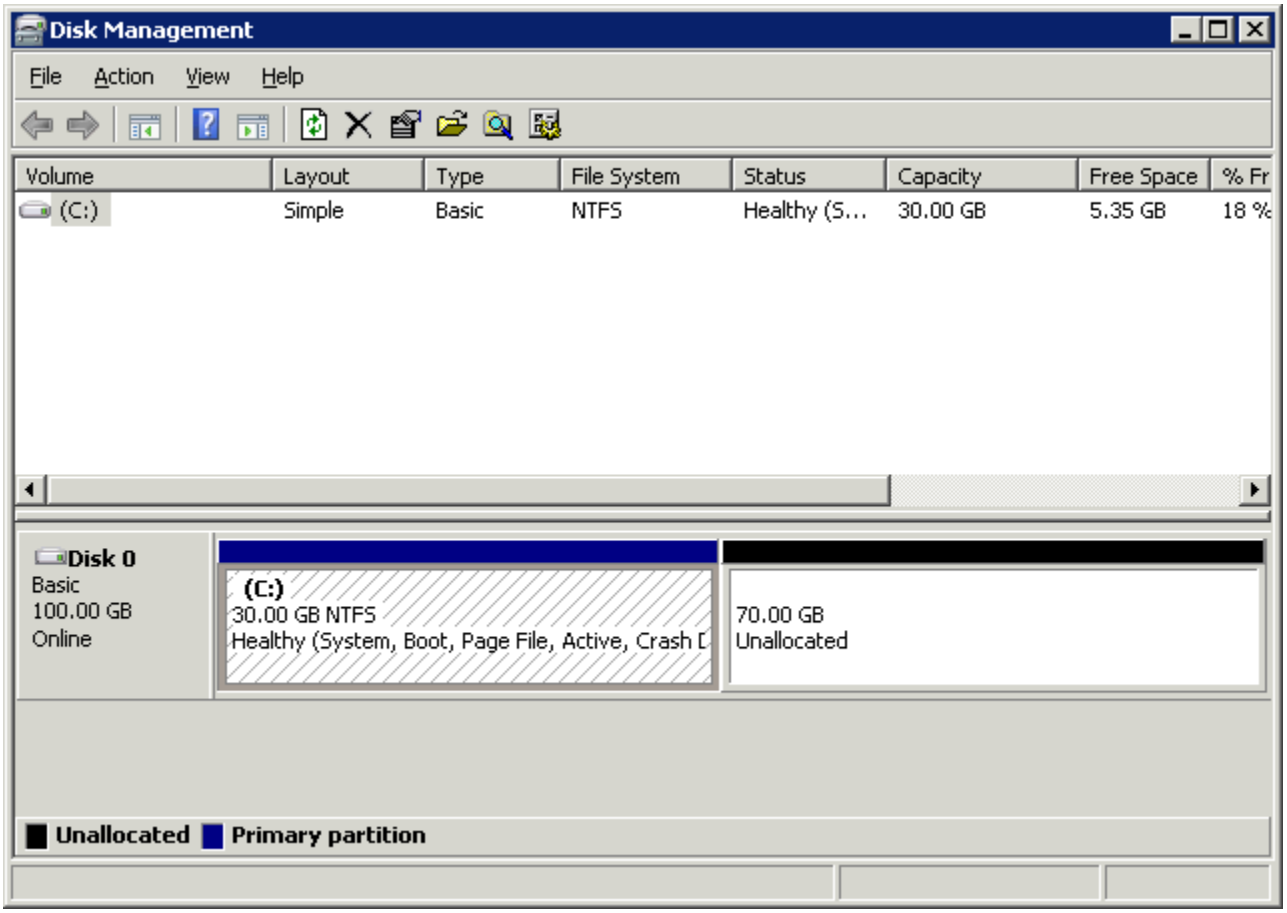
Windows 인스턴스

Windows 인스턴스에서 파일 시스템을 확장하려면 다음과 같은 방법 중 하나를 사용합니다.

Disk Management utility

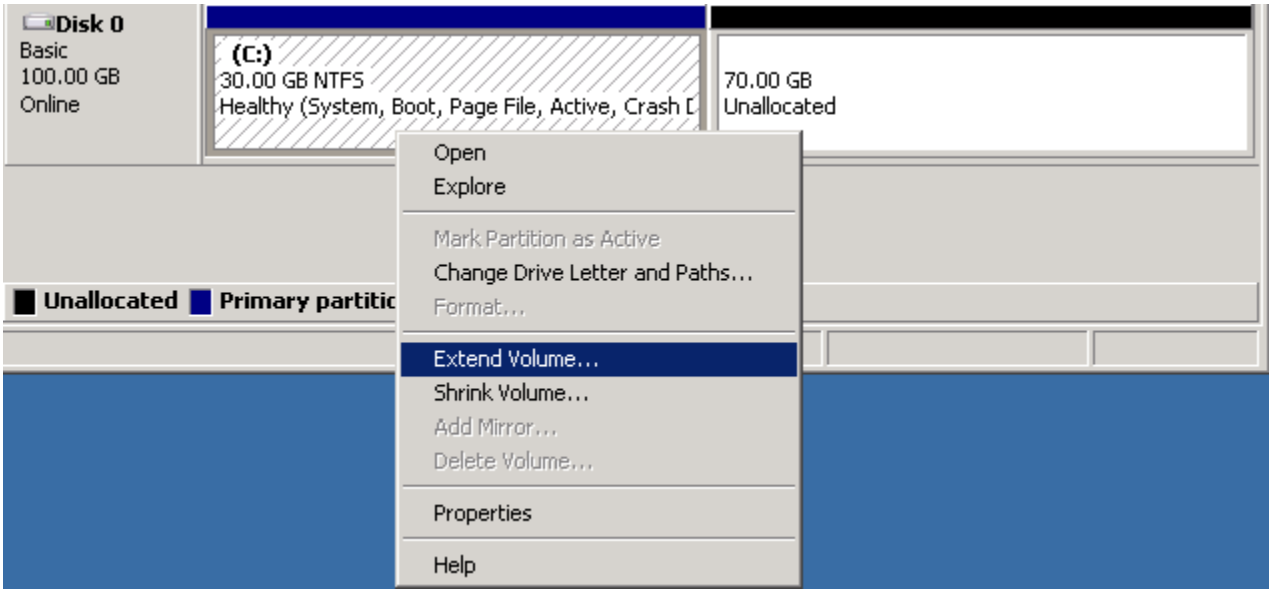
디스크 관리를 사용하여 파일 시스템을 확장하려면

1. 중요한 데이터가 저장된 파일 시스템을 확장하려면 먼저 변경 내용을 롤백해야 할 경우를 대비하여 파일 시스템이 저장된 볼륨 스냅샷을 생성하는 것이 바람직합니다. 자세한 내용은 Amazon EBS 스냅샷 생성 섹션을 참조하세요.
2. 원격 데스크톱을 사용하여 Windows 인스턴스에 로그인합니다.
3. [실행(Run)] 대화 상자에 diskmgmt.msc를 입력하고 Enter 키를 누릅니다. 디스크 관리 유틸리티가 열립니다.

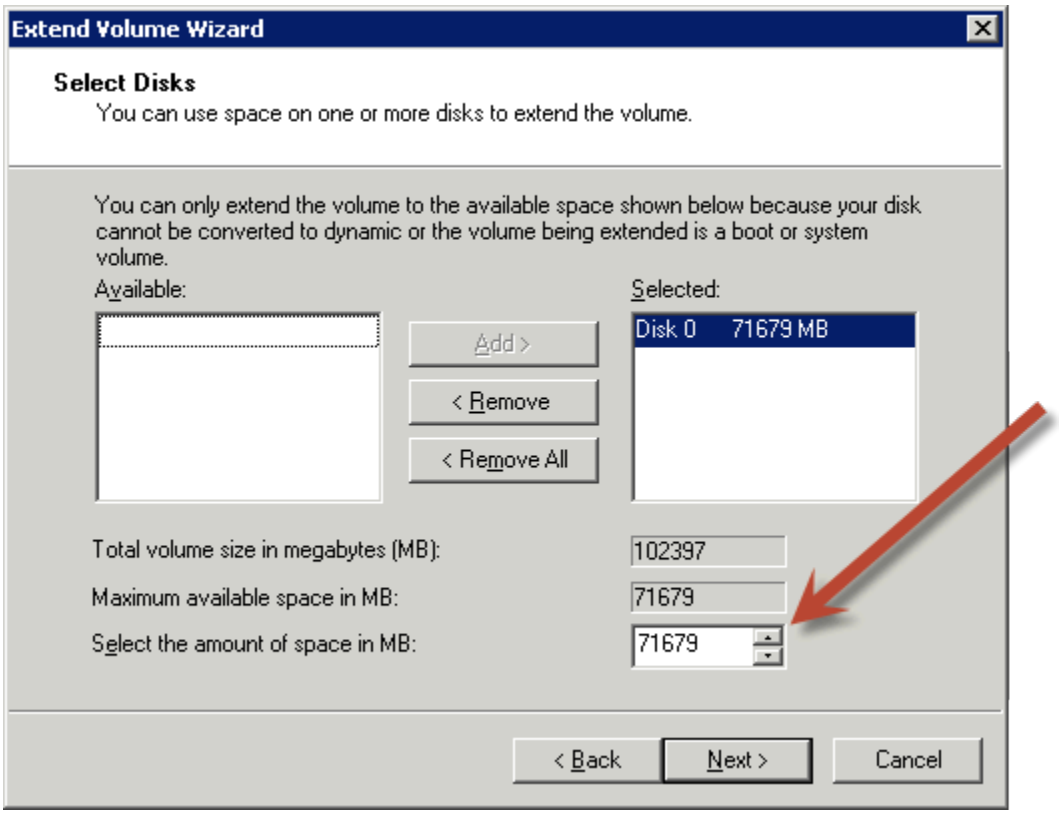


4. 디스크 관리(Disk Management) 메뉴에서 작업, 디스크 다시 스캔(Rescan Disks)을 선택합니다.
5. 확장된 드라이브를 오른쪽 클릭하여 컨텍스트 메뉴를 열고 볼륨 확장(Extend Volume)을 선택합니다.

> **Note**
>
> 다음과 같은 경우 [볼륨 확장]이 비활성화(회색으로 표시)될 수 있습니다.
>
> - 할당되지 않은 공간이 드라이브에 인접하지 않습니다. 할당되지 않은 공간이 확장할 드라이브의 오른쪽에 인접해야 합니다.
> - 볼륨은 마스터 부트 레코드(MBR) 파티션 스타일을 사용하며 이미 크기가 2TB입니다. MBR을 사용하는 볼륨의 크기는 2TB를 초과할 수 없습니다.



6. Extend Volume(볼륨 확장) 마법사에서 다음을 선택합니다. MB 단위로 공간 크기 선택(Select the amount of space in MB)에 볼륨 확장에 적용할 메가바이트 수를 입력합니다. 일반적으로 최대 사용 가능한 공간을 설정합니다. 선택(Selected) 아래에 강조된 텍스트는 추가되는 공간의 양이며, 볼륨의 최종 크기가 아닙니다. 마법사를 완료합니다.



7. AWS NVMe 드라이버가 없는 인스턴스에서 NVMe 볼륨의 크기를 늘릴 경우 Windows에서 새 볼륨의 크기를 확인할 수 있도록 인스턴스를 재부팅해야 합니다. AWS NVMe 드라이버 설치에 대한 자세한 내용은 AWS NVMe 드라이버를 참조하세요.

PowerShell

PowerShell을 사용하여 Windows 파일 시스템을 확장하려면 다음 절차를 따릅니다.

PowerShell을 사용하여 파일 시스템을 확장하려면

1. 중요한 데이터가 저장된 파일 시스템을 확장하려면 먼저 변경 내용을 롤백해야 할 경우를 대비하여 파일 시스템이 저장된 볼륨 스냅샷을 생성하는 것이 바람직합니다. 자세한 내용은 Amazon EBS 스냅샷 생성 섹션을 참조하세요.
2. 원격 데스크톱을 사용하여 Windows 인스턴스에 로그인합니다.
3. 관리자 권한으로 PowerShell을 실행합니다.
4. `Get-Partition` 명령을 실행합니다. PowerShell은 각 파티션에 해당하는 파티션 번호, 드라이브 문자, 오프셋, 크기 및 유형을 반환합니다. 확장할 파티션의 드라이브 문자를 확인해 둡니다.
5. 다음 명령을 실행하여 디스크를 다시 검색합니다.

```
"rescan" | diskpart
```

6. **<drive-letter>** 대신 4단계에서 확인한 드라이브 문자를 사용하여 다음 명령을 실행합니다. PowerShell은 파티션의 최소 및 최대 허용 크기를 바이트 단위로 반환합니다.

```
Get-PartitionSupportedSize -DriveLetter <drive-letter>
```

7. 파티션을 지정된 용량으로 확장하려면 **<size>** 대신 볼륨의 새 크기를 입력하여 다음 명령을 실행합니다. 예를 들어 KB, MB 및 GB로 크기(예: 50GB)를 입력할 수 있습니다.

```
Resize-Partition -DriveLetter <drive-letter> -Size <size>
```

파티션을 사용 가능한 최대 크기로 확장하려면 다음 명령을 실행합니다.

```
Resize-Partition -DriveLetter <drive-letter> -Size $(Get-PartitionSupportedSize
 -DriveLetter <drive-letter>).SizeMax
```

다음 PowerShell 명령은 파일 시스템을 특정 크기로 확장하기 위한 전체 명령 및 응답 흐름을 보여줍니다.

```
PS C:\> Get-Partition


   DiskPath: \\?\scsi#disk&ven_nvme&prod_amazon_elastic_b#4&26a12046&0&000000#{53f56307-b6bf-11d0-94f2-00a0c91efb8b}

PartitionNumber  DriveLetter Offset                                        Size Type
---------------  ----------- ------                                        ---- ----
1                C           1048576                                     30 GB IFS


   DiskPath: \\?\scsi#disk&ven_nvme&prod_amazon_elastic_b#4&3476342 3&0&000000#{53f56307-b6bf-11d0-94f2-00a0c91efb8b}

PartitionNumber  DriveLetter Offset                                        Size Type
---------------  ----------- ------                                        ---- ----
1                D           1048576                                     8 MB IFS


PS C:\> "rescan" | diskpart

Microsoft DiskPart version 10.0.17763.1911

Copyright (C) Microsoft Corporation.
On computer:

DISKPART>
Please wait while DiskPart scans your configuration...

DiskPart has finished scanning your configuration.

DISKPART>
PS C:\> Get-PartitionSupportedSize -DriveLetter D

SizeMin      SizeMax
-------      -------
8388608 107372085248


PS C:\> Resize-Partition -DriveLetter D -Size 50GB
PS C:\> Get-Partition


   DiskPath: \\?\scsi#disk&ven_nvme&prod_amazon_elastic_b#4&26a12046&0&000000#{53f56307-b6bf-11d0-94f2-00a0c91efb8b}

PartitionNumber  DriveLetter Offset                                        Size Type
---------------  ----------- ------                                        ---- ----
1                C           1048576                                     30 GB IFS


   DiskPath: \\?\scsi#disk&ven_nvme&prod_amazon_elastic_b#4&3476342 3&0&000000#{53f56307-b6bf-11d0-94f2-00a0c91efb8b}

PartitionNumber  DriveLetter Offset                                        Size Type
---------------  ----------- ------                                        ---- ----
1                D           1048576                                     50 GB IFS
```

다음 PowerShell 명령은 파일 시스템을 사용 가능한 최대 크기로 확장하기 위한 전체 명령 및 응답 흐름을 보여줍니다.

```
PS C:\> Get-Partition


   DiskPath: \\?\scsi#disk&ven_nvme&prod_amazon_elastic_b#4&26a12046&0&000000#{53f56307-b6bf-11d0-94f2-00a0c91efb8b}

PartitionNumber  DriveLetter Offset                                        Size Type
---------------  ----------- ------                                        ---- ----
1                C           1048576                                     30 GB IFS


   DiskPath: \\?\scsi#disk&ven_nvme&prod_amazon_elastic_b#4&34763423&0&000000#{53f56307-b6bf-11d0-94f2-00a0c91efb8b}

PartitionNumber  DriveLetter Offset                                        Size Type
---------------  ----------- ------                                        ---- ----
1                D           1048576                                     50 GB IFS


PS C:\> "rescan" | diskpart

Microsoft DiskPart version 10.0.17763.1911

Copyright (C) Microsoft Corporation.
On computer:

DISKPART>
Please wait while DiskPart scans your configuration...

DiskPart has finished scanning your configuration.

DISKPART>
PS C:\> Get-PartitionSupportedSize -DriveLetter D

 SizeMin      SizeMax
 -------      -------
59047936 107372085248


PS C:\> Resize-Partition -DriveLetter D -Size $(Get-PartitionSupportedSize -DriveLetter D).SizeMax
PS C:\> Get-Partition


   DiskPath: \\?\scsi#disk&ven_nvme&prod_amazon_elastic_b#4&26a12046&0&000000#{53f56307-b6bf-11d0-94f2-00a0c91efb8b}

PartitionNumber  DriveLetter Offset                                        Size Type
---------------  ----------- ------                                        ---- ----
1                C           1048576                                     30 GB IFS


   DiskPath: \\?\scsi#disk&ven_nvme&prod_amazon_elastic_b#4&34763423&0&000000#{53f56307-b6bf-11d0-94f2-00a0c91efb8b}

PartitionNumber  DriveLetter Offset                                        Size Type
---------------  ----------- ------                                        ---- ----
1                D           1048576                                    100 GB IFS
```

Amazon EC2 인스턴스에서 Amazon EBS 볼륨 분리

Amazon Elastic Block Store(Amazon EBS) 볼륨을 다른 인스턴스에 연결하거나 삭제하려면 먼저 인스턴스에서 분리해야 합니다. 볼륨을 분리해도 볼륨의 데이터에는 영향을 주지 않습니다.

주제

- 고려 사항
- 볼륨 마운트 해제 및 분리
- 문제 해결

고려 사항

- 인스턴스에서 Amazon Amazon EBS 볼륨을 분리하거나 인스턴스를 종료하는 것이 가능합니다. 그러나 인스턴스가 실행 중인 경우 인스턴스에서 먼저 해당 볼륨의 마운트를 해제해야 합니다.
- EBS 볼륨이 인스턴스의 루트 디바이스인 경우에는 볼륨을 분리하기 전에 인스턴스를 중지해야 합니다.
- 분리된(탑재를 해제하지 않고) 볼륨을 다시 연결할 수 있지만 동일 탑재 지점을 가져올 수는 없습니다. 분리된 상태에서 진행 중인 볼륨 쓰기 작업이 있으면 볼륨의 데이터가 동기화되지 않을 수 있습니다.
- 볼륨을 분리한 후에도 스토리지 용량이 AWS 프리 티어 한도를 초과하는 한 볼륨 스토리지에 대한 요금이 계속 청구됩니다. 추가 비용이 청구되지 않도록 하려면 볼륨을 삭제해야 합니다. 자세한 내용은 Amazon EBS 볼륨 삭제 섹션을 참조하세요.

볼륨 마운트 해제 및 분리

인스턴스에서 볼륨을 탑재 해제하고 분리하려면 다음 절차를 사용합니다. 이 절차는 볼륨을 다른 인스턴스에 연결해야 하거나 볼륨을 삭제해야 하는 경우 유용할 수 있습니다.

Steps

- 1단계: 볼륨 탑재 해제
- 2단계: 인스턴스에서 볼륨 분리
- 3단계: (Windows 인스턴스만 해당) 오프라인 디바이스 위치 제거

1단계: 볼륨 탑재 해제

Linux 인스턴스

Linux 인스턴스에서 다음 명령을 사용하여 /dev/sdh 디바이스의 탑재를 해제합니다.

```
[ec2-user ~]$ sudo umount -d /dev/sdh
```

Windows 인스턴스

Windows 인스턴스에서 다음과 같이 볼륨을 마운트 해제하세요.

1. 디스크 관리 유틸리티를 시작합니다.

- (Windows Server 2012 이상) 작업 표시줄에서 Windows 로고를 선택하고(마우스 오른쪽 버튼 클릭) 디스크 관리를 선택합니다.
- (Windows Server 2008) [시작(Start)], [관리 도구(Administrative Tools)], [컴퓨터 관리(Computer Management)], [디스크 관리(Disk Management)]를 선택합니다.

2. 디스크를 선택한 다음(예: (마우스 오른쪽 버튼 클릭) 디스크 1) 오프라인을 선택합니다. Amazon EC2 콘솔을 열기 전에 디스크 상태가 오프라인으로 변경될 때까지 기다리십시오.

2단계: 인스턴스에서 볼륨 분리

인스턴스에서 볼륨을 분리하려면 다음 방법 중 하나를 사용합니다.

Console

EBS 볼륨을 분리하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 볼륨을 선택합니다.
3. 볼륨을 선택합니다.
4. 작업, 볼륨 분리를 선택합니다.
5. 확인 메시지가 나타나면 [분리(Detach)]를 선택합니다.

AWS CLI

인스턴스에서 EBS 볼륨을 분리하려면

볼륨 탑재를 해제한 후 detach-volume 명령을 사용합니다.

```
aws ec2 detach-volume --volume-id vol-01234567890abcdef
```

PowerShell

인스턴스에서 EBS 볼륨을 분리하려면

볼륨 탑재를 해제한 후 Dismount-EC2Volume cmdlet을 사용합니다.

```
Dismount-EC2Volume -VolumeId vol-01234567890abcdef
```

3단계: (Windows 인스턴스만 해당) 오프라인 디바이스 위치 제거

인스턴스에서 볼륨을 탑재 해제하고 분리하면 Windows에서 디바이스 위치에 오프라인 플래그가 지정됩니다. 디바이스 위치는 인스턴스 재부팅, 중지 및 다시 시작 후에도 오프라인 상태로 유지됩니다. 인스턴스를 다시 시작하면 Windows에서 나머지 볼륨 중 하나가 오프라인 디바이스 위치에 탑재될 수 있습니다. 이로 인해 Windows에서 볼륨을 사용할 수 없게 됩니다. 이를 방지하고 다음에 Windows를 시작할 때 모든 볼륨이 온라인 디바이스 위치에 연결되도록 하려면 다음 단계를 수행합니다.

1. 인스턴스에서 디바이스 관리자를 엽니다.
2. 디바이스 관리자에서 [보기(View)], [숨겨진 디바이스 표시(Show hidden devices)]를 선택합니다.
3. 디바이스 목록에서 [스토리지 컨트롤러(Storage controllers)] 노드를 확장합니다.

   분리된 볼륨이 탑재된 디바이스 위치의 이름은 `AWS NVMe Elastic Block Storage Adapter`이며 회색으로 표시됩니다.
4. 회색으로 표시된 각 디바이스 위치를 선택하고(마우스 오른쪽 버튼 클릭) 디바이스 제거를 `AWS NVMe Elastic Block Storage Adapter`선택한 다음 제거를 선택합니다.

   > **Important**
   >
   > [이 디바이스의 드라이버 소프트웨어 삭제(Delete the driver software for this device)] 확인란을 선택하지 마세요.

문제 해결

다음에서는 볼륨을 분리할 때 발생할 수 있는 일반적인 문제와 해결 방법에 대해 설명합니다.

> **Note**
>
> 데이터 손실에 대비하여 볼륨을 해제하기 전 볼륨 스냅샷을 만들어 두세요. 고착된 볼륨을 강제로 분리할 경우 파일 시스템 또는 여기에 포함된 데이터가 손상되거나 인스턴스를 재부팅 하지 않는 이상 동일한 디바이스 이름으로 새 볼륨을 연결할 수 없게 될 수 있습니다.

- Amazon EC2 콘솔을 통해 볼륨을 분리하는 동안 문제가 발생할 경우 describe-volumes CLI 명령을 사용하여 문제를 진단하는 것이 좋습니다. 자세한 내용은 describe-volumes를 참조하세요.
- 볼륨이 `detaching` 상태를 유지하는 경우 강제 분리를 선택하여 강제 분리할 수 있습니다. 이 옵션은 오류가 발생한 인스턴에서 볼륨 분리 또는 삭제할 목적으로 볼륨을 분리하는 경우에만 최후의 수

단으로 사용하세요. 인스턴스는 파일 시스템 캐시 또는 파일 시스템 메타데이터를 플러시하지 않습니다. 이 옵션을 사용하는 경우 파일 시스템 확인 및 복구 절차를 수행해야 합니다.

- 몇 분 동안 강제 볼륨 분리를 수차례 시도하였지만 detaching 상태가 계속해서 유지되는 경우 AWS re:Post에 도움을 요청하세요. 해결 방법을 신속히 찾아내려면 볼륨 ID를 기재하고 어떤 단계를 수행했는지에 대해 설명하세요.
- 아직 마운트되어 있는 볼륨을 분리하려는 경우 분리 시도 중에 볼륨이 busy 상태로 고착될 수 있습니다. 다음의 describe-volumes 출력 화면은 이 조건을 보여주는 예입니다.

```
"Volumes": [
    {
        "AvailabilityZone": "us-west-2b",
        "Attachments": [
            {
                "AttachTime": "2022-07-21T23:44:52.000Z",
                "InstanceId": "i-1234567890abcdef0",
                "VolumeId": "vol-01234567890abcdef",
                "State": "busy",
                "DeleteOnTermination": false,
                "Device": "/dev/sdf"
            }
        ...
    }
]
```

이 상태가 발생하면 볼륨의 마운트를 해제하거나 강제 분리하거나 인스턴스를 재부팅하거나 세 가지 조치를 모두 실행하기 전까지 분리가 무한히 지연될 수 있습니다.

Amazon EBS 볼륨 삭제

더 이상 필요하지 않는 Amazon EBS 볼륨을 삭제할 수 있습니다. 볼륨을 삭제한 후에는 데이터가 사라지므로 해당 볼륨을 인스턴스에 연결할 수 없습니다. 따라서 삭제하기 전에 볼륨의 스냅샷을 저장하면 이 스냅샷을 사용하여 나중에 볼륨을 재생성할 수 있습니다.

인스턴스에 연결된 볼륨은 삭제할 수 없습니다. 볼륨을 삭제하려면 먼저 볼륨을 분리해야 합니다. 자세한 내용은 Amazon EC2 인스턴스에서 Amazon EBS 볼륨 분리 섹션을 참조하세요.

휴지통 보존 규칙과 일치하는 볼륨을 삭제하면 볼륨이 즉시 삭제되는 대신 휴지통에 보관됩니다. 자세한 내용은 휴지통을 참조하세요.

Console

EBS 볼륨을 삭제하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 볼륨을 선택합니다.
3. 볼륨을 선택합니다. 볼륨이 사용 가능 상태에 있는지 확인합니다.
4. 작업을 선택한 후 볼륨 삭제을 선택합니다.

   이 옵션이 비활성화된 경우 볼륨이 인스턴스에 연결되며 삭제할 수 없습니다.
5. 확인 메시지가 나타나면 **delete**를 입력한 다음 삭제(Delete)를 선택합니다.

AWS CLI

EBS 볼륨이 사용 중인지 여부를 확인하려면

describe-volumes 명령을 사용합니다. 볼륨이 사용 중인 경우 상태는 `in-use`입니다. 그렇지 않으면 `available`입니다.

```
aws ec2 describe-volumes \
    --volume-id vol-01234567890abcdef \
    --query Volumes[*].State \
    --output text
```

EBS 볼륨을 삭제하려면

delete-volume 명령을 사용합니다.

```
aws ec2 delete-volume --volume-id vol-01234567890abcdef
```

PowerShell

EBS 볼륨이 사용 중인지 여부를 확인하려면

Get-EC2Volume cmdlet을 사용합니다. 볼륨이 사용 중인 경우 상태는 `in-use`입니다. 그렇지 않으면 `available`입니다.

```
(Get-EC2Volume `
```

```
    -VolumeId vol-01234567890abcdef).State.Value
```

EBS 볼륨을 삭제하려면

Remove-EC2Volume cmdlet을 사용합니다.

```
Remove-EC2Volume -VolumeId vol-01234567890abcdef
```

스냅샷을 사용하여 Amazon EBS 볼륨 바꾸기

Amazon EBS 스냅샷은 속도, 편리성 및 비용으로 인해 Amazon EC2에서 선호하는 백업 도구입니다. 스냅샷에서 볼륨을 생성할 경우 특정 시점 그대로 유지한 상태로 저장된 데이터를 특정 시점에서 해당 상태를 재생성합니다. 스냅샷에서 생성된 볼륨을 인스턴스에 연결하면 여러 리전에서 데이터를 복제하고 테스트 환경을 생성하며 손상된 프로덕션 볼륨 전체를 바꾸거나 특정 파일 및 디렉터리를 검색하여 연결된 다른 볼륨으로 전송할 수 있습니다. 자세한 내용은 Amazon EBS 스냅샷 섹션을 참조하세요.

다음 절차 중 하나를 사용하여 Amazon EBS 볼륨을 해당 볼륨의 이전 스냅샷에서 생성된 다른 볼륨으로 대체할 수 있습니다.

요구 사항

인스턴스와 동일한 가용 영역에서 볼륨을 생성해야 합니다. 볼륨은 동일한 가용 영역의 인스턴스에 연결되어야 합니다.

Console

볼륨을 교체하려면

1. 스냅샷에서 볼륨을 생성하고 새 볼륨의 ID를 기록합니다. 자세한 내용은 Amazon EBS 볼륨 생성 섹션을 참조하세요.
2. 인스턴스 페이지에서 볼륨을 교체할 인스턴스를 선택하고 인스턴스 ID를 기록합니다.

   인스턴스가 선택된 상태에서 스토리지(Storage) 탭을 선택합니다. 블록 디바이스(Block devices) 섹션에서 교체할 볼륨을 찾고 볼륨의 디바이스 이름을 기록합니다(예: /dev/sda1).
3. 스토리지 탭에서 볼륨 ID를 선택한 다음 인스턴스에서 볼륨을 탐재 해제하고 분리합니다.
4. 1단계에서 생성한 새 볼륨을 선택하고 작업(Actions), 볼륨 연결(Attach volume)을 선택합니다.

   인스턴스(Instance) 및 디바이스 이름(Device name)에 2단계에서 적어 둔 인스턴스 ID 및 디바이스 이름을 입력한 다음 볼륨 연결(Attach volume)을 선택합니다.

5. 인스턴스에 연결하고 볼륨을 탑재합니다. 자세한 내용은 Amazon EBS 볼륨을 사용할 수 있도록 만들기 섹션을 참조하세요.

AWS CLI

볼륨을 교체하려면

1. 스냅샷에서 새 볼륨을 생성합니다. `--snapshot-id` 옵션과 함께 create-volume 명령을 사용합니다. `--availability-zone`에 인스턴스와 동일한 가용 영역을 지정합니다. 출력의 새 볼륨 ID를 메모해 둡니다.

```
aws ec2 create-volume \
    --volume-type gp3 \
    --snapshot-id snap-0abcdef1234567890 \
    --availability-zone us-east-1a
```

2. 대체할 볼륨의 디바이스 이름을 가져옵니다. 아래와 같이 describe-instances 명령을 사용합니다. `--instance-ids`에 볼륨을 대체할 인스턴스의 ID를 지정합니다. 대체할 볼륨의 디바이스 이름과 볼륨 ID를 적어둡니다.

```
aws ec2 describe-instances \
    --instance-ids i-1234567890abcdef0 \
    --query Reservations[].Instances[].BlockDeviceMappings
```

3. 인스턴스에서 대체할 볼륨을 분리합니다. detach-volume 명령을 사용합니다.

```
aws ec2 detach-volume --volume-id vol-xxxxxxxxxxxxxxxxx
```

4. 인스턴스에 대체 볼륨을 연결합니다. attach-volume 명령을 사용합니다. `--volume-id`에 대체 볼륨의 ID를 지정합니다. `--instance-id`에 볼륨을 연결할 인스턴스의 ID를 지정합니다. `--device`에 이전에 기록한 것과 동일한 디바이스 이름을 지정합니다.

```
aws ec2 attach-volume \
    --volume-id vol-01234567890abcdef \
    --instance-id i-1234567890abcdef0 \
    --device /dev/sdf
```

5. 인스턴스에 연결하고 볼륨을 탑재합니다. 자세한 내용은 Amazon EBS 볼륨을 사용할 수 있도록 만들기 섹션을 참조하세요.

PowerShell

볼륨을 교체하려면

1. 스냅샷에서 새 볼륨을 생성합니다. New-EC2Volume cmdlet을 -SnapshotId 옵션과 함께 사용합니다. -AvailabilityZone에 인스턴스와 동일한 가용 영역을 지정합니다. 출력의 새 볼륨 ID를 메모해 둡니다.

```
New-EC2Volume `
    -VolumeType gp3 `
    -SnapshotId snap-0abcdef1234567890 `
    -AvailabilityZone us-east-1a
```

2. 대체할 볼륨의 디바이스 이름을 가져옵니다. Get-EC2Instance cmdlet을 사용합니다. -InstanceId에 볼륨을 대체할 인스턴스의 ID를 지정합니다. 대체할 볼륨의 디바이스 이름과 볼륨 ID를 적어둡니다.

```
(Get-EC2Instance `
    -InstanceId i-1234567890abcdef0).Instances.BlockDeviceMappings | `
     Format-Table DeviceName, @{Name="VolumeId";Expression={$_.Ebs.VolumeId}}
```

3. 인스턴스에서 대체할 볼륨을 분리합니다. Dismount-EC2Volume cmdlet을 사용합니다.

```
DismountEC2Volume -VolumeId vol-xxxxxxxxxxxxxxxxx
```

4. 인스턴스에 대체 볼륨을 연결합니다. Add-EC2Volume cmdlet을 사용합니다. -VolumeId에 대체 볼륨의 ID를 지정합니다. -InstanceId에 볼륨을 연결할 인스턴스의 ID를 지정합니다. -Device에 이전에 기록한 것과 동일한 디바이스 이름을 지정합니다.

```
Add-EC2Volume`
    -VolumeId vol-01234567890abcdef `
    -InstanceId i-1234567890abcdef0 `
    -Device /dev/sdf
```

5. 인스턴스에 연결하고 볼륨을 탑재합니다. 자세한 내용은 Amazon EBS 볼륨을 사용할 수 있도록 만들기 섹션을 참조하세요.

Amazon EBS 볼륨 상태 확인

볼륨 상태 확인을 사용하여 Amazon EBS 볼륨에 있는 데이터의 잠재적 불일치를 더 잘 파악, 추적 및 관리할 수 있습니다. 볼륨 상태 확인은 Amazon EBS 볼륨이 손상되었는지 여부를 확인하는 데 필요한 정보를 제공하며, 잠재적으로 일치하지 않는 볼륨을 처리하는 방법을 제어하는 데 도움이 됩니다.

볼륨 상태 확인은 5분마다 테스트를 자동으로 실행하여 통과 또는 실패 상태를 반환합니다. 모든 확인을 통과한 경우 볼륨의 상태는 `ok`이고, 확인에 실패한 경우 볼륨의 상태는 `impaired`입니다. 상태가 `insufficient-data`인 경우 볼륨에 대한 확인이 아직 진행 중일 수 있습니다. 볼륨 상태 확인의 결과를 보고 손상된 볼륨을 식별하고 필요한 조치를 취할 수 있습니다.

Amazon EBS에서 볼륨의 데이터가 잠재적으로 일치하지 않는 것으로 확인하면 데이터 손상을 방지하기 위해 기본적으로 연결된 EC2 인스턴스에서 볼륨으로의 I/O가 비활성화됩니다. I/O가 비활성화되면 다음 볼륨 상태 확인에 실패하고 볼륨 상태는 `impaired`가 됩니다. 또한 I/O가 비활성화되었으며 볼륨에 대한 I/O를 활성화하여 볼륨의 손상된 상태를 해결할 수 있다고 알려주는 이벤트가 표시됩니다. EBS는 I/O를 활성화하여 인스턴스가 볼륨을 계속 사용하도록 허용할지 또는 fsck (Linux 인스턴스) 또는 chkdsk (Windows 인스턴스)와 같은 명령을 사용하여 일관성 검사를 실행할지 결정할 수 있는 기회를 제공할 때까지 기다린 후 이를 수행합니다.

> **Note**
>
> 볼륨 상태는 볼륨 상태 검사 결과를 기준으로 한 것으로, 볼륨 상태를 직접 반영하는 것은 아닙니다. 따라서 볼륨 상태가 `error` 상태의 볼륨을 나타내는 것은 아닙니다(예: 볼륨이 I/O를 허용할 수 없을 때). 볼륨 상태에 대한 자세한 내용은 볼륨 상태 섹션을 참조하세요.

특정 볼륨의 일관성은 문제가 아니고, 볼륨이 손상된 경우 볼륨을 즉시 사용할 수 있게 하려면 I/O를 자동으로 사용하도록 볼륨을 구성하여 기본 동작을 무시할 수 있습니다. [IO 자동 사용(Auto-Enable IO)] 볼륨 속성(API의 `autoEnableIO`)을 사용하면 볼륨 상태 확인이 계속해서 통과됩니다. 또한 볼륨이 잠재적으로 일치하지 않는 것으로 확인되었지만 I/O가 자동으로 활성화되었다고 알려주는 이벤트가 표시됩니다. 그러면 볼륨의 일관성을 확인하거나 나중에 볼륨을 교체할 수 있습니다.

I/O 성능 상태 확인은 실제 볼륨 성능과 볼륨의 예상 성능을 비교합니다. 볼륨 성능이 예상보다 낮은 경우 알림을 제공합니다. 이 상태 확인은 인스턴스에 연결된 프로비저닝된 IOPS SSD(`io1` 및 `io2`) 및 범용 SSD(`gp3`) 볼륨에만 사용할 수 있습니다. 범용 SSD(`gp2`), 처리량 최적화 HDD(`st1`), 콜드 HDD(`sc1`) 또는 마그네틱(`standard`) 볼륨에는 상태 확인이 유효하지 않습니다. I/O 성능 상태 확인은 1분에 한 번씩 수행되며 CloudWatch는 이 데이터를 5분 간격으로 수집합니다. `io1` 또는 `io2` 볼륨을 인스턴스에 연결한 후 상태 확인에서 I/O 성능 상태를 보고하기까지 최대 5분이 소요될 수 있습니다.

> **Important**
>
> 스냅샷에서 복원한 Provisioned IOPS SSD 볼륨을 초기화할 경우 볼륨의 성능이 예상 수준보다 50퍼센트 이하로 떨어질 수 있으며, 이로 인해 볼륨의 I/O 성능 상태 확인에 `warning` 상태가 표시될 수 있습니다. 이는 예상된 동작이므로 초기화 중에는 Provisioned IOPS SSD 볼륨에 대한 `warning` 상태를 무시해도 됩니다. 자세한 내용은 볼륨 생성 후 수동 초기화 섹션을 참조하세요.

다음 표에는 Amazon EBS 볼륨에 대한 상태가 나와 있습니다.

볼륨 상태	I/O 활성화 상태	I/O 성능 상태(**io1**, **io2** 및 **gp3** 볼륨에만 해당)
`ok`	활성화됨(I/O 활성화 또는 I/O 자동 활성화)	정상(볼륨 성능이 예상대로임)
`warning`	활성화됨(I/O 활성화 또는 I/O 자동 활성화)	성능 저하(볼륨 성능이 예상보다 낮음) 심각한 성능 저하(볼륨 성능이 예상보다 훨씬 낮음)
`impaired`	활성화됨(I/O 활성화 또는 I/O 자동 활성화) 비활성화됨(볼륨이 오프라인 상태이고 복구 보류 중이거나 I/O 활성화를 기다리고 있음)	중단됨(볼륨 성능이 저하됨) 사용할 수 없음(I/O가 비활성화되어 I/O 성능을 확인할 수 없음)
`insufficient-data`	활성화됨(I/O 활성화 또는 I/O 자동 활성화) 데이터 부족	데이터 부족

Console

상태 확인 보기

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 볼륨을 선택합니다.

   볼륨 상태(Volume status_ 열에 각 볼륨의 작업 상태가 나열됩니다.
3. 특정 볼륨의 상태 세부 정보를 보려면 그리드를 선택하고 상태 검사(Status checks) 탭을 선택합니다.
4. 상태 확인에 실패한 볼륨이 있는 경우(`impaired` 상태) 손상된 Amazon EBS 볼륨 작업 섹션을 참조하세요.

또는 탐색기에서 이벤트 창을 선택하여 인스턴스와 볼륨에 대한 모든 이벤트를 볼 수 있습니다. 자세한 내용은 Amazon EBS 볼륨 이벤트 섹션을 참조하세요.

AWS CLI

볼륨 상태 정보를 보기

describe-volume-status 명령을 사용합니다.

```
aws ec2 describe-volume-status --volume-ids vol-01234567890abcdef
```

다음 예제를 사용하여 손상된 볼륨을 식별합니다.

```
aws ec2 describe-volume-status --filters Name=volume-status.status,Values=impaired
```

PowerShell

볼륨 상태 정보를 보기

Get-EC2VolumeStatus cmdlet을 사용합니다.

```
Get-EC2VolumeStatus -VolumeId vol-01234567890abcdef
```

다음 예제를 사용하여 손상된 볼륨을 식별합니다.

```
Get-EC2VolumeStatus -Filter @{Name="volume-status.status"; Values="impaired"}
```

Amazon EBS 볼륨 이벤트

Amazon EBS에서 볼륨의 데이터가 잠재적으로 일치하지 않는 것으로 확인하면 기본적으로 연결된 EC2 인스턴스에서 볼륨으로의 I/O가 비활성화됩니다. 그러면 볼륨 상태 확인에 실패하고 실패의 원인을 나타내는 볼륨 상태 이벤트가 생성됩니다.

데이터가 잠재적으로 일치하지 않는 볼륨에서 I/O를 자동으로 활성화하려면 IO 자동 활성화 볼륨 속성(API의 `autoEnableIO`)의 설정을 변경합니다. 이 속성 변경에 대한 자세한 내용은 손상된 Amazon EBS 볼륨 작업 섹션을 참조하세요.

각 이벤트에는 이벤트가 발생한 시간을 나타내는 시작 시간과 볼륨에 대한 I/O가 비활성화된 시간을 나타내는 기간이 포함됩니다. 볼륨에 대한 I/O가 활성화되면 이벤트에 종료 시간이 추가됩니다.

볼륨 상태 이벤트

`Awaiting Action: Enable IO`

볼륨 데이터가 잠재적으로 일치하지 않습니다. 사용자가 명시적으로 활성화할 때까지 볼륨에 대해 I/O가 비활성화됩니다. I/O를 명시적으로 활성화하면 이벤트 설명이 IO Enabled로 변경됩니다.

`IO Enabled`

이 볼륨에 대해 I/O 작업이 명시적으로 활성화되었습니다.

`IO Auto-Enabled`

이벤트가 발생한 후 이 볼륨에서 I/O 작업이 자동으로 활성화되었습니다. 데이터를 계속 사용하려면 먼저 데이터 불일치를 확인하는 것이 좋습니다.

`Normal`

`io1`, `io2` 및 `gp3` 볼륨에만 해당합니다. 볼륨 성능이 예상대로입니다.

`Degraded`

`io1`, `io2` 및 `gp3` 볼륨에만 해당합니다. 볼륨 성능이 예상보다 낮습니다.

`Severely Degraded`

`io1`, `io2` 및 `gp3` 볼륨에만 해당합니다. 볼륨 성능이 예상보다 훨씬 낮습니다.

`Stalled`

`io1`, `io2` 및 `gp3` 볼륨에만 해당합니다. 볼륨 성능이 저하되었습니다.

I/O가 비활성화된 볼륨이 있는 경우 손상된 Amazon EBS 볼륨 작업 섹션을 참조하세요. I/O 성능이 정상보다 낮은 볼륨이 있는 경우 수행한 작업(예: 사용량이 많은 동안 볼륨의 스냅샷 생성, 필요한 I/O 대역폭을 지원할 수 없는 인스턴스에서 볼륨 실행, 볼륨의 데이터에 처음 액세스하는 등)으로 인해 일시적인 조건일 수 있습니다.

Console

볼륨에 대한 이벤트 보기

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [Events]를 선택합니다. 이벤트가 있는 모든 인스턴스와 볼륨이 나열됩니다.
3. 볼륨을 기준으로 필터링하여 볼륨 상태만 볼 수 있습니다. 특정 상태 유형을 기준으로 필터링할 수도 있습니다.
4. 특정 이벤트를 보려는 볼륨을 선택합니다.

AWS CLI

볼륨에 대한 이벤트 보기

describe-volume-status 명령을 사용합니다.

```
aws ec2 describe-volume-status --volume-ids vol-01234567890abcdef
```

PowerShell

볼륨에 대한 이벤트 보기

Get-EC2VolumeStatus cmdlet을 사용합니다.

```
Get-EC2VolumeStatus -VolumeId vol-01234567890abcdef
```

손상된 Amazon EBS 볼륨 작업

볼륨의 데이터가 잠재적으로 일치하지 않아서 볼륨이 손상된 경우 다음 옵션을 사용합니다.

옵션

- 옵션 1: 인스턴스에 연결된 볼륨에 대한 일관성 확인 수행

- 옵션 2: 다른 인스턴스를 사용하여 볼륨에 대한 일관성 확인 수행
- 옵션 3: 볼륨이 더 이상 필요하지 않은 경우 볼륨 삭제

옵션 1: 인스턴스에 연결된 볼륨에 대한 일관성 확인 수행

가장 간단한 옵션은 볼륨이 Amazon EC2 인스턴스에 연결된 상태에서 I/O를 활성화한 다음 볼륨에 대한 데이터 일관성 확인을 수행하는 것입니다.

연결된 볼륨에 대해 일관성 확인을 수행하려면

1. 모든 애플리케이션의 볼륨 사용을 중지합니다.
2. 볼륨에서 I/O를 활성화합니다. 다음 방법 중 하나를 사용합니다.

   Console

   볼륨에 대한 I/O 활성화

   1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
   2. 탐색 창에서 [이벤트(Events)]를 선택합니다.
   3. 볼륨을 선택합니다.
   4. 작업(Actions), I/O 활성화(Enable I/O)를 선택합니다.

   AWS CLI

   볼륨에 대한 I/O 활성화

   enable-volume-io 명령을 사용합니다.

   ```
   aws ec2 enable-volume-io --volume-id vol-01234567890abcdef
   ```

   PowerShell

   볼륨에 대한 I/O 활성화

   Enable-EC2VolumeIO cmdlet을 사용합니다.

   ```
   Enable-EC2VolumeIO -VolumeId vol-01234567890abcdef
   ```

3. 볼륨의 데이터를 확인합니다.

a. fsck(Linux 인스턴스) 또는 chkdsk(Windows 인스턴스) 명령을 실행합니다.

b. (선택 사항) 애플리케이션 또는 시스템 로그에 관련 오류 메시지가 있는지 검토합니다.

c. 볼륨이 20분 이상 손상된 경우 AWS 지원 센터에 문의할 수 있습니다. 문제 해결을 선택한 다음 상태 검사 문제 해결 대화 상자에서 고객 지원을 선택하여 지원 사례를 제출합니다.

옵션 2: 다른 인스턴스를 사용하여 볼륨에 대한 일관성 확인 수행

다음 절차에 따라 프로덕션 환경 외부의 볼륨을 확인합니다.

> **Important**
>
> 이 절차를 수행하면 볼륨 I/O가 비활성화된 상태에서 일시 중지된 쓰기 I/O가 손실될 수 있습니다.

격리 중인 볼륨에 대한 일관성 확인을 수행하려면

1. 모든 애플리케이션의 볼륨 사용을 중지합니다.
2. 인스턴스에서 볼륨을 분리합니다. 자세한 내용은 Amazon EC2 인스턴스에서 Amazon EBS 볼륨 분리 섹션을 참조하세요.
3. 볼륨에서 I/O를 활성화합니다. 다음 방법 중 하나를 사용합니다.

   Console

   볼륨에 대한 I/O 활성화

   1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
   2. 탐색 창에서 [이벤트(Events)]를 선택합니다.
   3. 이전 단계에서 분리한 볼륨을 선택합니다.
   4. 작업(Actions), I/O 활성화(Enable I/O)를 선택합니다.

   AWS CLI

   볼륨에 대한 I/O 활성화

   enable-volume-io 명령을 사용합니다.

```
aws ec2 enable-volume-io --volume-id vol-01234567890abcdef
```

PowerShell

볼륨에 대한 I/O 활성화

Enable-EC2VolumeIO cmdlet을 사용합니다.

```
Enable-EC2VolumeIO -VolumeId vol-01234567890abcdef
```

4. 볼륨을 다른 인스턴스에 연결합니다. 자세한 내용은 인스턴스 시작 및 Amazon EC2 인스턴스에 Amazon EBS 볼륨 연결 섹션을 참조하세요.
5. 볼륨의 데이터를 확인합니다.

   a. fsck(Linux 인스턴스) 또는 chkdsk(Windows 인스턴스) 명령을 실행합니다.
   b. (선택 사항) 애플리케이션 또는 시스템 로그에 관련 오류 메시지가 있는지 검토합니다.
   c. 볼륨이 20분 이상 손상된 경우 AWS 지원 센터에 문의할 수 있습니다. 문제 해결을 선택하고 문제 해결 대화 상자에서 고객 지원을 선택하여 지원 사례를 제출합니다.

옵션 3: 볼륨이 더 이상 필요하지 않은 경우 볼륨 삭제

환경에서 볼륨을 제거하려면 볼륨을 삭제하면 됩니다. 볼륨 삭제에 대한 자세한 내용은 Amazon EBS 볼륨 삭제 섹션을 참조하세요.

볼륨의 데이터를 백업하는 최근 스냅샷이 있는 경우 해당 스냅샷에서 새 볼륨을 생성할 수 있습니다. 자세한 내용은 Amazon EBS 볼륨 생성 섹션을 참조하세요.

손상된 Amazon EBS 볼륨에 대해 I/O 자동 활성화

Amazon EBS에서 볼륨의 데이터가 잠재적으로 일치하지 않는 것으로 확인하면 기본적으로 연결된 EC2 인스턴스에서 볼륨으로의 I/O가 비활성화됩니다. 그러면 볼륨 상태 확인에 실패하고 실패의 원인을 나타내는 볼륨 상태 이벤트가 생성됩니다. 특정 볼륨의 일관성은 문제가 아니고, 볼륨이 손상된 경우 볼륨을 즉시 사용할 수 있게 하려면 I/O를 자동으로 사용하도록 볼륨을 구성하여 기본 동작을 무시할 수 있습니다. [IO 자동 사용(Auto-Enable IO)] 볼륨 속성(API의 `autoEnableIO`)을 사용하면 볼륨과 인스턴스 간 I/O가 자동으로 다시 사용되고 볼륨 상태 확인이 계속해서 통과됩니다. 또한 볼륨이 잠재적으로 일치하지 않는 상태인 것으로 결정되었지만 I/O가 자동으로 활성화되었다고 알려주는 이벤트가 표시됩니다. 이 이벤트가 발생하면 볼륨의 일관성을 확인하고 필요한 경우 볼륨을 교체해야 합니다. 자세한 내용은 Amazon EBS 볼륨 이벤트 섹션을 참조하세요.

Console

볼륨의 자동 활성화 IO 속성 보기

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 볼륨을 선택합니다.
3. 볼륨을 선택하고 상태 검사(Status checks)를 선택합니다.

   자동 활성화된 I/O(Auto-Enabled I/O) 필드는 선택된 볼륨에 대한 현재 설정(활성화됨(Enabled) 또는 비활성됨(Disabled))을 표시합니다.

볼륨의 자동 활성화 IO 속성 수정

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 볼륨을 선택합니다.
3. 볼륨을 선택하고 작업(Actions), I/O 자동 활성화 관리(Manage auto-enabled I/O)를 선택합니다.
4. 손상된 볼륨에 대한 I/O를 자동으로 활성화하려면 손상된 볼륨에 대한 IO 자동 활성화(Auto-enable I/O for impaired volumes) 확인란을 선택합니다. 이 기능을 비활성화하려면 확인란의 선택을 취소합니다.
5. 업데이트를 선택합니다.

AWS CLI

볼륨의 autoEnableIO 속성 보기

describe-volume-attribute 명령을 사용합니다.

```
aws ec2 describe-volume-attribute \
    --attribute autoEnableIO \
    --volume-id vol-01234567890abcdef
```

다음은 예제 출력입니다.

```
{
    "AutoEnableIO": {
        "Value": true
    },
```

```
    "VolumeId": "vol-01234567890abcdef"
}
```

볼륨의 autoEnableIO IO 속성 수정

modify-volume-attribute 명령을 사용합니다.

```
aws ec2 modify-volume-attribute \
    --auto-enable-io \
    --volume-id vol-01234567890abcdef
```

PowerShell

볼륨의 autoEnableIO 속성 보기

Get-EC2VolumeAttribute cmdlet을 사용합니다.

```
(Get-EC2VolumeAttribute `
    -Attribute autoEnableIO `
    -VolumeId vol-01234567890abcdef).AutoEnableIO
```

다음은 예제 출력입니다.

```
True
```

볼륨의 autoEnableIO IO 속성 수정

Edit-EC2VolumeAttribute cmdlet을 사용합니다.

```
Edit-EC2VolumeAttribute `
    -AutoEnableIO $true `
    -VolumeId vol-01234567890abcdef
```

Amazon EBS에서 오류 테스트

AWS Fault Injection Service (AWS FIS)는 AWS 워크로드에서 결함 주입 실험을 수행하는 데 도움이 되는 완전 관리형 서비스입니다. 에서 EBS 작업을 사용하면 애플리케이션이 스토리지 장애에 어떻게 반응하여 볼륨에서 I/O 중단 및 성능 저하를 초래할 수 있는지 테스트 AWS FIS할 수 있습니다. 이 제어된 테스트 환경을 사용하면 애플리케이션이 장애에 어떻게 대응하는지 관찰하여 아키텍처의 약점을

식별하고 애플리케이션의 전반적인 복원력을 개선할 수 있습니다. I/O 일시 중지 작업과 지연 시간 주입 작업을 사용하여 Amazon CloudWatch 경보 및 장애 조치 워크플로와 같은 모니터링 및 복구 메커니즘을 테스트하고, 스토리지 장애에 대한 미션 크리티컬 애플리케이션의 복원력을 개선할 수 있습니다. 에 대한 자세한 내용은 AWS Fault Injection Service 사용 설명서를 AWS FIS참조하세요.

사용 가능한 실험

Amazon EBS는 현재 두 가지 AWS FIS 장애 주입을 지원합니다.

- I/O 오류 주입 일시 중지
- 지연 시간 주입

고려 사항

다음 사항을 고려하세요.

- 모든 Amazon EBS 볼륨 유형이 지원됩니다. 루트 볼륨과 데이터 볼륨이 모두 지원됩니다. 인스턴스 스토어 볼륨이 지원되지 않습니다.
- 볼륨은 Nitro 기반 EC2 인스턴스에 연결해야 합니다.
- 실험이 지속 시간에 따라 완료되고 나면 볼륨이 원래 I/O 성능을 재개합니다. 실행 중인 실험을 완료 전에 중지할 수도 있습니다. 또는 CloudWatch 경보에서 정의한 임계값에 도달하는 경우 실험을 중지하는 중지 조건을 생성할 수 있습니다.
- 다중 연결 지원 볼륨 AWS FIS 에서를 사용할 수 있습니다. 연결된 인스턴스가 모두 영향을 받습니다. 실험에 대해 특정 볼륨-인스턴스 연결만 선택할 수는 없습니다.
- FIS는 현재 로컬 영역, Outpost 또는 Wavelength Zone에서 사용할 수 없습니다.
- 콘솔에서 볼륨 ARN을 지정할 때 동일한 가용 영역에서 최대 5개의 볼륨을 동시에 테스트할 수 있습니다.
- , AWS Wavelength 영역 또는 로컬 영역에서 생성된 AWS FIS 볼륨Outpost에는를 사용할 수 없습니다.

I/O 오류 주입 일시 중지

AWS Fault Injection Service 및 I/O 일시 중지 작업을 사용하여 Amazon EBS 볼륨과 연결된 인스턴스 간의 I/O를 일시적으로 중지하여 워크로드가 I/O 중단을 처리하는 방법을 테스트합니다.

에 대한 자세한 내용은 AWS Fault Injection Service 사용 설명서를 AWS FIS참조하세요.

고려 사항

볼륨 I/O 일시 중지에 대한 다음 고려 사항에 유의하세요.

- I/O 일시 중지는 모든 Nitro 기반 인스턴스 유형에서 지원됩니다.
- OS 제한 시간 구성을 테스트하려면 실험 기간을 `nvme_core.io_timeout`에 지정된 값 이상으로 설정합니다. 자세한 내용은 Amazon EBS 볼륨에 대한 NVMe I/O 작업 시간 제한 섹션을 참조하세요.
- I/O가 일시 중지된 볼륨으로 I/O를 구동하면 다음과 같은 상황이 발생합니다.
  - 볼륨 상태가 120초 이내에 `impaired`로 전환됩니다. 자세한 내용은 Amazon EBS 볼륨 상태 확인 섹션을 참조하세요.
  - 볼륨 I/O가 60초를 초과하여 일시 중지된 경우, `VolumeStalledIOCheck`에 대한 CloudWatch 지표는 `1`입니다. 자세한 내용은 Amazon EBS 볼륨 지표 섹션을 참조하세요.
  - 대기열 길이(`VolumeQueueLength`)에 대한 CloudWatch 지표가 `0`이 아닙니다. 모든 경보 또는 모니터링에서 `0`이 아닌 대기열 깊이를 모니터링해야 합니다.
  - `VolumeReadOps` 또는 `VolumeWriteOps`에 대한 CloudWatch 지표가 `0`입니다. 이는 볼륨이 더 이상 I/O를 처리하지 않음을 나타냅니다.

Amazon EC2 콘솔에서 기본 실험을 수행하거나 AWS FIS 콘솔을 사용하여 고급 실험을 수행할 수 있습니다. AWS FIS 콘솔을 사용하여 고급 실험을 수행하는 방법에 대한 자세한 내용은 AWS Fault Injection Service 사용 설명서의 자습서를 참조하세요 AWS FIS.

Amazon EC2 콘솔을 사용하여 기본 실험 수행

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 볼륨을 선택합니다.
3. I/O를 일시 중지할 볼륨을 선택하고 작업, 오류 주입, 볼륨 I/O 일시 중지를 선택합니다.
4. 기간에 볼륨과 인스턴스 간의 I/O를 일시 중지할 기간을 입력합니다. 기간 드롭다운 목록 옆의 필드에는 기간이 ISO 8601 형식으로 표시됩니다.
5. 서비스 액세스 섹션에서가 실험을 수행하기 위해 수임 AWS FIS 할에 대한 IAM 서비스 역할을 선택합니다. 기본 역할을 사용하거나 생성한 기존 역할을 사용할 수 있습니다. 자세한 내용은 AWS FIS 실험을 위한 IAM 역할을 참조하세요.
6. 볼륨 I/O 일시 중지를 선택합니다. 메시지가 나타나면 확인 필드에 `start`를 입력하고 실험 시작을 선택합니다.
7. 실험의 진행 상황과 영향을 모니터링합니다. 자세한 내용은 AWS FIS 사용 설명서의 AWS FIS모니터링을 참조하세요.

지연 시간 주입

의 지연 시간 주입 작업(`aws:ebs:volume-io-latency`) AWS FIS 을 사용하여 Amazon EBS 볼륨에서 늘어난 I/O 지연 시간을 시뮬레이션하여 애플리케이션이 스토리지 성능 저하에 어떻게 반응하는지 테스트합니다. 이 작업을 사용하면 주입될 지연 시간 값과 대상 볼륨에 영향을 미칠 I/O의 백분율을 지정할 수 있습니다. 를 사용하면 사전 구성된 지연 시간 실험 템플릿을 AWS FIS사용하여 스토리지 장애 발생 시 관찰될 수 있는 다양한 I/O 지연 시간 패턴을 테스트할 수 있습니다. 이러한 템플릿은 복원력 테스트를 위해 애플리케이션에 중단을 일으키는 데 사용할 수 있는 초기 시나리오 세트로 설계되었습니다. 실제 환경에서 애플리케이션이 경험할 수 있는 모든 유형의 영향을 포함하도록 설계된 것은 아닙니다. 애플리케이션의 성능 요구 사항에 따라 여러 가지의 테스트를 실행하도록 조정할 것이 권장됩니다. 사용 가능한 템플릿을 사용자 지정하거나 새로운 실험 템플릿을 생성하여 애플리케이션의 고유한 요구 사항을 테스트할 수 있습니다.

사전 구성된 지연 시간 실험 템플릿

Amazon EBS는 EBS 콘솔 및 AWS FIS 시나리오 라이브러리를 통해 다음과 같은 지연 시간 실험 템플릿을 제공합니다. 이러한 템플릿을 대상 볼륨에 직접 사용하여 지연 시간 주입 실험을 실행할 수 있습니다.

- 지속 지연 시간 - 일정한 지연 시간을 시뮬레이션합니다. 이 실험은 단일 지연 시간 주입 작업을 활용하며 총 시간은 15분입니다. 이 실험은 15분 동안 50%의 읽기 I/O와 100%의 쓰기 I/O에서 500ms의 지속 지연 시간을 시뮬레이션합니다.
- 증가 지연 시간 - 점진적으로 증가하는 지연 시간을 시뮬레이션합니다. 이 실험은 5개의 지연 시간 주입 작업을 활용하며 총 시간은 15분입니다. 이 실험은 10%의 읽기 I/O와 25%의 쓰기 I/O에서 점진적으로 증가하는 지연 시간(3분 동안 50ms, 3분 동안 200ms, 3분 동안 700ms, 3분 동안 1초 및 3분 동안 15초)을 시뮬레이션합니다.
- 간헐적 지연 시간 - 급격한 간헐적 지연 시간 스파이크와 그 사이의 복구 기간을 시뮬레이션합니다. 이 실험은 3개의 지연 시간 주입 작업을 활용하며 총 시간은 15분입니다. 이 실험은 0.1%의 읽기 및 쓰기 I/O에서 3개의 지연 시간 스파이크(1분 동안 지속되는 30초 스파이크, 2분 동안 지속되는 10초 스파이크 및 2분 동안 지속되는 20초 스파이크)를 시뮬레이션합니다. 각 지연 시간 스파이크 사이에는 5분의 복구 기간을 둡니다.
- 감소 지연 시간- 점진적으로 감소되는 지연 시간을 시뮬레이션합니다. 이 실험은 5개의 지연 시간 주입 작업을 활용하며 총 시간은 15분입니다. 이 실험은 10%의 읽기 I/O 및 쓰기 I/O에서 점진적으로 감소하는 지연 시간(3분 동안 20초, 3분 동안 5초, 3분 동안 900ms, 3분 동안 300ms 및 3분 동안 40ms)을 시뮬레이션합니다.

사전 구성된 시나리오의 사용자 지정

위의 사전 구성된 템플릿을 사용자 지정하거나 다음과 같은 사용자 지정 가능한 파라미터를 사용하여 새로운 실험 템플릿을 생성합니다.

- `readIOPercentage` - 지연 시간이 주입될 읽기 I/O 작업의 백분율. 이는 작업의 영향을 받을 볼륨에 대한 모든 읽기 I/O 작업의 백분율입니다.

  범위: 최소 0.1%, 최대 100%

- `readIOLatencyMilliseconds` - 읽기 I/O 작업에 주입될 지연 시간. 이는 실험 중에 읽기 I/O의 지정된 백분율에서 관찰될 지연 시간 값입니다.

  범위: 최소 1ms(io2)/10ms(비io2), 최대 60초

- `writeIOPercentage` - 지연 시간이 주입될 쓰기 I/O 작업의 백분율. 이는 작업의 영향을 받을 볼륨에 대한 모든 쓰기 I/O 작업의 백분율입니다.

  범위: 최소 0.1%, 최대 100%

- `writeIOLatencyMilliseconds` - 쓰기 I/O 작업에 주입될 지연 시간. 이는 실험 중에 쓰기 I/O의 지정된 백분율에서 관찰될 지연 시간 값입니다.

  범위: 최소 1ms(io2)/10ms(비io2), 최대 60초

- `duration` - 선택한 I/O의 백분율에 지연 시간이 주입되는 기간입니다.

  범위: 최소 1초, 최대 12시간

지연 시간 주입 모니터링

볼륨에 대한 성능 영향은 다음과 같은 방법으로 모니터링할 수 있습니다.

- CloudWatch의 평균 지연 시간 지표를 사용하여 분당 평균 I/O 지연 시간을 얻습니다. 자세한 내용을 알아보려면 CloudWatch를 사용하여 EBS 볼륨 모니터링을 참조하세요.
- NVMe-CLI, CloudWatch 에이전트 및 Prometheus를 통해 사용 가능한 EBS 세부 성능 통계를 사용하여 초당 평균 I/O 지연 시간을 얻습니다. 이 세부 지표는 볼륨의 지연 시간 변동을 분석하는 데 사용할 수 있는 I/O 지연 시간 히스토그램도 제공합니다. 자세한 내용을 알아보려면 NVMe 세부 성능 통계를 참조하세요.
- Amazon EBS 볼륨 상태 확인를 사용합니다. I/O 지연 시간을 주입하면 볼륨의 상태가 `warning` 상태로 전환됩니다.

고려 사항

EBS 지연 시간 주입을 사용할 때는 다음 사항을 고려하세요.

- 지연 시간 주입은 P4d, P5, P5e, Trn2u, G6, G6f, Gr6, Gr6f, M8i, M8i-flex, C8i-flex, R8i, R8i-flex, I8ge, Mac-m4pro, Mac-m4를 제외한 모든 Nitro 기반 인스턴스 유형에서 지원됩니다.
- 실험에 지정된 지연 시간 값과 관찰된 지연 시간에서 최대 5%의 차이가 표시될 수 있습니다.
- 매우 적은 수의 I/O 작업을 실행하는 경우 작업 파라미터에 지정된 I/O 비율이 작업의 영향을 받는 I/O의 실제 백분율과 일치하지 않을 수 있습니다.

Amazon EBS 볼륨에서 지연 시간 주입 실험을 실행하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 볼륨을 선택합니다.
3. 실험을 실행할 볼륨을 선택하고 작업, 복원력 테스트, 볼륨 I/O 지연 시간 주입을 선택합니다.

   AWS Fault Injection Service 콘솔이 열립니다.
4. 실험 생성 창에서, 실행할 실험 유형(간헐적, 증가, 지속 또는 감소)을 선택합니다.
5. IAM 역할 선택에서 새 역할 생성을 선택하여 AWS FIS 가 사용자를 대신하여 실험을 수행하는 데 사용할 새 역할을 생성합니다. 또는 필요한 권한이 있는 IAM 역할을 이전에 생성한 경우 기존 IAM 역할 사용을 선택합니다.
6. 비용 추정 섹션은 실험 실행에 대한 추정 비용을 제공합니다. 를 사용하면 실험의 대상 계정 수에 따라 처음부터 끝까지 작업이 실행되는 분당 요금이 청구 AWS FIS됩니다.
7. 실험 시작을 선택합니다.

Amazon EBS 스냅샷

Amazon EBS 스냅샷이라는 시점 사본을 만들어 Amazon EBS 볼륨의 데이터를 백업할 수 있습니다. 스냅샷은 증분 백업입니다. 즉, 볼륨에서 가장 최근 스냅샷 이후 변경된 블록만 저장됩니다. 그러면 스냅샷을 만드는 데 필요한 시간이 최소화되며 데이터를 복제하지 않으므로 스토리지 비용이 절약됩니다.

> **Important**
>
> AWS 는 EBS 볼륨에 저장된 데이터를 자동으로 백업하지 않습니다. 데이터 복원력과 재해 복구를 위해 정기적으로 EBS 스냅샷을 생성하거나 Amazon Data Lifecycle Manager를 사용하여 백업 자동화 또는 AWS Backup을 사용하여 자동 스냅샷 생성을 설정하는 것은 사용자의 책임입니다.

스냅샷은 Amazon S3에서 사용자가 직접 액세스할 수 없는 S3 버킷에 저장됩니다. Amazon EC2 콘솔 또는 Amazon EC2 API를 사용하여 스냅샷을 생성하고 관리할 수 있습니다. Amazon S3 콘솔 또는 Amazon S3 API를 사용하여 스냅샷에 액세스할 수 없습니다.

스냅샷 데이터는 리전의 모든 가용 영역에 자동 복제됩니다. 이는 스냅샷 데이터에 대한 고가용성과 내구성을 제공하고 해당 리전의 모든 가용 영역에서 볼륨을 복원할 수 있도록 합니다.

각 스냅샷에는 (스냅샷을 만든 시점의) 데이터를 새 EBS 볼륨에 복원하는 데 필요한 모든 정보가 들어 있습니다. 스냅샷에서 EBS 볼륨을 생성하는 경우, 새 볼륨은 해당 스냅샷을 생성하는 데 사용된 볼륨과 정확히 일치합니다.

자세한 내용은 Amazon EBS Snapshots 제품 페이지를 참조하세요.

스냅샷 이벤트

CloudWatch Events를 통해 EBS 스냅샷의 상태를 추적할 수 있습니다. 자세한 내용은 EBS 스냅샷 이벤트 단원을 참조하십시오.

스냅샷 요금

스냅샷에 대한 요금은 저장된 데이터 양에 따라 결정됩니다. 스냅샷은 증분이므로 스냅샷을 삭제하면 데이터 스토리지 비용이 줄어들지 않을 수 있습니다. 스냅샷에서 독점적으로 참조하는 데이터는 해당 스냅샷이 삭제될 때 제거되지만 다른 스냅샷에서 참조하는 데이터는 보존됩니다. 자세한 내용은 AWS Billing 사용 설명서에서 Amazon Elastic Block Store 볼륨 및 스냅샷을 참조하세요.

내용

- Amazon EBS 스냅샷 작동 방식
- Amazon EBS 스냅샷 수명 주기
- Amazon EBS 빠른 스냅샷 복원
- Amazon EBS 스냅샷 잠금
- Amazon EBS 스냅샷에 대한 퍼블릭 액세스 차단
- Outposts의 Amazon EBS 로컬 스냅샷
- 로컬 영역의 로컬 스냅샷

Amazon EBS 스냅샷 작동 방식

볼륨에서 생성하는 첫 번째 스냅샷은 항상 전체 스냅샷입니다. 스냅샷을 생성할 때 볼륨에 작성된 모든 데이터 블록이 여기에 포함됩니다. 동일한 볼륨의 후속 스냅샷은 증분 스냅샷입니다. 마지막 스냅샷이 생성된 이후 볼륨에 작성된 변경된 데이터 블록과 새 데이터 블록만 여기에 포함됩니다.

전체 스냅샷의 크기는 소스 볼륨의 크기가 아니라 백업되는 데이터의 크기에 따라 결정됩니다. 마찬가지로 전체 스냅샷과 관련된 스토리지 비용은 소스 볼륨의 크기가 아니라 스냅샷의 크기에 따라 결정됩니다. 예를 들어, `50 GiB`의 데이터만 포함하는 `200 GiB` Amazon EBS 볼륨의 첫 번째 스냅샷을 생성합니다. 그 결과 전체 스냅샷 크기가 `50 GiB`이며, `50 GiB` 스냅샷 스토리지에 대한 요금이 청구됩니다.

마찬가지로 증분 스냅샷의 크기와 스토리지 비용은 이전 스냅샷이 생성된 이후 볼륨에 작성된 데이터의 크기에 따라 결정됩니다. 이전 예제에서 계속해서 `20 GiB`의 데이터를 변경하고 `10 GiB`의 데이터를 추가한 후 동일한 `200 GiB` 볼륨의 두 번째 스냅샷을 생성하는 경우 증분 스냅샷의 크기는 `30 GiB`입니다. 그러면 추가 `30 GiB` 스냅샷 스토리지에 대한 요금이 청구됩니다.

스냅샷 요금에 대한 자세한 내용은 Amazon EBS 요금을 참조하세요.

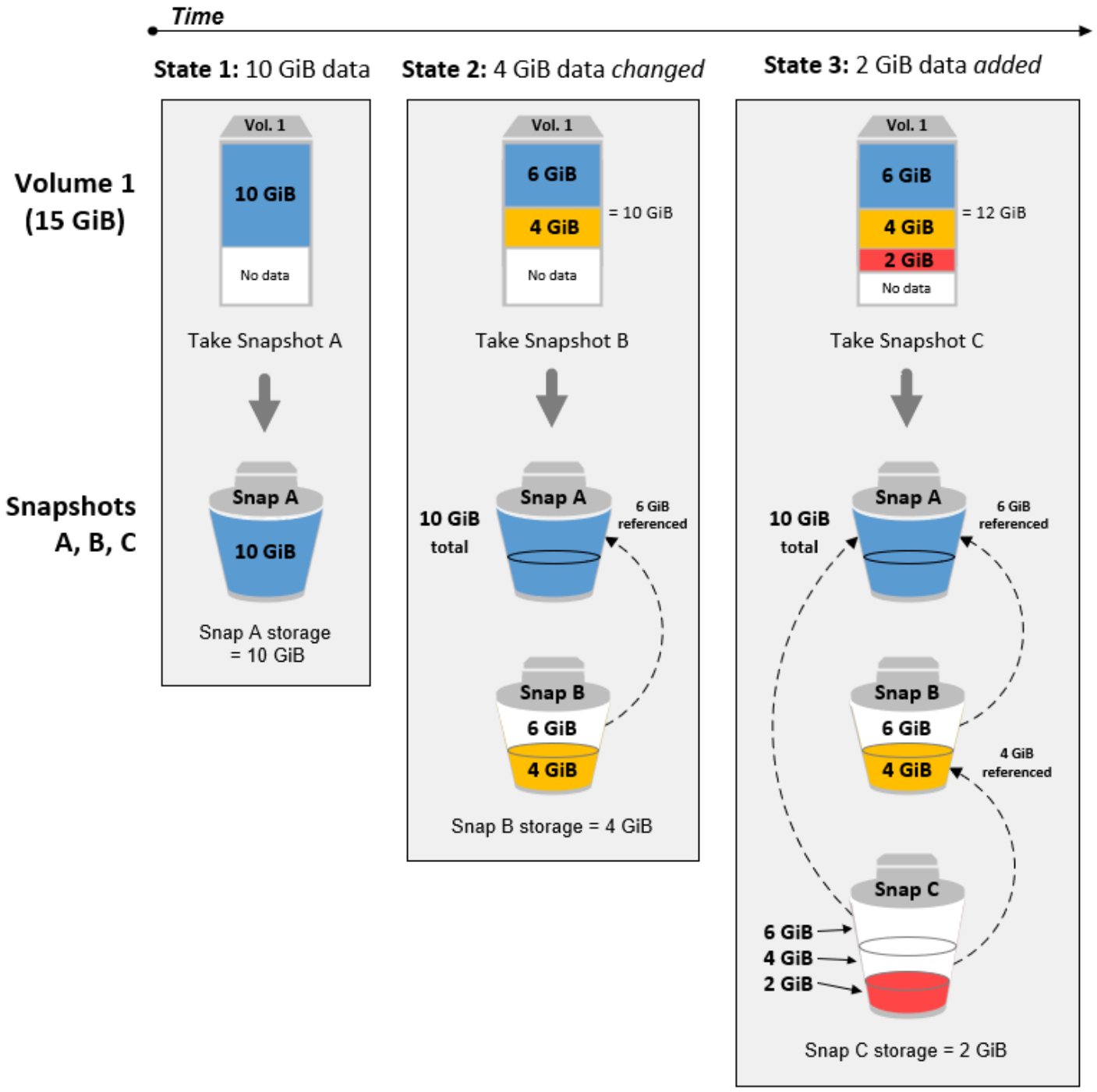
> **Important**
>
> 증분 스냅샷을 아카이빙하면 스냅샷이 생성된 시점에 볼륨에 작성된 모든 블록을 포함하는 전체 스냅샷으로 변환됩니다. 그런 다음 Amazon EBS 스냅샷 아카이브 계층으로 이동됩니다. 아카이브 계층의 스냅샷은 표준 계층의 스냅샷과 다른 요금으로 청구됩니다. 자세한 내용은 Amazon EBS 스냅샷 아카이빙 요금 및 결제 단원을 참조하십시오.

다음 섹션에서는 EBS 스냅샷이 특정 시점의 볼륨 상태를 캡처하는 방법과 변화하는 볼륨의 후속 스냅샷에 이러한 변경 기록이 표시되는 방법을 보여줍니다.

동일한 볼륨의 여러 스냅샷

이 섹션의 다이어그램은 크기가 15 GiB인 볼륨 1을 세 가지 시점에서 보여줍니다. 이 각 세 가지 볼륨 상태에 대한 스냅샷이 만들어집니다. 이 다이어그램은 구체적으로 다음을 보여줍니다.

- 상태 1의 볼륨에는 10 GiB의 데이터가 있습니다. 스냅 A는 볼륨의 첫 번째 스냅샷입니다. 스냅 A는 전체 스냅샷이며 전체 10 GiB 데이터가 백업됩니다.
- 상태 2의 볼륨에는 여전히 10 GiB의 데이터가 포함되어 있지만 스냅 A를 가져온 후에는 4 GiB만 변경되었습니다. 스냅 B는 증분 스냅샷입니다. 변경된 4 GiB만 백업하면 됩니다. 스냅 A에 이미 백업된 나머지 6 GiB의 변경되지 않은 데이터는 다시 백업되지 않고 스냅 B에서 참조됩니다. 이는 파선 모양 화살표로 표시됩니다.
- 스냅 B를 가져온 후 상태 3에서 2 GiB의 데이터가 볼륨에 추가되어 총 12 GiB가 되었습니다. 스냅 C는 증분 스냅샷입니다. 스냅 B를 가져온 후에 추가된 2 GiB만 백업하면 됩니다. 파선 모양 화살표로 표시되었듯이 스냅 C는 스냅 B에 저장된 4 GiB의 데이터 및 스냅 A에 저장된 6 GiB의 데이터를 참조합니다.
- 세 스냅샷에 필요한 총 스토리지는 총 16 GiB입니다. 이는 스냅 A의 경우 10GiB, 스냅 B의 경우 4GiB, 스냅 C의 경우 2GiB를 차지합니다.



서로 다른 볼륨의 증분 스냅샷

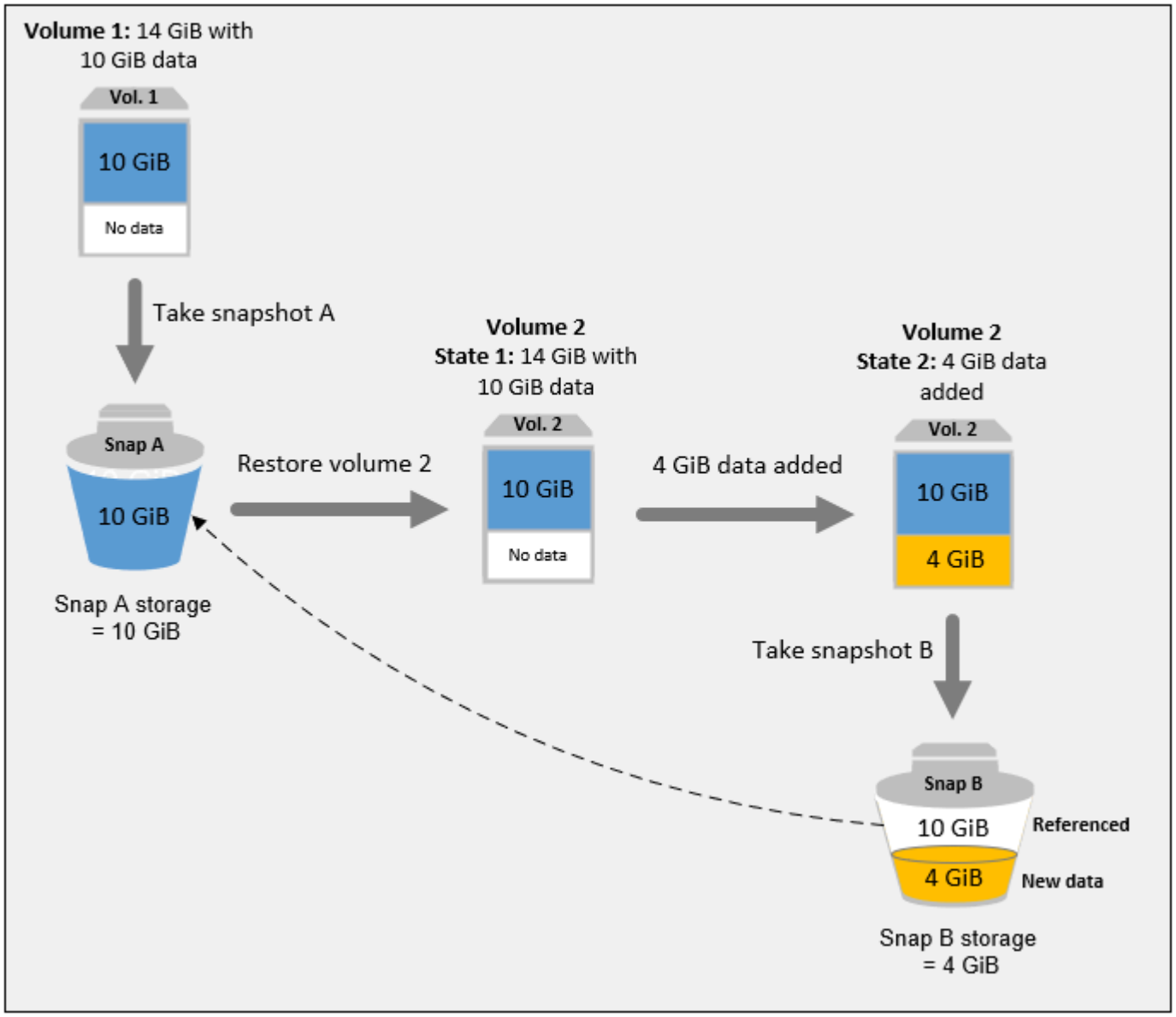
이 섹션의 다이어그램은 여러 볼륨에서 증분 스냅샷을 만드는 방법을 보여줍니다.

1. 크기가 14 GiB인 볼륨 1에는 10 GiB의 데이터가 있습니다. 스냅 A는 이 볼륨의 첫 번째 스냅샷이므로 전체 스냅샷이며 10 GiB의 전체 데이터가 백업됩니다.

2. 볼륨 2는 스냅 A에서 생성되므로 스냅샷이 생성될 때 볼륨 1의 정확한 복제본입니다.
3. 시간이 지나면서 4 GiB의 데이터가 볼륨 2에 추가되고 총 데이터 크기는 14 GiB입니다.
4. 스냅 B는 볼륨 2에서 만들어집니다. 스냅 B의 경우 스냅 A에서 볼륨이 생성된 후 추가된 4 GiB의 데이터만 백업됩니다. 이미 스냅 A에 저장되어 변경되지 않은 다른 10 GiB의 데이터는 다시 백업되는 것이 아니라 스냅 B에서 참조됩니다.

   스냅 B는 다른 볼륨에서 생성되었어도 스냅 A의 증분 스냅샷입니다.

> **Important**
>
> 이 다이어그램에서는 사용자가 볼륨 1과 스냅 A를 소유하고 있으며 볼륨 2가 볼륨 1과 동일한 KMS 키로 암호화되어 있다고 가정합니다. Vol 1을 다른 AWS 계정이 소유하고 해당 계정이 스냅 A를 가져와 사용자와 공유한 경우 스냅 B는 전체 스냅샷이 됩니다. 또는 볼륨 2가 볼륨 1과 다른 KMS 키로 암호화된 경우 스냅 B는 전체 스냅샷이 됩니다.

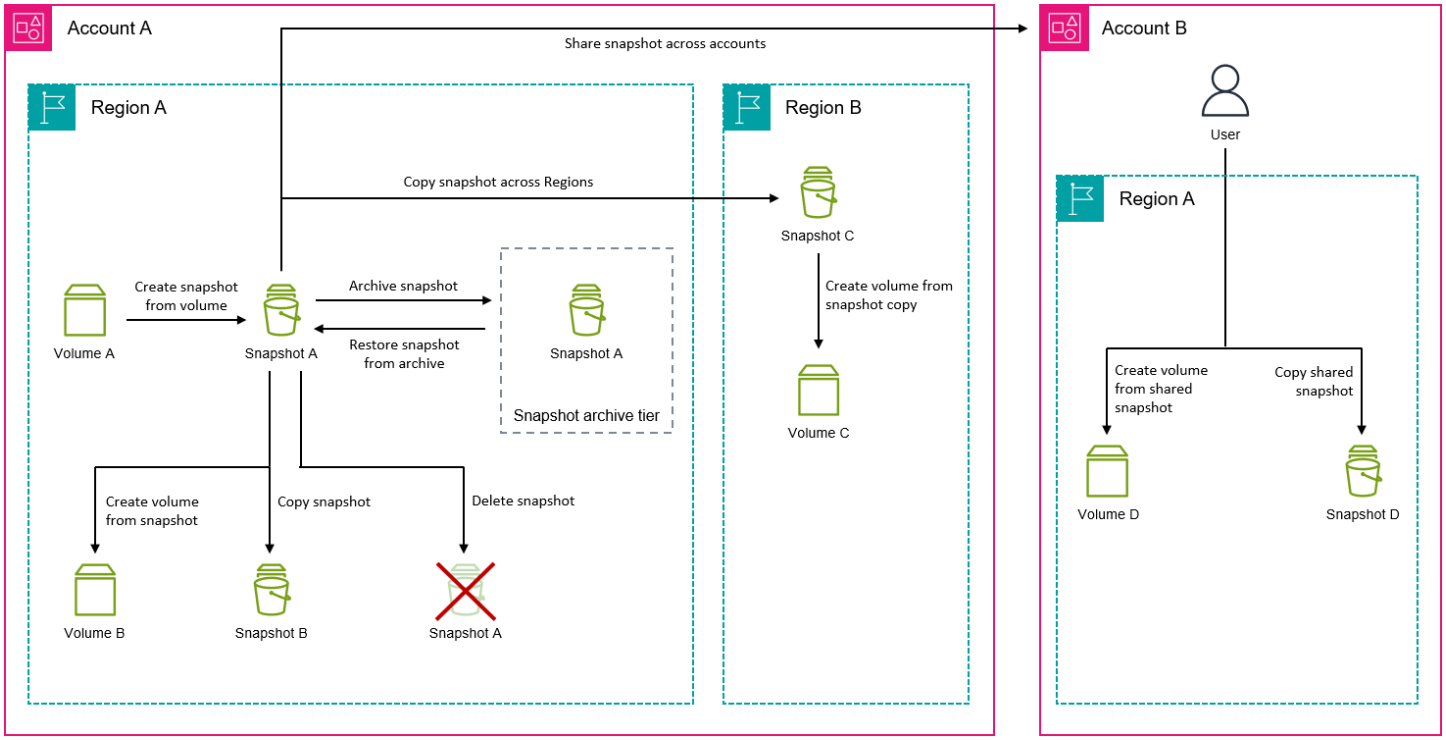


스냅샷을 삭제할 때 데이터가 관리되는 방법에 대한 자세한 내용은 Amazon EBS 스냅샷 삭제 섹션을 참조하세요.

Amazon EBS 스냅샷 수명 주기

Amazon EBS 스냅샷의 수명 주기는 생성 프로세스로 시작합니다. Amazon EBS 볼륨에서 스냅샷을 생성할 수 있습니다. 스냅샷을 사용하여 새 Amazon EBS 볼륨을 복원할 수 있습니다. 동일한 리전 또는 상이한 리전에서 스냅샷 복사본을 생성할 수 있습니다. 스냅샷은 공개적으로 AWS 계정또는 비공개로 다른와 공유할 수 있습니다. 해당 계정에서는 공유 스냅샷에서 볼륨을 복원하거나 자체 계정의 공유 스냅샷 복사본을 생성할 수 있습니다. 스냅샷에 즉시 액세스하지 않아도 된다면 보관하여 스토리지 비용을 절약할 수 있습니다.

다음 이미지는 스냅샷 수명 주기의 일부로 스냅샷에서 수행할 수 있는 작업을 보여줍니다.



작업

- Amazon EBS 스냅샷 생성
- Amazon EBS 스냅샷 정보 보기
- Amazon EBS 스냅샷 복사
- Amazon EBS 스냅샷을 다른 AWS 계정과 공유
- Amazon EBS 스냅샷 아카이브
- Amazon EBS 스냅샷 삭제

Amazon EBS 스냅샷 생성

Amazon EBS 볼륨의 Amazon EBS 스냅샷을 생성하여 해당 볼륨의 특정 시점 백업을 생성할 수 있습니다. 개별 Amazon EBS 볼륨의 스냅샷을 생성하거나 Amazon EC2 인스턴스에 연결된 볼륨의 전체 또는 하위 집합의 다중 볼륨 스냅샷을 생성할 수 있습니다.

스냅샷 생성은 비동기식입니다. 스냅샷은 즉시 생성되지만 모든 데이터가 Amazon S3로 전송될 때까지 `pending` 상태를 유지합니다. 볼륨의 수정된 블록 수에 따라 완료하는 데 몇 시간이 걸릴 수 있습니다. 이 시간 동안 스냅샷에 영향을 주지 않고 볼륨을 계속 사용할 수 있습니다. 스냅샷에는 스냅샷이 요

청된 시점을 기준으로 볼륨에 기록되어 있는 데이터만 포함됩니다. 애플리케이션 또는 운영 체제에 의해 캐시된 데이터는 포함되지 않습니다.

> **Tip**
>
> 스냅샷이 일관되고 완전한지 확인하려면 스냅샷을 생성하기 전에 볼륨에 대한 쓰기를 일시 중지하는 것이 좋습니다. 볼륨에 대한 쓰기를 일시 중지할 수 없는 경우 스냅샷을 생성하기 전에 인스턴스 내에서 볼륨을 탑재 해제하는 것이 좋습니다. 스냅샷이 `pending` 상태로 전환되면 볼륨을 다시 탑재하고 쓰기를 재개할 수 있습니다.
>
> Amazon EC2 인스턴스의 루트 디바이스 역할을 하는 볼륨의 스냅샷을 생성하는 경우에는 스냅샷을 생성하기 전에 인스턴스를 중지하는 것이 좋습니다.

내용

- 스냅샷 암호화
- 스냅샷 대상
- 스냅샷 자동화
- 스냅샷 생성 시 고려 사항
- EBS 볼륨의 스냅샷 생성
- EC2 인스턴스에서 다중 볼륨 EBS 스냅샷 생성

스냅샷 암호화

스냅샷은 소스 볼륨과 동일한 암호화 상태를 자동으로 가져옵니다. 암호화되지 않은 볼륨에서 생성된 스냅샷은 암호화되지 않습니다. 암호화된 볼륨에서 생성된 스냅샷은 볼륨과 동일한 KMS 키를 사용하여 자동으로 암호화됩니다.

> **Tip**
>
> 암호화되지 않은 볼륨에서 암호화된 스냅샷을 생성해야 하는 경우 먼저 볼륨의 암호화되지 않은 스냅샷을 생성한 다음 해당 스냅샷의 암호화된 사본을 생성합니다.

스냅샷 대상

소스 리소스(볼륨 또는 인스턴스)의 위치에 따라 스냅샷을 생성할 수 있는 위치가 결정됩니다.

- 소스 리소스가 리전에 있는 경우, 소스 리소스와 동일한 리전에 스냅샷을 생성해야 합니다.
- 소스 리소스가 로컬 영역에 있는 경우 동일한 로컬 영역 또는 상위 리전에서 스냅샷을 생성할 수 있습니다. 자세한 내용은 로컬 영역의 로컬 스냅샷 단원을 참조하십시오.
- 소스 리소스가 Outpost에 있는 경우 동일한 Outpost 또는 상위 리전에서 스냅샷을 생성할 수 있습니다. 자세한 내용은 Outposts의 Amazon EBS 로컬 스냅샷 단원을 참조하십시오.

스냅샷 자동화

Amazon Data Lifecycle Manager 및 AWS Backup을 사용하여 스냅샷 생성을 자동화할 수 있습니다.

스냅샷 생성 시 고려 사항

- 최대 절전 모드로 전환되었거나 최대 절전 모드가 활성화된 Amazon EC2 인스턴스에 연결된 볼륨의 스냅샷은 생성하지 않는 것이 좋습니다. 자세한 내용은 Amazon EC2 인스턴스 최대 절전 모드 작동 방식을 참조하세요.
- 볼륨의 이전 스냅샷이 `pending` 상태일 때에도 볼륨의 스냅샷을 생성할 수는 있지만 동일한 볼륨에 대해 `pending` 상태의 스냅샷을 여러 개 생성하면 스냅샷이 완료될 때까지 볼륨 성능이 저하될 수 있습니다.
- `pending` 상태로 유지할 수 있는 스냅샷 수와 볼륨 유형당 요청할 수 있는 동시 스냅샷 수에는 제한이 있습니다. 자세한 내용은 Amazon EBS 할당량을 참조하세요. 이러한 할당량 중 하나를 초과하는 경우 현재 스냅샷이 완료될 때까지 기다린 다음 다시 시도하세요.

EBS 볼륨의 스냅샷 생성

단일 볼륨의 단일 스냅샷을 생성할 수 있습니다. 또는 스냅샷 생성을 자동화하려면 Amazon Data Lifecycle Manager 또는AWS Backup를 사용합니다. EC2 인스턴스에 대한 모든 볼륨의 스냅샷을 생성하려면 다중 볼륨 스냅샷을 사용합니다.

Console

스냅샷 생성

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 스냅샷(Snapshots), 스냅샷 생성(Create snapshot)을 선택합니다.
3. [리소스 유형(Resource type)]에서 [볼륨(Volume)]을 선택합니다.
4. 볼륨 ID(Volume ID)에서 스냅샷을 생성할 볼륨을 선택합니다. 암호화 필드에는 볼륨 및 결과 스냅샷의 암호화 상태가 표시됩니다. 이를 수정할 수는 없습니다.

5. (선택 사항) 설명에 스냅샷에 대한 간략한 설명을 입력합니다.
6. 볼륨이 Outpost 또는 로컬 영역에 있는 경우 스냅샷 대상 필드가 나타납니다. 다음 중 하나를 수행하세요.
   - 볼륨이 로컬 영역에 있는 경우, 로컬 영역을 선택하여 동일한 로컬 영역에 스냅샷을 생성하거나 AWS 리전을 선택하여 로컬 영역의 상위 리전에 스냅샷을 생성합니다.
   - 볼륨이 Outpost에 있는 경우, AWS Outpost를 선택하여 동일한 Outpost에 스냅샷을 생성하거나 AWS 리전을 선택하여 Outpost의 상위 리전에 스냅샷을 생성합니다.

   > **Note**
   >
   > 볼륨이 리전에 있는 경우 스냅샷 대상이 표시되지 않습니다. 스냅샷은 볼륨과 동일한 리전에 자동 생성됩니다.
7. (선택 사항) 스냅샷에 사용자 지정 태그를 할당하려면 태그 섹션에서 태그 추가를 선택한 다음 키 값 페어를 입력합니다. 최대 50개의 태그를 추가할 수 있습니다.
8. 스냅샷 생성(Create snapshot)을 선택합니다.

AWS CLI

스냅샷 생성

create-snapshot 명령을 사용합니다.

```
aws ec2 create-snapshot  \
    --volume-id vol-01234567890abcdef \
    --description "Snapshot of the root volume for i-1234567890abcdef0"
```

PowerShell

스냅샷 생성

New-EC2Snapshot cmdlet을 사용합니다.

```
New-EC2Snapshot `
    -VolumeId vol-01234567890abcdef `
    -Description "Snapshot of the root volume for i-1234567890abcdef0"
```

EC2 인스턴스에서 다중 볼륨 EBS 스냅샷 생성

기본적으로 Amazon EC2 인스턴스에서 다중 볼륨 스냅샷을 생성하면 Amazon EBS는 인스턴스에 연결된 모든 Amazon EBS 볼륨의 스냅샷을 생성합니다. 그러나 필요한 경우 루트 볼륨 또는 특정 데이터 볼륨을 제외하도록 선택할 수 있습니다.

> **Tip**
>
> 다중 볼륨 스냅샷을 쉽게 식별 및 관리할 수 있도록 태그를 지정하는 것이 좋습니다. 소스 볼륨에서 해당 스냅샷으로 태그를 복사하여 액세스 정책, 연결 정보, 비용 할당과 같은 스냅샷 메타데이터를 소스 볼륨과 일치하도록 설정할 수도 있습니다.

다중 볼륨 스냅샷 고려 사항

- 모든 스냅샷이 성공적으로 완료되면 결과가 인 `createSnapshots` CloudWatch 이벤트`succeeded`가 AWS 계정으로 전송됩니다. 다중 볼륨 스냅샷 세트에서 스냅샷 하나가 실패하면 다른 모든 스냅샷은 `error` 상태로 전환되고 `failed`의 결과가 있는 `createSnapshots` CloudWatch 이벤트가 계정으로 전송됩니다. 자세한 내용은 스냅샷 생성(createSnapshots) 단원을 참조하십시오.
- 다중 볼륨 스냅샷은 단일 인스턴스에 연결되는 Amazon EBS 볼륨을 루트 볼륨 및 최대 127개의 데이터 볼륨을 포함하여 최대 128개까지 지원합니다.
- 다중 볼륨 스냅샷 세트의 각 스냅샷은 개별 스냅샷으로, 동일한 방식으로 사용할 수 있으며 개별 스냅샷과 동일한 기능을 지원합니다.
- AWS Systems Manager 명령 문서를 사용하여 Amazon EC2 Windows 인스턴스에 연결된 모든 Amazon EBS 볼륨에 대해 애플리케이션에 일관되게 적용되는 스냅샷을 생성할 수 있습니다.

Console

다중 볼륨 스냅샷을 생성하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 스냅샷(Snapshots), 스냅샷 생성(Create snapshot)을 선택합니다.
3. 리소스 유형(Resource type)에서 인스턴스(Instance)를 선택합니다.
4. 설명(Description)에 스냅샷에 대한 간략한 설명을 입력합니다. 이 설명은 모든 스냅샷에 적용됩니다.
5. 인스턴스가 Outpost 또는 로컬 영역에 있는 경우 스냅샷 대상 필드가 나타납니다. 다음 중 하나를 수행하세요.

- 인스턴스가 로컬 영역에 있는 경우, 로컬 영역을 선택하여 동일한 로컬 영역에 스냅샷을 생성하거나 AWS 리전을 선택하여 로컬 영역의 상위 리전에 스냅샷을 생성합니다.
- 인스턴스가 Outpost에 있는 경우 AWS Outpost를 선택하여 동일한 Outpost에 스냅샷을 생성하거나 AWS 리전을 선택하여 Outpost의 상위 리전에 스냅샷을 생성합니다.

> **Note**
>
> 인스턴스가 리전에 있는 경우 스냅샷 대상이 표시되지 않습니다. 스냅샷은 인스턴스와 동일한 리전에 자동 생성됩니다.

6. (선택 사항) 인스턴스의 루트 볼륨을 제외하려면 루트 볼륨 제외를 선택합니다.
7. (선택 사항) 데이터 볼륨을 제외하려면 특정 데이터 볼륨 제외를 선택합니다. Attached data volumes(연결된 데이터 볼륨) 섹션에는 선택한 인스턴스에 현재 연결된 모든 데이터 볼륨이 나열됩니다.

   제외할 데이터 볼륨을 선택합니다. 선택되지 않은 상태로 남아 있는 볼륨만 다중 볼륨 스냅샷 세트에 포함됩니다.
8. (선택 사항) 소스 볼륨에서 해당 스냅샷으로 태그를 자동으로 복사하려면 소스 볼륨에서 태그 복사에서 태그 복사를 선택합니다.
9. (선택 사항) 스냅샷에 추가 사용자 지정 태그를 할당하려면 태그 섹션에서 태그 추가를 선택한 다음 키 값 페어를 입력합니다. 최대 50개의 태그를 추가할 수 있습니다.
10. 스냅샷 생성(Create snapshot)을 선택합니다.

AWS CLI

다중 볼륨 스냅샷을 생성하려면

create-snapshots 명령을 사용합니다.

```
aws ec2 create-snapshots \
    --instance-specification InstanceId=i-1234567890abcdef0 \
    --description "from a multi-volume snapshot of i-1234567890abcdef0"
```

루트 볼륨을 제외하려면 `--instance-specification` 옵션에 다음을 추가합니다.

```
ExcludeBootVolume=true
```

데이터 볼륨을 제외하려면 `--instance-specification` 옵션에 다음을 추가합니다.

```
ExcludeDataVolumeIds=vol-01234567890abcdef
```

PowerShell

다중 볼륨 스냅샷을 생성하려면

New-EC2SnapshotBatch cmdlet을 사용합니다.

```
New-EC2SnapshotBatch `
    -InstanceSpecification_InstanceId i-1234567890abcdef0 `
    -Description "from a multi-volume snapshot of i-1234567890abcdef0"
```

루트 볼륨을 제외하려면 다음 파라미터를 추가합니다.

```
-InstanceSpecification_ExcludeBootVolume $true
```

데이터 볼륨을 제외하려면 다음 파라미터를 추가합니다.

```
-InstanceSpecification_ExcludeDataVolumes "vol-01234567890abcdef"
```

Amazon EBS 스냅샷 정보 보기

스냅샷에 대한 세부 정보를 볼 수 있습니다.

Console

스냅샷 정보를 보려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [스냅샷(Snapshots)]을 선택합니다.
3. 소유한 스냅샷만 보려면 화면 왼쪽 위 모서리에서 내 소유(Owned by me)를 선택합니다. 태그 및 다른 스냅샷 속성을 사용하여 스냅샷 목록을 필터링할 수도 있습니다. 필터(Filter) 필드에서 속성 필드를 선택한 다음 속성 값을 선택하거나 입력합니다. 예를 들어 암호화된 스냅샷만 보려면 암호화(Encryption)를 선택한 다음 `true`를 입력합니다.
4. 특정 스냅샷에 대한 자세한 내용을 보려면 목록에서 해당 ID를 선택합니다.

> **Note**
>
> 전체 스냅샷 크기 필드에는 스냅샷의 전체 크기가 바이트 단위로 표시됩니다. 이는 스냅샷의 증분 크기가 아닙니다. 대신 스냅샷이 생성될 때 소스 볼륨에 기록된 모든 블록의 크기를 나타냅니다.
> 볼륨 크기 필드에는 다른 크기가 지정되지 않은 경우 스냅샷에서 생성되는 EBS 볼륨의 크기가 표시됩니다.

AWS CLI

스냅샷 정보를 보려면

describe-snapshots 명령을 사용합니다.

Example예 1: 태그를 기준으로 필터링

다음 예제는 Stack=production 태그를 가진 스냅샷을 설명합니다.

```
aws ec2 describe-snapshots --filters Name=tag:Stack,Values=production
```

Example예 2: 볼륨을 기준으로 필터링

다음 예제는 지정된 볼륨에서 생성된 스냅샷을 설명합니다.

```
aws ec2 describe-snapshots --filters Name=volume-id,Values=vol-049df61146c4d7901
```

Example예 3: 스냅샷 경과 시간을 기준으로 필터링

JMESPath를 사용하면 표현식으로 결과를 필터링할 수 있습니다. 예를 들어, 다음 명령은 지정된 날짜 이전에 계정에 의해 생성된 모든 스냅샷의 ID를 표시합니다. 소유자를 지정하지 않으면 모든 퍼블릭 스냅샷이 결과에 포함됩니다.

```
aws ec2 describe-snapshots \
    --filters Name=owner-id,Values=123456789012 \
    --query "Snapshots[?(StartTime<='2024-03-31')].[SnapshotId]" \
    --output text
```

다음 명령은 지정된 날짜 범위에 생성된 모든 스냅샷의 ID를 표시합니다.

```
aws ec2 describe-snapshots \
    --filters Name=owner-id,Values=123456789012 \
    --query "Snapshots[?(StartTime>='2024-01-01') && (StartTime<='2024-12-31')].
[SnapshotId]" \
    --output text
```

PowerShell

스냅샷 정보를 보려면

Get-EC2Snapshot cmdlet을 사용합니다.

예제 1: 스냅샷 설명

다음 예제에서는 지정된 스냅샷을 설명합니다.

```
Get-EC2Snapshot -SnapshotId snap-0abcdef1234567890
```

예 2: 볼륨을 기준으로 필터링

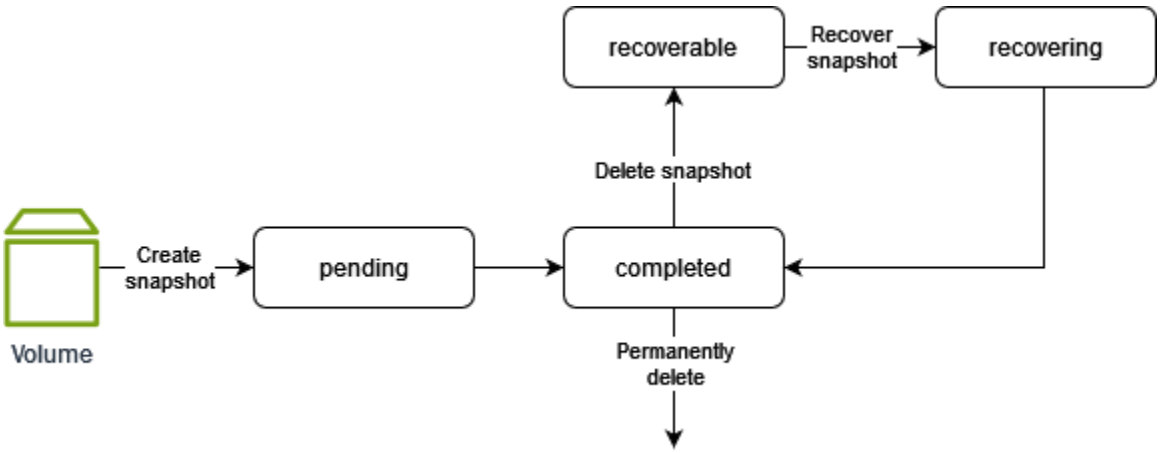
다음 예제는 지정된 볼륨에서 생성된 스냅샷을 설명합니다.

```
Get-EC2Snapshot`
    -Filter @{Name="volume-id"; Values="vol-01234567890abcdef"}
```

스냅샷 상태

Amazon EBS 스냅샷은 생성되는 순간부터 영구적으로 삭제될 때까지 다양한 상태로 전환됩니다.

다음 그림에서는 스냅샷 상태 간 전환을 보여줍니다. 스냅샷은 생성하면 pending 상태로 전환됩니다. 스냅샷이 사용할 준비가 되면 completed 상태로 전환됩니다. 더 이상 스냅샷이 필요하지 않다고 판단되면 스냅샷을 삭제할 수 있습니다. 휴지통 보존 규칙과 일치하는 스냅샷을 삭제하면 휴지통에서 유지되고 recoverable 상태로 전환됩니다. 휴지통에서 스냅샷을 복구하면 recovering 상태와 completed 상태로 차례로 전환됩니다. 그러지 않으면 영구적으로 삭제됩니다.



다음 표에는 스냅샷 상태가 요약되어 있습니다.

Status	설명
pending	스냅샷 생성 프로세스가 아직 진행 중입니다. pending 상태인 동안에는 스냅샷을 사용할 수 없습니다.
completed	스냅샷 생성 프로세스가 완료되었으며, 스냅샷을 사용할 준비가 되었습니다.
recoverable	스냅샷이 현재 휴지통에 있습니다. 스냅샷을 사용하려면 먼저 휴지통에서 복구해야 합니다.
recovering	스냅샷이 휴지통에서 복구되는 중입니다. 스냅샷이 복구되면 completed 상태로 전환되고 사용할 준비가 됩니다.
error	스냅샷 생성 프로세스가 실패했습니다. error 상태라면 스냅샷을 사용할 수 없습니다.

Amazon EBS 스냅샷 복사

스냅샷을 생성한 후 스냅샷이 completed 상태에 도달하면 복사본을 생성할 수 있습니다. 스냅샷 복사본은 원본의 정확한 복사본이지만 고유한 리소스 ID를 가집니다. 소유한 스냅샷과 비공개적으로 또는 공개적으로 공유된 스냅샷을 복사할 수 있습니다. 다음 사용 사례에서 스냅샷을 복사해야 할 수 있습니다.

- 지리적 확장 - 새 리전에서 애플리케이션을 시작해야 하는 경우.
- 마이그레이션 - 가용성을 향상하고 비용을 최소화하기 위해 새 대상으로 애플리케이션을 마이그레이션해야 하는 경우.
- 재해 복구 - 데이터 중복성을 위해 보조 리전에 데이터 및 로그를 백업해야 하는 경우.
- 암호화 - 이전에 암호화되지 않은 스냅샷을 암호화하거나 다른 KMS 키를 사용하여 암호화된 스냅샷을 다시 암호화해야 하는 경우.
- 공유 스냅샷 복사 - 공유된 스냅샷을 복사해야 하는 경우.
- 데이터 보존 및 감사 요구 사항 - 감사 또는 데이터 보존을 위해 데이터를 보존하려면 암호화된 스냅샷을 한 AWS 계정에서 다른 계정으로 복사해야 합니다. 다른 계정을 사용하면 기본 AWS 계정이 손상된 경우 사용자를 보호할 수 있습니다.

다중 볼륨 스냅샷을 복사하려면 생성 중에 할당한 태그를 사용하여 해당 세트의 일부인 스냅샷을 모두 식별한 다음 개별적으로 복사합니다.

Amazon RDS 스냅샷 복사에 대한 자세한 내용은 Amazon RDS 사용 설명서의 DB 스냅샷 복사를 참조하십시오.

가격 책정

스냅샷 복수에 대한 요금 정보는 Amazon EBS 요금을 참조하세요.

내용

- 스냅샷 복사 시 고려 사항
- 스냅샷 복사본의 대상
- 증분 스냅샷 복사
- Amazon EBS 스냅샷 및 EBS 지원 AMI의 시간 기반 복사본
- 암호화 및 스냅샷 복사
- 스냅샷 복사

스냅샷 복사 시 고려 사항

- AWS Marketplace VM Import/Export 및 Storage Gateway 스냅샷을 복사할 수 있지만 대상 리전에서 스냅샷이 지원되는지 확인해야 합니다.

- 대상당 동시 스냅샷 복사 요청은 20개로 제한됩니다. 이 할당량을 초과하면 `ResourceLimitExceeded` 오류가 발생합니다. 이 오류가 발생하면 새 스냅샷 복사를 요청하기 전에 하나 이상의 복사 요청이 완료될 때까지 기다립니다.
- 사용자 정의 태그는 소스 스냅샷에서 스냅샷 복사본으로 복사되지 않습니다. 복사 작업 도중이나 이후에 사용자 정의 태그를 추가할 수 있습니다.
- 스냅샷 복사 작업을 통해 생성된 스냅샷에는 `vol-ffff` 또는 `vol-ffffffff`와 같은 임의 볼륨 ID가 있습니다. 어떠한 용도로도 사용되지 않는 임의 볼륨 ID입니다.
- 스냅샷 복사 작업에 지정된 리소스 수준 권한은 스냅샷 복사본과 소스 스냅샷에만 적용될 수 있습니다. 예시는 예: 스냅샷 복사를 참조하세요.
- 빠른 스냅샷 복원이 활성화된 스냅샷을 복사하는 경우 스냅샷 복사본은 빠른 스냅샷 복원이 자동으로 활성화되지 않습니다. 스냅샷 복사본에 대해 빠른 스냅샷 복원을 명시적으로 활성화해야 합니다.
- 스냅샷을 복사하고 새 KMS 키로 암호화하면 전체(비증분) 복사본이 생성됩니다. 이로 인해 추가 스토리지 비용이 발생합니다.
- 스냅샷을 새 리전에 복사하면 전체(비증분) 복사본이 생성됩니다. 이로 인해 추가 스토리지 비용이 발생합니다.
- 외부 또는 리전 간 데이터 전송을 사용하는 경우 추가 EC2 데이터 전송 요금이 적용됩니다. 시작 후 스냅샷을 삭제해도 이미 전송된 데이터에 대한 요금은 부과됩니다.

스냅샷 복사본의 대상

소스 스냅샷의 위치가 복사 가능 여부를 결정합니다.

- 소스 스냅샷이 리전에 있는 경우 해당 리전 내, 다른 리전 또는 해당 리전과 연결된 Outpost 또는 해당 리전의 로컬 영역으로 복사할 수 있습니다.
- 소스 스냅샷이 로컬 영역에 있는 경우 해당 로컬 영역 내, 동일한 영역 그룹의 다른 로컬 영역 또는 해당 로컬 영역의 상위 리전으로 복사할 수 있습니다.
- 소스 스냅샷이 Outpost에 있는 경우 복사할 수 없습니다.

증분 스냅샷 복사

동일한 KMS 키를 사용하는 동일한 계정 및 리전 내 스냅샷 복사 작업은 항상 증분 복사입니다. 그러나 다른 KMS 키를 사용하여 스냅샷 복사본을 암호화하는 경우 복사는 전체 복사입니다.

리전 또는 계정 간에 스냅샷을 복사할 때 다음 조건이 충족되면 복사를 증분 복사입니다.

- 이전에 스냅샷이 대상 리전 또는 계정에 복사되었습니다.
- 가장 최근 스냅샷 복사는 여전히 대상 리전 또는 계정에 있습니다.
- 가장 최근의 스냅샷 복사본이 아카이브되지 않았습니다.
- 대상 리전 또는 계정에 있는 스냅샷의 모든 복사는 암호화되지 않거나 동일한 KMS 키를 사용하여 암호화되었습니다.

> **Tip**
>
> 대상 리전 또는 계정에서 볼륨의 가장 최근 스냅샷 복사본을 추적할 수 있도록 스냅샷 복사본에 볼륨 ID와 생성 시간을 표시하는 태그를 지정하는 것이 좋습니다.

스냅샷 복사가 증분인지 확인하려면 copySnapshot CloudWatch 이벤트를 점검하십시오.

Amazon EBS 스냅샷 및 EBS 지원 AMI의 시간 기반 복사본

시간 기반 복사본은 EBS 스냅샷 및 EBS 지원 AMIs 지정된 기간 내에 AWS 리전 내부 및 리전 간에 복사되도록 하여 데이터 복제에 대한 규정 준수 또는 비즈니스 요구 사항을 충족하는 데 도움이 될 수 있습니다. 또한 시간 기반 복사본은 백업 관리자가 엄격한 재해 복구 요구 사항(복구 시점 목표 및 복구 시간 목표)을 충족하는 데 도움이 될 수 있으며 스냅샷 및 EBS 지원 AMI의 예측 가능한 복사 시간을 보장하여 개발 민첩성을 향상하도록 합니다.

시간 기반 스냅샷 및 EBS 지원 AMI 복사 작업에서는 복사가 완료되어야 하는 완료 기간(15분에서 48시간 사이)을 지정합니다. 완료 기간은 15분 단위로 지정해야 합니다.

주제

- 할당량
- 완료 기간 결정
- 고려 사항
- 모니터링
- 요금 및 결제

할당량

다음 할당량은 시간 기반 스냅샷 및 EBS 지원 AMI 복사 작업에 적용됩니다.

할당량	설명	할당량 값	조정 가능
스냅샷 복사 작업 처리량 할당량	단일 시간 기반 스냅샷 복사 작업으로 달성할 수 있는 최대 처리량. Note AMI 복사 작업의 경우, 할당량은 AMI와 연결된 각각의 개별 스냅샷에 적용됩니다.	500MiB/s	아니요
누적 스냅샷 복사 처리량 할당량	소스 및 대상 리전 간의 동시 시간 기반 스냅샷 복사 작업을 통해 달성할 수 있는 최대 누적 처리량. Note AMI 복사 작업의 경우 AMI와 연결된 각각의 개별 스냅샷은 할당량 계산에 포함됩니다.	2,000MiB/s	예

시간 기반 스냅샷 복사 작업을 시작할 때는 완료 기간을 지정합니다. 요청에 사용되는 처리량은 스냅샷 데이터의 크기와 요청된 완료 기간의 의해 결정됩니다. 예를 들어 225,000MiB(0.214TiB)의 데이터가 있는 스냅샷을 복사하고 15분의 완료 기간을 요청하는 경우, 처리량은 250MiB/s(225,000MiB ÷ 15분 = 250MiB/s)입니다.

시간 기반 AMI 복사 작업을 시작하면, 지정한 완료 기간이 AMI와 연결된 각 스냅샷에 적용됩니다. 각 스냅샷의 크기가 다를 수 있으므로 각 스냅샷은 다른 처리량으로 복사되어 모든 스냅샷이 완료 기간 내에 복사되도록 합니다. 예를 들어 다음과 같은 스냅샷이 연결된 AMI가 있다고 가정해 보겠습니다.

- 스냅샷 1: 200,000MiB
- 스냅샷 2: 500,000MiB
- 스냅샷 3: 450,000MiB

이 AMI에 대해 시간 기반 복사를 시작하고 60분의 완료 기간을 지정하면 요청은 다음 처리량을 사용합니다.

- 스냅샷 1: 55.56MiB/s(200,000MiB ÷ 60분 = 55.56MiB/s)
- 스냅샷 2: 138.89MiB/s(500,000MiB ÷ 60분 = 138.89MiB/s)
- 스냅샷 3: 125MiB/s(450,000MiB ÷ 60분 = 125MiB/s)

즉, 요청은 319.45MiB/s의 누적 스냅샷 복사 처리량 할당량을 사용하여 복사가 60분 내에 완료되도록 합니다.

시간 기반 스냅샷 또는 EBS 지원 AMI 복사 요청을 시작하고 사용 가능한 누적 스냅샷 복사 처리량 할당량이 다음과 같은 경우:

- 필요한 처리량 속도보다 크거나 같은 경우, 요청된 완료 기간 내에 복사가 완료됩니다.
- 필요한 처리량 속도보다 작지만 0보다 큰 경우, 요청이 성공하지만 요청한 것보다 오래 걸립니다. 복사는 사용 가능한 처리량 할당량을 사용하여 완료됩니다.
- 0(할당량에 도달)인 경우, 요청이 실패합니다.

완료 기간 결정

시간 기반 스냅샷 또는 EBS 지원 AMI 복사 작업에 대해 요청할 수 있는 최소 완료 기간은 15분이고, 요청할 수 있는 최대 완료 기간은 48시간입니다. 완료 기간은 15분 단위로 지정해야 합니다.

동시 시간 기반 스냅샷 복사 작업

모든 동시 작업의 총 처리량이 누적 스냅샷 복사 처리량 할당량(기본적으로 2,000MiB/s) 내에 있는 한 동일한 소스 및 대상 리전 간에 동시 시간 기반 스냅샷 복사 작업을 수행할 수 있습니다.

기존 스냅샷에 필요한 완료 기간을 달성할 수 있는지 확인하려면 모든 스냅샷의 합산 크기를 필요한 완료 기간으로 나누어 필요한 처리량 비율을 결정합니다.

> **Tip**
>
> 스냅샷의 정확한 데이터 크기를 모르는 경우, 전체 스냅샷 크기를 대신 사용할 수 있습니다. 전체 스냅샷 크기를 가져오려면 describe-snapshots AWS CLI 명령을 사용합니다.

```
required throughput rate = combined snapshot size ÷ required completion duration
```

필요한 처리량 속도가 누적 스냅샷 복사 처리량 할당량보다 작은 경우, 필요한 완료 기간을 달성할 수 있습니다. 필요한 처리량 속도가 누적 스냅샷 복사 처리량 할당량보다 큰 경우, 필요한 처리량 속도보다 10% 이상 높은 할당량 증가를 요청할 것이 권장됩니다.

> **Tip**
>
> Amazon EC2 콘솔은 특정 기간 동안 두 리전 간에 얼마나 많은 스냅샷 데이터를 복사했는지와 특정 누적 스냅샷 복사 처리량 할당량을 기준으로 해당 데이터 양에 대해 달성할 수 있는 최소 완료 기간을 확인하는 데 사용할 수 있는 계산기를 제공합니다. 계산기는 `SnapshotCopyBytesTransferred` CloudWatch 지표를 사용하여 일정 기간 동안 두 리전 간에 복사된 데이터를 계산합니다. 계산기를 열려면 Amazon EC2 콘솔 탐색 패널에서 스냅샷을 선택한 다음 작업, 복사 기간 계산기 시작을 선택합니다.
> 스냅샷 복사 기간 계산기는 AWS Outposts로컬 영역 및 Wavelength 영역에서 지원되지 않습니다.

개별 시간 기반 스냅샷 복사 작업

개별 시간 기반 스냅샷 복사 작업의 최소 완료 기간은 스냅샷 데이터의 크기를 스냅샷 복사 작업 처리량 할당량(500MiB/s)으로 나누어 계산할 수 있습니다.

> **Tip**
>
> 스냅샷의 정확한 데이터 크기를 모르는 경우, 전체 스냅샷 크기를 대신 사용할 수 있습니다. 전체 스냅샷 크기를 가져오려면 describe-snapshots AWS CLI 명령을 사용합니다.

```
minimum completion duration = Max(15 minutes, (snapshot data size ÷ 500 MiB/s)
```

예를 들어 900,000MiB의 데이터에 대한 스냅샷의 최소 완료 기간은 30분입니다.

```
minimum completion duration = Max(15 minutes, (900,000 MiB ÷ 500 MiB/s)
= Max(15 minutes, 30 minutes)
= 30 minutes
```

시간 기반 AMI 복사 작업

연결된 단일 스냅샷이 있는 EBS 지원 AMI에 대해 시간 기반 AMI 복사 작업을 시작하는 경우 개별 시간 기반 스냅샷 복사 작업과 동일한 방식으로 작동하며 동일한 처리량 제한이 적용됩니다.

연결된 여러 스냅샷이 있는 EBS 지원 AMI에 대해 시간 기반 AMI 복사 작업을 시작하는 경우 동시 시간 기반 스냅샷 복사 작업과 동일한 방식으로 작동하며 동일한 처리량 제한이 적용됩니다. 연결된 각 스냅샷은 별도의 스냅샷 복사 요청을 생성하며, 각 요청은 누적 스냅샷 복사 처리량 할당량에 포함됩니다. 사용자가 지정한 완료 기간은 연결된 각 스냅샷에 적용됩니다.

고려 사항

- 동일한 리전 내에서 스냅샷을 복사하거나 리전 간에 스냅샷을 복사할 때 시간 기반 스냅샷 및 EBS 지원 AMI 복사 작업을 시작할 수 있습니다.
- 동일한 스냅샷 또는 AMI에 대해 두 개의 시간 기반 복사 작업을 시작하는 경우, 첫 번째 복사 작업이 완료된 후에만 두 번째 복사 작업의 완료 기간이 시작됩니다.
- 시간 기반 복사 작업 및 스냅샷 복사 기간 계산기는 AWS Outposts로컬 영역 및 Wavelength 영역에서 지원되지 않습니다.

모니터링

시간 기반 스냅샷 및 EBS 지원 AMI 복사 작업의 진행 상황은 Amazon EC2 콘솔 및 AWS CLI를 사용하여 모니터링할 수 있습니다. 콘솔에서 스냅샷을 선택한 다음 세부 정보 탭에서 진행률 필드를 검사하세요. 를 사용하여 describe-snapshots 명령 응답에서 `Progress` 출력 요소를 AWS CLI검사합니다.

콘솔에서 시작 시간과 완료 시간 간의 차이를 확인하거나 `describe-snapshots` 응답에서 `StartTime`과 `CompletionTime` 간의 차이를 확인하여 요청된 완료 기간 내에 시간 기반 스냅샷 또는 EBS 지원 AMI 복사 작업이 완료되었는지 확인할 수 있습니다.

`copySnapshot` Amazon EventBridge 이벤트를 사용하여 시간 기반 복사 작업의 결과를 모니터링할 수도 있습니다. 이 이벤트는 작업이 완료되었는지 여부와 요청된 완료 기간이 충족되었는지 여부를 나타냅니다. 완료 기간이 충족되지 않은 경우 이벤트에는 원인에 대한 추가 정보가 포함됩니다. 자세한 내용은 EBS 스냅샷 이벤트 단원을 참조하십시오.

요금 및 결제

> **Note**
>
> 표준 스냅샷 복사 작업과 마찬가지로, 스냅샷을 새 리전에 복사하는 경우 전체(비증분) 복사본이 생성되고 추가 스토리지 비용이 발생합니다. 동일한 스냅샷의 후속 복사본은 증분입니다. 또한, 외부 또는 리전 간 데이터 전송을 사용하는 경우 추가 Amazon EC2 데이터 전송 요금이 적용됩니다.

시간 기반 스냅샷 및 EBS 지원 AMI 복사 작업에 대해서는 추가 요금이 부과됩니다. 시간 기반 복사 작업에는 복사된 스냅샷 데이터의 GiB당 요청된 완료 기간을 기준으로 하는 요금이 부과됩니다. 고정 요금은 다음과 같습니다.

> **Note**
>
> 완료 기간은 15분 단위로 지정해야 합니다. 최소 완료 시간은 15분이고 최대 완료 시간은 48시간입니다.

- 15분 - 데이터 GiB당 0.020 USD
- 30분 및 45분 - 데이터 GiB당 0.018 USD
- 1시간~1시간 45분 - 데이터 GiB당 0.016 USD
- 2시간~3시간 45분 - 데이터 GiB당 0.014 USD
- 4시간~7시간 45분 - 데이터 GiB당 0.012 USD
- 8시간~15시간 45분 - 데이터 GiB당 0.010 USD
- 16시간 이상 - 데이터 GiB당 0.005 USD

예를 들어 8시간의 완료 기간에 3,000GiB의 데이터에 대한 스냅샷을 복사하는 경우 30 USD(0.010 USD x 3,000GiB)가 청구됩니다.

시간 기반 복사 작업을 시작하지만 할당량 초과로 인해 요청된 완료 기간이 충족되지 않는 경우, 요청된 완료 기간 대신 실제 완료 기간을 기준으로 요금이 청구됩니다. 예를 들어 1시간의 완료 기간을 요청했지만 작업이 2시간 후에 완료되는 경우 2시간의 완료 기간에 대한 요금을 기준으로 요금이 청구됩니다.

Amazon EBS가 요청된 완료 기간을 달성할 수 없거나 서비스 측 문제로 인해 요청이 취소된 경우에는 시간 기반 스냅샷 복사 작업에 대한 추가 요금이 청구되지 않습니다.

시간 기반 스냅샷 복사 작업이 진행 중일 때 스냅샷 복사본을 삭제하면 지정된 완료 기간에 해당하는 요율로 해당 시점까지 복사된 데이터에 대한 요금이 청구됩니다.

암호화 및 스냅샷 복사

> **Note**
>
> Amazon S3 서버 측 암호화(256비트 AES)는 복사 작업 중에 전송되는 스냅샷 데이터를 보호합니다.

암호화되지 않은 소스 스냅샷의 암호화된 스냅샷 복사본을 생성할 수 있습니다. 또한 소스 스냅샷과 다른 KMS 키로 스냅샷 복사본을 암호화할 수 있습니다. 하지만 복사 작업 중 스냅샷 복사본의 암호화 상태를 변경하면 증분식이 아닌 전체 복사본이 생성되어 많은 양의 데이터가 전송되고 스토리지 요금이 많이 발생할 수 있습니다.

> **Tip**
>
> 자신에게 공유된 암호화된 스냅샷을 사용할 때는 자체 KMS 키로 스냅샷을 복사하여 다시 암호화하는 것이 좋습니다. 이렇게 하면 원본 KMS 키가 손상되거나 소유자가 액세스 권한을 취소하는 바람에 스냅샷 및 자신이 해당 스냅샵으로 생성한 암호화된 볼륨에 액세스하지 못하게 되는 상황을 피할 수 있습니다.

암호화된 스냅샷을 복사하는 데 필요한 권한

암호화된 스냅샷을 복사하려면 사용자에게 Amazon EBS 암호화 사용 권한이 있어야 합니다.

- • `kms:DescribeKey`
  - `kms:CreateGrant`
  - `kms:GenerateDataKey`
  - `kms:GenerateDataKeyWithoutPlaintext`
  - `kms:ReEncrypt`
  - `kms:Decrypt`

- 다른 AWS 계정에서 공유된 암호화된 스냅샷을 복사하려면 해당 스냅샷을 암호화하는 데 사용된 고객 관리형 키를 사용할 권한이 있어야 합니다. 자세한 내용은 공유 Amazon EBS 스냅샷을 암호화하는 데 사용되는 KMS 키 공유 단원을 참조하십시오.

스냅샷 복사본의 암호화 결과

다음 표에서는 사용자가 소유한 스냅샷과 사용자에게 공유된 스냅샷을 복사할 때의 암호화 결과를 설명합니다.

대상 리전에 대한 기본적으로 암호화	소스 스냅샷	스냅샷 복사본 암호화 결과	참고
비활성화됨	암호화되지 않음	선택적 암호화	복사본을 암호화하는 경우 사용할 KMS 키를 지정할 수 있습니다. 복사본을 암호화하지만 KMS 키를 지정하지 않으면 기본적으로 암호화에 지정된 키가 사용됩니다.
비활성화됨	암호화됨	자동 암호화	사용할 KMS 키를 지정할 수 있습니다. KMS 키를 지정하지 않으면 AWS 관리형 키 (aws/ebs)가 사용됩니다.
활성화됨	암호화되지 않음	자동 암호화	사용할 KMS 키를 지정할 수 있습니다. KMS 키를 지정하지 않으면 기본적으로 암호화에 지정된 키가 사용됩니다.
활성화됨	암호화됨	자동 암호화	사용할 KMS 키를 지정할 수 있습니다. KMS 키를 지정하지 않으면 기본적으로 암호화에 지정된 키가 사용됩니다.

스냅샷 복사

한 리전에서 다른 리전으로 스냅샷을 복사할 수 있습니다. 비암호화된 스냅샷을 암호화된 스냅샷으로 복사할 수 있습니다. 그러나, 암호화 키를 사용할 권한 없이 암호화된 스냅샷을 복사하려고 하면 작업이 자동으로 실패하고 스냅샷 복사본이 '지정된 키 ID에 액세스할 수 없음' 상태 메시지를 수신합니다.

Console

스냅샷 복사

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [스냅샷(Snapshots)]을 선택합니다.
3. 복사할 스냅샷을 선택한 다음 작업(Actions), 스냅샷 복사(Copy snapshot)를 선택합니다.
4. 설명(Description)에 스냅샷 사본에 대한 간략한 설명을 입력합니다.

   기본적으로 설명에는 소스 스냅샷에 대한 정보가 포함되어 사용자는 원본과 사본을 구분할 수 있습니다.
5. 스냅샷 복사본의 대상을 지정합니다.

   - 스냅샷을 동일한 리전 또는 다른 리전에 복사하려면 AWS 리전을 선택한 다음 대상 리전을 선택합니다.
   - 스냅샷을 로컬 영역에 복사하려면 AWS 로컬 영역을 선택한 다음 대상 로컬 영역을 선택합니다.
   - (Outpost 고객에만 해당) 스냅샷을 Outpost로 복사하려면 AWS Outpost를 선택한 다음 대상 Outpost의 ARN을 입력합니다.
6. 특정 기간에 스냅샷 복사를 완료해야 하는 경우 시간 기반 복사 활성화를 선택하세요. 완료 기간에 필요한 완료 기간(15분 증분 단위)을 입력하세요. 자세한 설명은 Amazon EBS 스냅샷 및 EBS 지원 AMI의 시간 기반 복사본 섹션을 참조하세요.

   특정 기간에 스냅샷 복사를 완료할 필요가 없는 경우 시간 기반 복사를 활성화하지 마세요. 이 경우 최선의 작업으로 스냅샷 복사가 완료됩니다.
7. (Outpost 고객에만 해당) 선택한 리전의 Outpost에 스냅샷 복사본을 생성하려면 스냅샷 대상에서 AWS Outpost를 선택한 다음 대상 Outpost ARN에 스냅샷을 복사할 Outpost의 ARN을 입력합니다. 스냅샷 대상 필드는 선택한 리전에 Outpost가 있는 경우에만 표시됩니다.
8. 스냅샷 사본에 대한 암호화 상태를 지정합니다.

   소스 스냅샷이 암호화되었거나 계정에 기본적으로 암호화가 활성화되어 있으면 스냅샷 복사본이 자동으로 암호화됩니다. 소스 스냅샷이 암호화되지 않았으며 계정이 기본적으로 암호화를 사용하도록 설정되어 있지 않은 경우 암호화는 선택 사항입니다.
9. 스냅샷 복사를 선택합니다.

AWS CLI

다른 리전에 스냅샷을 복사하려면

copy-snapshot 명령을 사용합니다. 다음 예제는 소스 리전에서 --region 옵션에 지정된 현재 리전으로 지정된 스냅샷을 복사합니다.

```
aws ec2 copy-snapshot \
    --source-snapshot-id snap-0abcdef1234567890 \
    --source-region us-east-1 \
    --region us-west-2
```

암호화되지 않은 스냅샷을 암호화된 스냅샷으로 복사하려면

copy-snapshot 명령을 사용합니다. 다음 예제는 지정된 암호화되지 않은 스냅샷을 소스 리전에서 현재 리전으로 복사하고 지정된 KMS 키를 사용하여 새 스냅샷을 암호화합니다.

```
aws ec2 copy-snapshot \
    --source-snapshot-id snap-0abcdef1234567890 \
    --source-region us-east-1 \
    --encrypted \
    --kms-key-id alias/my-kms-key
```

PowerShell

다른 리전으로 스냅샷을 복사하려면

Copy-EC2Snapshot cmdlet을 사용합니다. 다음 예제는 소스 리전에서 --region 옵션에 지정된 현재 리전으로 지정된 스냅샷을 복사합니다.

```
Copy-EC2Snapshot `
   -SourceSnapshotId snap-0abcdef1234567890 `
   -SourceRegion us-east-1 `
   -Region us-west-2
```

암호화되지 않은 스냅샷을 암호화된 스냅샷으로 복사하려면

Copy-EC2Snapshot cmdlet을 사용합니다. 다음 예제는 지정된 암호화되지 않은 스냅샷을 소스 리전에서 현재 리전으로 복사하고 지정된 KMS 키를 사용하여 새 스냅샷을 암호화합니다.

```
Copy-EC2Snapshot `
```

```
    -SourceSnapshotId snap-0abcdef1234567890 `
    -SourceRegion us-east-1 `
    -Encrypted $true `
    -KmsKeyId alias/my-kms-key
```

Amazon EBS 스냅샷을 다른 AWS 계정과 공유

다른 AWS 계정과 스냅샷을 공유하려는 경우 스냅샷 권한을 수정할 수 있습니다. 스냅샷을 다른 모든 AWS 계정과 공개적으로 공유하거나 지정한 개별 AWS 계정과 비공개로 공유할 수 있습니다. 권한을 부여받은 사용자는 공유 스냅샷을 사용하여 자체 EBS 볼륨을 생성할 수 있습니다. 원본 스냅샷은 영향을 받지 않습니다.

> **Important**
>
> 스냅샷을 공유하면 다른 사람들이 해당 스냅샷의 모든 데이터에 액세스할 수 있게 됩니다. 모든 스냅샷 데이터를 신뢰할 수 있는 사용자하고만 공유하세요.

스냅샷의 퍼블릭 공유를 방지하려면 Amazon EBS 스냅샷에 대한 퍼블릭 액세스 차단을 활성화합니다.

주제

- 스냅샷을 공유하기 전에
- 스냅샷 공유
- 공유 Amazon EBS 스냅샷을 암호화하는 데 사용되는 KMS 키 공유
- 자신에게 공유된 Amazon EBS 스냅샷 사용
- 공유한 스냅샷의 사용 확인

스냅샷을 공유하기 전에

다음은 스냅샷을 공유할 때 고려할 사항입니다.

- 리전에 스냅샷에 대한 퍼블릭 액세스 차단이 활성화된 경우 스냅샷을 공개적으로 공유하려는 시도가 차단됩니다. 스냅샷은 여전히 비공개로 공유할 수 있습니다.
- 스냅샷은 생성된 리전으로 제한됩니다. 스냅샷을 다른 리전과 공유하려면 스냅샷을 해당 리전에 복사한 다음 복사본을 공유합니다. 자세한 내용은 Amazon EBS 스냅샷 복사 단원을 참조하십시오.

- 기본 AWS 관리형 키로 암호화된 스냅샷은 공유할 수 없습니다. 고객 관리형 키로 암호화된 스냅샷만 공유할 수 있습니다. 자세한 내용은 AWS Key Management Service 개발자 안내서에서 키 생성을 참조하세요.
- 암호화되지 않은 스냅샷만 공개적으로 공유할 수 있습니다.
- 암호화된 스냅샷을 공유할 때는 해당 스냅샷을 암호화하는 데 사용된 고객 관리형 키도 공유해야 합니다. 자세한 내용은 공유 Amazon EBS 스냅샷을 암호화하는 데 사용되는 KMS 키 공유 섹션을 참조하세요.

스냅샷 공유

스냅샷을 공개적으로 또는 특정 AWS 계정과 공유할 수 있습니다.

Console

스냅샷을 공유하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [스냅샷(Snapshots)]을 선택합니다.
3. 공유할 스냅샷을 선택하고 작업(Actions), 권한 수정(Modify permissions)을 선택합니다.
4. 스냅샷의 권한을 지정합니다. 현재 설정(Current setting)은 스냅샷의 현재 공유 권한을 나타냅니다.
   - 스냅샷을 모든 AWS 계정과 공개적으로 공유하려면 퍼블릭을 선택합니다.
   - 스냅샷을 특정 AWS 계정과 비공개로 공유하려면 비공개를 선택합니다. 그런 다음 계정 공유(Sharing accounts) 섹션에서 계정 추가(Add account)를 선택하고 공유할 계정의 12자리 계정 ID(하이픈 제외)를 입력합니다.
5. 변경 사항 저장을 선택합니다.

AWS CLI

스냅샷에 대한 권한은 스냅샷의 `createVolumePermission` 속성을 사용하여 지정됩니다. 스냅샷을 퍼블릭으로 설정하려면 그룹을 `all`로 설정합니다. 스냅샷을 특정 AWS 계정과 공유하려면 사용자를 AWS 계정의 ID로 설정합니다.

스냅샷을 공개적으로 공유하려면

modify-snapshot-attribute 명령을 사용합니다.

--attribute에 createVolumePermission을 지정합니다. --operation-type에 add을 지정합니다. --group-names에 all을 지정합니다.

```
aws ec2 modify-snapshot-attribute \
    --snapshot-id snap-0abcdef1234567890 \
    --attribute createVolumePermission \
    --operation-type add \
    --group-names all
```

스냅샷을 비공개로 공유하려면

modify-snapshot-attribute 명령을 사용합니다.

--attribute에 createVolumePermission을 지정합니다. --operation-type에 add을 지정합니다. 에서 스냅샷을 공유할 AWS 계정의 12자리 IDs를 --user-ids지정합니다.

```
aws ec2 modify-snapshot-attribute \
    --snapshot-id snap-0abcdef1234567890 \
    --attribute createVolumePermission \
    --operation-type add \
    --user-ids 123456789012 111122223333
```

PowerShell

스냅샷에 대한 권한은 스냅샷의 createVolumePermission 속성을 사용하여 지정됩니다. 스냅샷을 퍼블릭으로 설정하려면 그룹을 all로 설정합니다. 스냅샷을 특정 AWS 계정과 공유하려면 사용자를 AWS 계정의 ID로 설정합니다.

스냅샷을 공개적으로 공유하려면

Edit-EC2SnapshotAttribute cmdlet을 사용합니다.

-Attribute에 CreateVolumePermission을 지정합니다. -OperationType에 Add을 지정합니다. -GroupName에 all을 지정합니다.

```
Edit-EC2SnapshotAttribute `
    -SnapshotId snap-0abcdef1234567890 `
    -Attribute CreateVolumePermission `
    -OperationType Add `
    -GroupName all
```

스냅샷을 비공개로 공유하려면

Edit-EC2SnapshotAttribute cmdlet을 사용합니다.

-Attribute에 CreateVolumePermission을 지정합니다. -OperationType에 Add을 지정합니다. 에서 스냅샷을 공유할 AWS 계정의 12자리 IDs를 UserId지정합니다.

```
Edit-EC2SnapshotAttribute `
    -SnapshotId snap-0abcdef1234567890 `
    -Attribute CreateVolumePermission `
    -OperationType Add `
    -UserId 123456789012 111122223333
```

공유 Amazon EBS 스냅샷을 암호화하는 데 사용되는 KMS 키 공유

암호화된 스냅샷을 공유할 때는 해당 스냅샷을 암호화하는 데 사용된 고객 관리형 키도 공유해야 합니다. 생성 당시에 또는 나중에 고객 관리형 키에 교차 계정 권한을 적용할 수 있습니다.

암호화된 스냅샷에 액세스하는 공유 고객 관리형 키의 사용자에게 키에 대해 다음 작업을 수행할 수 있는 권한을 부여해야 합니다.

- kms:DescribeKey
- kms:CreateGrant
- kms:GenerateDataKey
- kms:GenerateDataKeyWithoutPlaintext
- kms:ReEncrypt
- kms:Decrypt

> **Tip**
>
> 최소 권한의 원칙을 따르려면 kms:CreateGrant에 대한 전체 액세스 권한을 허용하지 마세요. 대신 kms:GrantIsForAWSResource 조건 키를 사용하여 AWS 서비스가 사용자를 대신하여 권한 부여를 생성하는 경우에만 사용자가 KMS 키에 대한 권한 부여를 생성하도록 허용합니다.

고객 관리형 키 액세스 제어에 대한 자세한 내용은AWS Key Management Service 개발자 안내서에서 AWS KMS의 키 정책 사용을 참조하세요.

AWS KMS 콘솔을 사용하여 고객 관리형 키를 공유하려면

1. https://console.aws.amazon.com/kms AWS KMS 콘솔을 엽니다.
2. 를 변경하려면 페이지 오른쪽 상단에 있는 리전 선택기를 AWS 리전사용합니다.
3. 탐색 창에서 고객 관리형 키(Customer managed keys)를 선택합니다.
4. 별칭 열에서 스냅샷을 암호화하는 데 사용한 고객 관리형 키의 별칭(텍스트 링크)을 선택합니다. 키 세부 정보가 새 페이지에서 열립니다.
5. Key policy(키 정책) 섹션에는 정책 보기 또는 기본 보기가 표시됩니다. 정책 보기에는 주요 정책 문서가 표시됩니다. 기본 보기에는 키 관리자(Key administrators), 키 삭제(Key deletion), 키 사용(Key Use) 및 기타 AWS 계정(Other accounts)에 대한 섹션이 표시됩니다. 콘솔에서 정책을 생성하고 사용자 지정하지 않은 경우 기본 보기가 표시됩니다. 기본 보기를 사용할 수 없는 경우 정책 보기에서 정책을 수동으로 편집해야 합니다. 자세한 내용은 AWS Key Management Service 개발자 안내서에서 키 정책 보기(콘솔)을 참조하세요.

   액세스할 수 있는 보기에 따라 정책 보기 또는 기본 보기를 사용하여 다음과 같이 정책에 하나 이상의 AWS 계정 IDs 추가합니다.

   - (정책 보기) Edit(편집)를 선택합니다. `"Allow use of the key"` 및 문에 하나 이상의 AWS 계정 IDs를 추가합니다`"Allow attachment of persistent resources"`. 변경 사항 저장을 선택합니다. 다음 예제에서는 AWS 계정 ID444455556666가 정책에 추가됩니다.

```
{
  "Sid": "Allow use of the key",
  "Effect": "Allow",
  "Principal": {"AWS": [
    "arn:aws:iam::111122223333:user/KeyUser",
    "arn:aws:iam::444455556666:root"
  ]},
  "Action": [
    "kms:Encrypt",
    "kms:Decrypt",
    "kms:ReEncrypt*",
    "kms:GenerateDataKey*",
    "kms:DescribeKey"
  ],
  "Resource": "*"
},
{
  "Sid": "Allow attachment of persistent resources",
```

```
  "Effect": "Allow",
  "Principal": {"AWS": [
    "arn:aws:iam::111122223333:user/KeyUser",
    "arn:aws:iam::444455556666:root"
  ]},
  "Action": [
    "kms:CreateGrant",
    "kms:ListGrants",
    "kms:RevokeGrant"
  ],
  "Resource": "*",
  "Condition": {"Bool": {"kms:GrantIsForAWSResource": true}}
}
```

- (기본 보기) 다른 AWS 계정까지 아래로 스크롤합니다. 다른 AWS 계정 추가를 선택하고 메시지가 표시되면 AWS 계정 ID를 입력합니다. 다른 계정을 추가하려면 다른 AWS 계정 추가를 선택하고 AWS 계정 ID를 입력합니다. AWS 계정을 모두 추가했으면 [변경 사항 저장(Save changes)]을 선택합니다.

자신에게 공유된 Amazon EBS 스냅샷 사용

암호화되지 않은 공유 스냅샷을 사용하려면

ID 또는 설명으로 공유 스냅샷을 찾습니다. 계정에 소유한 다른 스냅샷과 마찬가지로 이 스냅샷을 사용할 수 있습니다. 예를 들어 스냅샷에서 볼륨을 생성하거나 볼륨을 다른 리전으로 복사할 수 있습니다.

암호화된 공유 스냅샷을 사용하려면

ID 또는 설명으로 공유 스냅샷을 찾습니다. 계정에 공유 스냅샷의 복사본을 생성하고 소유한 KMS 키로 복사본을 암호화합니다. 그런 다음 복사본을 사용하여 볼륨을 생성하거나 다른 리전에 복사할 수 있습니다.

Console

스냅샷 권한을 보려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [Snapshots]를 선택합니다.
3. 스냅샷을 선택합니다.

4. 필터를 내가 소유한 경우 스냅샷은 이 계정이 소유합니다. 필터가 프라이빗 스냅샷인 경우 스냅샷은 이 계정이 소유하거나 이 계정과 구체적으로 공유됩니다. 스냅샷을 선택하고 세부 정보 탭에서 소유자가 이 계정을 지정하는지 또는 다른 계정을 지정하는지 여부를 확인합니다.

AWS CLI

스냅샷 권한을 보려면

describe-snapshot-attribute 명령을 사용하여 지정된 스냅샷의 스냅샷 권한을 가져옵니다.

```
aws ec2 describe-snapshot-attribute \
    --snapshot-id snap-0abcdef1234567890 \
    --attribute createVolumePermission
```

다음은 예제 출력입니다.

```
{
    "SnapshotId": "snap-0abcdef1234567890",
    "CreateVolumePermissions": [
        {
            "UserId": "111122223333"
        }
    ]
}
```

PowerShell

스냅샷 권한을 보려면

Get-EC2SnapshotAttribute cmdlet을 사용합니다.

```
(Get-EC2SnapshotAttribute `
    -SnapshotId snap-0abcdef1234567890 `
    -Attribute createVolumePermission).CreateVolumePermissions
```

다음은 예제 출력입니다.

```
Group UserId
----- ------
      111122223333
```

공유한 스냅샷의 사용 확인

AWS CloudTrail 를 사용하여 다른 사용자와 공유한 스냅샷이 복사되는지 또는 볼륨을 생성하는 데 사용되는지 모니터링할 수 있습니다. 공유한 스냅샷에서 작업이 수행되면 CloudTrail에 다음 이벤트가 로깅됩니다.

- SharedSnapshotCopyInitiated — 공유 스냅샷이 복사되고 있습니다.
- SharedSnapshotVolumeCreated — 공유 스냅샷이 볼륨을 생성하는 데 사용되고 있습니다.

CloudTrail 사용에 대한 자세한 내용은 를 사용하여 Amazon EC2 및 Amazon EBS API 호출 로깅 AWS CloudTrail을 참조하세요.

Amazon EBS 스냅샷 아카이브

Amazon EBS 스냅샷 아카이브는 자주 또는 빠르게 검색할 필요가 없는 거의 액세스하지 않는 스냅샷의 저렴한 장기 스토리지에 사용할 수 있는 스토리지 계층입니다.

기본적으로 스냅샷을 생성하면 Amazon EBS 스냅샷 표준 계층(표준 계층)에 저장됩니다. 표준 계층에 저장된 스냅샷은 증분적입니다. 가장 최근의 스냅샷 이후에 변경된 볼륨의 블록만 저장됨을 의미합니다.

스냅샷을 아카이빙하면 증분 스냅샷이 전체 스냅샷으로 변환되고 표준 계층에서 Amazon EBS 스냅샷 아카이브 계층으로 이동됩니다(아카이브 계층). 전체 스냅샷에는 스냅샷 생성 시 볼륨에 기록된 모든 블록이 포함됩니다.

아카이빙된 스냅샷을 액세스해야 하는 경우 해당 스냅샷을 아카이브 계층에서 표준 계층으로 복원한 다음 계정의 다른 스냅샷을 사용하는 것과 동일한 방식으로 사용할 수 있습니다.

Amazon EBS 스냅샷 아카이브는 90일 이상 저장할 계획이고 액세스할 필요가 거의 없는 스냅샷에 대해 최대 75% 더 낮은 스냅샷 스토리지 비용을 제공합니다.

몇 가지 일반적인 사용 사례는 다음과 같습니다.

- 프로젝트 종료 스냅샷과 같은 볼륨의 유일한 스냅샷 아카이빙
- 규정 준수를 위해 전체 시점 증분 스냅샷 아카이빙
- 월별, 분기별 또는 연간 증분 스냅샷 아카이빙

주제

- 할당량

- Amazon EBS 스냅샷 아카이빙 시 고려 사항 및 제한 사항
- Amazon EBS 스냅샷 아카이빙 요금 및 결제
- Amazon EBS 스냅샷 아카이빙 지침 및 모범 사례
- Amazon EBS 스냅샷 아카이빙에 필요한 IAM 권한
- Amazon EBS 스냅샷 아카이빙
- 아카이브된 Amazon EBS 스냅샷 복원
- 임시로 복원된 Amazon EBS 스냅샷의 복원 기간 수정
- 아카이브된 Amazon EBS 스냅샷 보기
- CloudWatch Events를 사용하여 Amazon EBS 스냅샷 아카이빙 모니터링

할당량

이 섹션에서는 아카이빙된 스냅샷과 진행 중인 스냅샷의 기본 할당량에 대해 설명합니다.

할당량	기본 할당량			
볼륨당 아카이빙된 스냅샷	25			
계정당 동시 진행 중인 스냅샷 아카이브	25			
계정당 동시 진행 중인 스냅샷 복원	5			

기본 한도를 초과해야 하는 경우 지원 Center 사례 생성 양식을 작성하여 한도 증가를 요청합니다.

Amazon EBS 스냅샷 아카이빙 시 고려 사항 및 제한 사항

Amazon EBS 스냅샷을 아카이빙할 때는 다음 사항에 유의해야 합니다.

고려 사항

- 최소 아카이브 기간은 90일입니다. 최소 아카이브 기간이 90일 이전에 아카이빙된 스냅샷을 삭제하거나 영구적으로 복원할 경우 아카이브 계층의 남은 일수에 대한 요금이 가장 가까운 시간으로 반올림됩니다. 자세한 내용은 Amazon EBS 스냅샷 아카이빙 요금 및 결제 단원을 참조하십시오.
- 스냅샷의 크기에 따라 아카이브 계층에서 표준 계층으로 아카이빙된 스냅샷을 복원하는 데 최대 72시간이 걸릴 수 있습니다.
- 아카이빙된 스냅샷은 항상 전체 스냅샷입니다. 전체 스냅샷에는 스냅샷 생성 시 볼륨에 기록된 모든 블록이 포함됩니다. 증분 스냅샷에서 생성된 전체 스냅샷이 증분 스냅샷보다 클 수 있습니다. 그러나 표준 계층에 볼륨의 스냅샷이 하나만 있는 경우 아카이브 계층에 있는 전체 스냅샷의 크기는 표준 계층의 스냅샷과 동일합니다. 이는 볼륨의 첫 번째 스냅샷이 항상 전체 스냅샷이기 때문입니다. 전체 스냅샷 크기를 가져오려면 describe-snapshots AWS CLI 명령을 사용합니다.
- 월별, 분기별 또는 연도별 스냅샷 아카이빙 작업을 권장합니다. 단일 볼륨의 일일 증분 스냅샷을 보관하면 표준 계층에 보관하는 것보다 비용이 많이 들 수도 있습니다.
- 스냅샷이 아카이빙되면 스냅샷 계보의 다른 스냅샷에서 참조하는 스냅샷의 데이터가 표준 계층에 보관됩니다. 표준 계층에 보관된 참조 데이터와 관련된 데이터 및 스토리지 비용은 계보의 다음 스냅샷에 할당됩니다. 이렇게 하면 계보의 후속 스냅샷이 아카이브의 영향을 받지 않습니다.
- 휴지통 보존 규칙과 일치하는 아카이빙된 스냅샷을 삭제하면 보존 규칙에 정의된 보존 기간 동안 아카이빙된 스냅샷이 휴지통에 보관됩니다. 스냅샷을 사용하려면 먼저 휴지통에서 스냅샷을 복구한 다음 아카이브 계층에서 복원해야 합니다. 자세한 내용은 휴지통과 Amazon EBS 스냅샷 아카이빙 요금 및 결제을 참조하세요.
- 보관된 스냅샷은 블록 디바이스 매핑에서 사용하거나 Amazon EBS 볼륨을 생성하는 데 사용할 수 없습니다.
- AWS Backup 콘솔, APIs 또는 명령줄 도구를 AWS Backup 사용하여 생성된 스냅샷을 아카이브할 수 있습니다. 자세한 내용은 AWS Backup 개발자 안내서의 백업 계획 생성을 참조하세요.

제한 사항

- `completed` 상태의 스냅샷만 아카이빙할 수 있습니다.
- 계정에서 소유한 스냅샷만 아카이빙할 수 있습니다. 공유되는 스냅샷을 아카이빙하려면 먼저 스냅샷을 계정에 복사한 다음 스냅샷 사본을 아카이빙합니다.

- 아카이빙된 스냅샷을 사용하려면 먼저 표준 계층으로 복원해야 합니다. 스냅샷을 공유 또는 복사할 뿐만 아니라 `CreateVolume` 및 `RunInstances` API 작업을 통해 스냅샷에서 볼륨을 생성하려면 표준 계층으로 복원해야 합니다. 자세한 내용은 아카이브된 Amazon EBS 스냅샷 복원 단원을 참조하십시오.
- 연결된 AMI가 모두 비활성화된 경우에만 하나 이상의 AMI와 연결된 스냅샷을 보관할 수 있습니다. 자세한 내용은 AMI 활성화를 참조하세요.
- 연결된 스냅샷이 일시적으로 복원된 경우 비활성화된 AMI를 활성화할 수 없습니다. AMI를 활성화하기 전에 모든 연결된 스냅샷을 영구적으로 복원해야 합니다.
- 스냅샷 아카이브 또는 스냅샷 복원 프로세스를 시작한 후에는 취소할 수 없습니다.
- 아카이빙된 스냅샷을 공유할 수 없습니다. 다른 계정과 공유한 스냅샷을 아카이빙하는 경우 스냅샷을 공유하는 계정은 스냅샷이 아카이빙된 후 액세스 권한을 잃게 됩니다.
- 아카이빙된 스냅샷을 복사할 수 없습니다. 아카이빙된 스냅샷을 복사해야 하는 경우 먼저 복원해야 합니다.
- 아카이빙된 스냅샷에 대해 빠른 스냅샷 복원을 사용할 수 없습니다. 스냅샷이 아카이빙되면 빠른 스냅샷 복원이 자동으로 비활성화됩니다. 빠른 스냅샷 복원을 사용해야 하는 경우 스냅샷을 복원한 후 수동으로 활성화해야 합니다.

Amazon EBS 스냅샷 아카이빙 요금 및 결제

아카이빙된 스냅샷은 월별 GB당 0.0125 USD의 요금이 청구됩니다. 예를 들어 100GiB 스냅샷을 아카이빙하는 경우 매월 1.25 USD(100GiB * 0.0125 USD)가 청구됩니다.

스냅샷 복원은 복원된 데이터 1GB당 0.03 USD의 요금이 청구됩니다. 예를 들어 아카이브 계층에서 100GiB 스냅샷을 복원하는 경우 3 USD(100GiB * 0.03 USD)에 대해 한 번 청구됩니다.

스냅샷이 표준 계층으로 복원되면 스냅샷은 월별 GB당 0.05 USD의 스냅샷에 대해 표준 요금으로 청구됩니다.

자세한 내용은 Amazon EBS 요금을 참조하세요.

최소 아카이브 기간에 대한 청구

최소 아카이브 기간은 90일입니다. 최소 아카이브 기간인 90일 전에 아카이빙된 스냅샷을 삭제하거나 영구적으로 복원하는 경우 남은 일수에 대하여 아카이브 계층 스토리지 요금과 같이 비례적으로 청구되며 가장 가까운 시간으로 반올림됩니다. 예를 들어 40일 후에 아카이빙된 스냅샷을 삭제하거나 영구적으로 복원하는 경우 최소 아카이브 기간의 나머지 50일에 대한 요금이 청구됩니다.

> **Note**
>
> 최소 아카이브 기간인 90일 전에 아카이빙된 스냅샷을 임시로 복원하는 경우에는 이 요금이 부과되지 않습니다.

임시 복원

스냅샷을 임시로 복원하면 스냅샷이 아카이브 계층에서 표준 계층으로 복원되고 스냅샷 사본은 아카이브 계층에 유지됩니다. 임시 복원 기간 동안 표준 계층의 스냅샷과 아카이브 계층의 스냅샷 사본에 대한 비용이 모두 청구됩니다. 임시로 복원된 스냅샷이 표준 계층에서 제거되면 해당 스냅샷에 대한 요금이 더 이상 청구되지 않으며 아카이브 계층의 스냅샷에 대해서만 요금이 청구됩니다.

영구 복원

스냅샷을 영구적으로 복원하면 스냅샷이 아카이브 계층에서 표준 계층으로 복원되고 스냅샷은 아카이브 계층에서 삭제됩니다. 표준 계층의 스냅샷에 대해서만 요금이 청구됩니다.

스냅샷 삭제

아카이빙하는 동안 스냅샷을 삭제하면 이미 아카이브 계층으로 이동한 스냅샷 데이터에 대한 요금이 청구됩니다. 이 데이터에는 최소 아카이브 기간이 90일이며 삭제 시 그에 따라 요금이 청구됩니다. 예를 들어 100GiB 스냅샷을 아카이빙하는 경우 40GiB만 아카이빙된 후에 스냅샷을 삭제하면 이미 아카이빙된 40GiB에 대한 최소 아카이브 기간인 90일 동안 1.50 USD가 청구됩니다(월 0.0125 USD/GB * 40GB * (90일 * 24시간) / (24시간/일 * 월 30일)).

아카이브 계층에서 복원하는 동안 스냅샷을 삭제하면 스냅샷의 전체 크기에 대한 스냅샷 복원 요금이 청구됩니다(스냅샷 크기* 0.03 USD). 예를 들어 아카이브 계층에서 100GiB 스냅샷을 복원하는 경우 스냅샷 복원이 완료되기 전에 언제든지 스냅샷을 삭제하면 3 USD(100GiB 스냅샷 크기* 0.03 USD)가 청구됩니다.

휴지통

아카이빙된 스냅샷은 휴지통에 있는 동안 아카이빙된 스냅샷에 대한 요금으로 청구됩니다. 휴지통의 아카이빙된 스냅샷은 최소 아카이브 기간인 90일이 적용되며 최소 아카이브 기간 전에 휴지통에서 삭제되면 그에 따라 요금이 청구됩니다. 즉, 보존 규칙이 최소 기간인 90일 전에 아카이빙된 스냅샷을 휴지통에서 삭제하면 남은 일수에 대한 요금이 청구됩니다.

스냅샷을 아카이빙하면서 보존 규칙과 일치하는 스냅샷을 삭제하면 보존 규칙에 정의된 보존 기간 동안 아카이빙된 스냅샷이 휴지통에 보관됩니다. 아카이빙된 스냅샷에 대한 요금이 청구됩니다.

스냅샷을 복원하면서 보존 규칙과 일치하는 스냅샷을 삭제하면 복원된 스냅샷은 보존 기간의 나머지 기간 동안 휴지통에 보관되고 표준 스냅샷 요금으로 청구됩니다. 복원된 스냅샷을 사용하려면 먼저 휴지통에서 스냅샷을 복구해야 합니다.

자세한 내용은 휴지통을 참조하세요.

비용 추적

보관된 스냅샷은 리소스 ID와 Amazon 리소스 이름(ARN) AWS Cost and Usage Report 이 동일한에 나타납니다. 자세한 내용은 AWS Cost and Usage Report 사용 설명서를 참조하십시오.

다음 사용 유형을 사용하여 관련 비용을 식별할 수 있습니다.

- `SnapshotArchiveStorage` - 월별 데이터 스토리지 요금
- `SnapshotArchiveRetrieval` - 스냅샷 복원에 대한 일회성 요금
- `SnapshotArchiveEarlyDelete` - 최소 아카이브 기간(90일) 전에 스냅샷을 삭제하거나 영구적으로 복원하는 경우 요금

Amazon EBS 스냅샷 아카이빙 지침 및 모범 사례

이 섹션에서는 스냅샷 아카이빙에 대한 지침 및 모범 사례를 제공합니다.

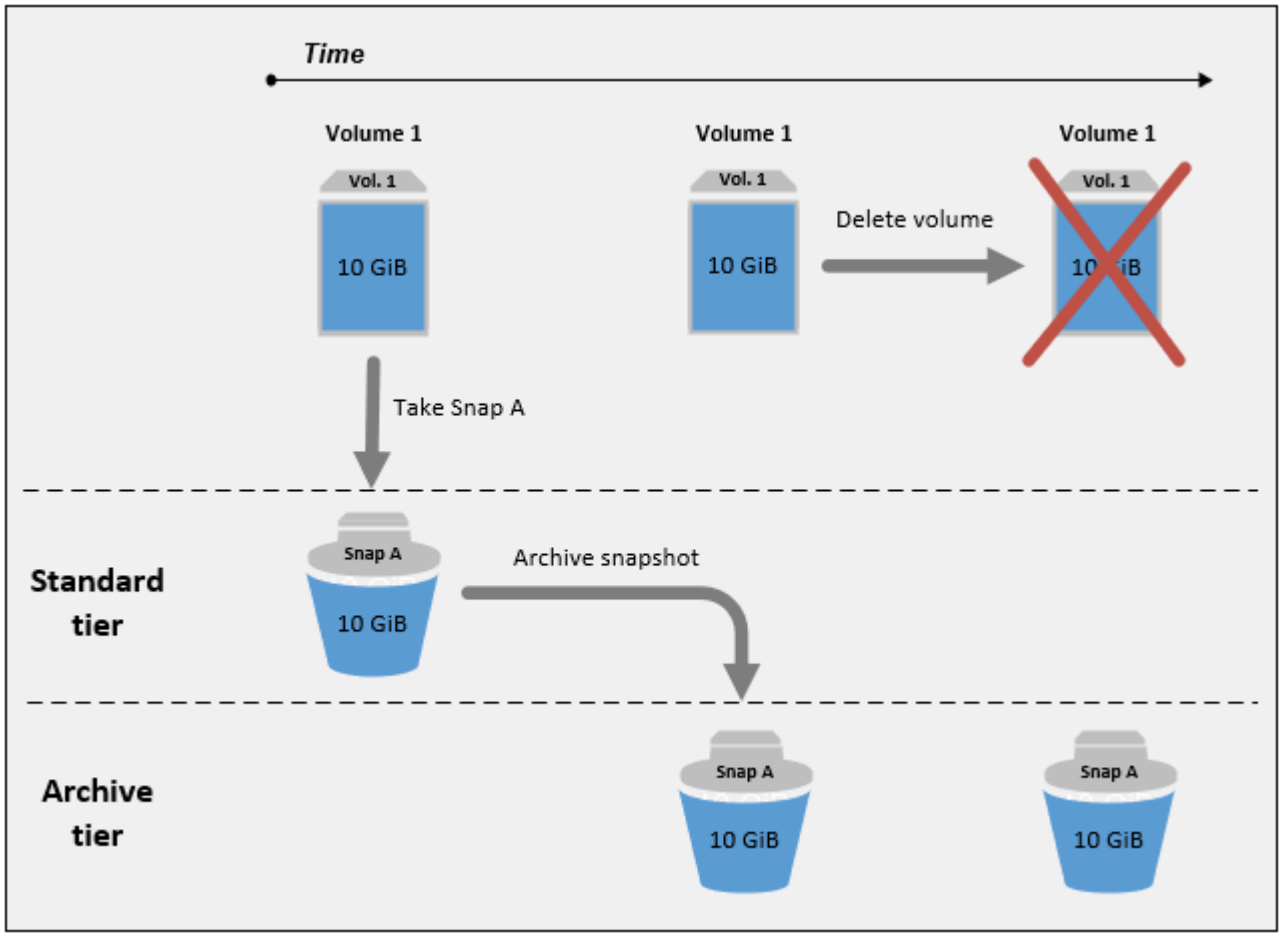
주제

- 볼륨의 유일한 스냅샷 아카이빙
- 단일 볼륨의 증분 스냅샷 아카이빙
- 규정 준수를 위해 전체 스냅샷 아카이빙
- 표준 계층 스토리지 비용 절감 결정

볼륨의 유일한 스냅샷 아카이빙

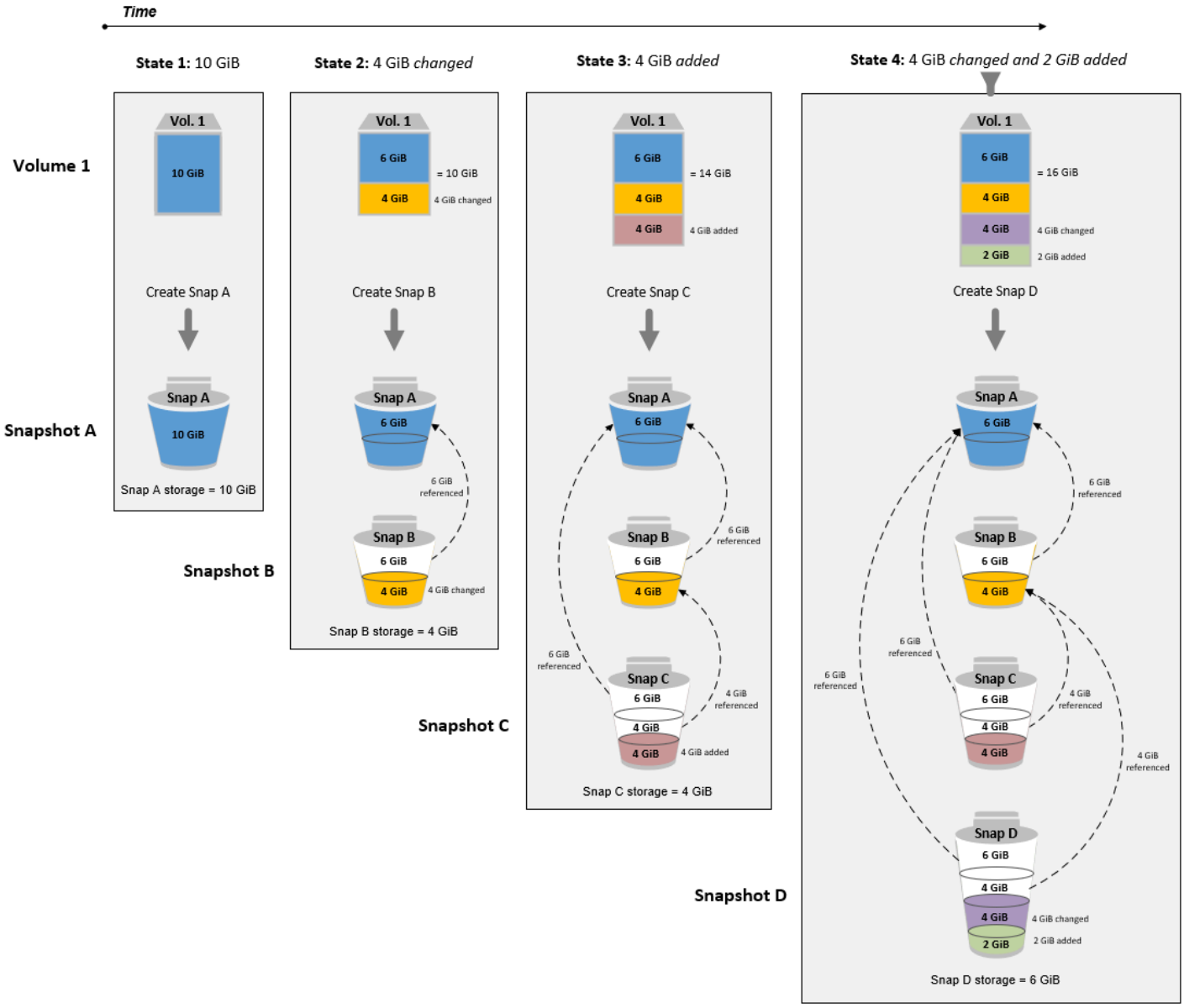
볼륨의 스냅샷이 하나만 있는 경우 스냅샷은 생성 시 볼륨에 기록된 블록과 항상 같은 크기입니다. 이러한 스냅샷을 아카이빙하면 표준 계층의 스냅샷이 동일한 크기의 전체 스냅샷으로 변환되고 표준 계층에서 아카이브 계층으로 이동됩니다.

이러한 스냅샷을 아카이빙하면 스토리지 비용을 절감할 수 있습니다. 소스 볼륨이 더 이상 필요하지 않으면 볼륨을 삭제하여 스토리지 비용을 추가로 절감할 수 있습니다.



단일 볼륨의 증분 스냅샷 아카이빙

증분 스냅샷을 아카이빙하면 스냅샷이 전체 스냅샷으로 변환되고 아카이브 계층으로 이동됩니다. 예를 들어 다음 이미지에서 스냅 B를 아카이빙하는 경우 스냅샷은 크기가 10GiB인 전체 스냅샷으로 변환되고 아카이브 계층으로 이동됩니다. 마찬가지로 스냅 C를 아카이빙하는 경우 아카이브 계층의 전체 스냅샷 크기는 14GiB입니다.



표준 계층에서 스토리지 비용을 줄이기 위해 스냅샷을 아카이빙하는 경우 증분 스냅샷 세트의 첫 번째 스냅샷을 아카이빙하면 안 됩니다. 이러한 스냅샷은 스냅샷 계보의 후속 스냅샷에서 참조됩니다. 대부분의 경우 이러한 스냅샷을 아카이빙해도 스토리지 비용이 줄지 않습니다.

> **Note**
>
> 증분 스냅샷 세트의 마지막 스냅샷을 아카이빙하면 안 됩니다. 마지막 스냅샷은 볼륨에서 가장 최근에 생성된 스냅샷입니다. 볼륨이 손상되거나 손실되었으며 스냅샷에서 볼륨을 생성하려는 경우 표준 계층에 이 스냅샷이 필요합니다.

계보의 이후 스냅샷에서 참조하는 데이터가 포함된 스냅샷을 아키이빙하는 경우 참조된 데이터와 관련된 데이터 스토리지 및 스토리지 비용이 계보의 이후 스냅샷에 할당됩니다. 이 경우 스냅샷을 아카이빙해도 데이터 스토리지 또는 스토리지 비용이 줄지는 않습니다. 예를 들어 이전 이미지에서 스냅 B를 아카이빙하는 경우 4GiB의 데이터는 스냅 C에 기인합니다. 이 경우 아카이브 계층에서 스냅 B의 전체 버전에 대한 스토리지 비용이 발생하기 때문에 전체 스토리지 비용이 증가하지만 표준 계층에 대한 스토리지 비용은 변경되지 않습니다.

스냅 C를 아카이빙하는 경우 표준 계층 스토리지는 나중에 계보의 다른 스냅샷에서 데이터를 참조하지 않기 때문에 4GiB 감소합니다. 또한 스냅샷이 전체 스냅샷으로 변환되므로 아카이브 계층 스토리지가 14GiB 증가합니다.

규정 준수를 위해 전체 스냅샷 아카이빙

규정 준수를 위해 월별, 분기별 또는 연간 단위로 볼륨의 전체 백업을 생성해야 할 수 있습니다. 이러한 백업의 경우 스냅샷 계보의 다른 스냅샷에 대한 역방향 또는 정방향 참조가 없는 독립 실행형 스냅샷이 필요할 수 있습니다. EBS 스냅샷 아카이브를 사용하여 아카이빙된 스냅샷은 전체 스냅샷이며 계보 내의 다른 스냅샷에 대한 참조가 없습니다. 또한 규정 준수를 위해 이러한 스냅샷을 몇 년 동안 보관해야 할 수도 있습니다. EBS 스냅샷 아카이브를 사용하면 장기 보존을 위해 이러한 전체 스냅샷을 경제적으로 아카이빙할 수 있습니다.

표준 계층 스토리지 비용 절감 결정

스토리지 비용을 절감하기 위해 증분 스냅샷을 아카이빙하려면 아카이브 계층의 전체 스냅샷 크기와 표준 계층의 스토리지 감소를 고려해야 합니다. 이 섹션에서는 이 작업을 수행하는 방법을 설명합니다.

> **Important**
>
> API 응답은 API가 호출된 시점의 정확한 데이터입니다. 스냅샷 계보의 변경으로 인해 스냅샷과 연결된 데이터가 변경됨에 따라 API 응답이 다를 수 있습니다.

표준 계층의 스토리지 및 스토리지 비용 절감을 확인하려면 다음 단계를 따르세요.

1. 아카이빙할 스냅샷에 대해 전체 스냅샷 크기와 해당 스냅샷이 생성된 소스 볼륨을 확인합니다. describe-snapshots 명령을 사용하고 `--snapshot-id`에 아카이빙할 스냅샷의 ID를 지정합니다.

   ```
   $ aws ec2 describe-snapshots --snapshot-id snapshot_id
   ```

   `FullSnapshotSizeInBytes` 응답 값은 전체 스냅샷 크기를 바이트 단위로 나타내며 `VolumeId` 응답 값은 소스 볼륨의 ID를 나타냅니다.

예를 들어 다음 명령은 스냅샷 snap-09c9114207084f0d9에 대한 정보를 반환합니다.

```
$ aws ec2 describe-snapshots --snapshot-id snap-09c9114207084f0d9
```

다음 예제 출력은 전체 스냅샷 크기가 5678912341바이트(5.28GiB)이고 소스 볼륨이 vol-0f3e2c292c52b85c3임을 보여줍니다.

```
{
    "Snapshots": [
        {
            "Description": "",
            "Tags": [],
            "Encrypted": false,
            "VolumeId": "vol-0f3e2c292c52b85c3",
            "State": "completed",
            "VolumeSize": 8,
            "StartTime": "2021-11-16T08:29:49.840Z",
            "Progress": "100%",
            "OwnerId": "123456789012",
            "FullSnapshotSizeInBytes" : "5678912341",
            "SnapshotId": "snap-09c9114207084f0d9"
        }
    ]
}
```

2. 소스 볼륨에서 생성된 모든 스냅샷을 찾습니다. describe-snapshots 명령을 사용합니다. volume-id 필터를 지정하고, 필터 값으로 이전 단계의 볼륨 ID를 지정합니다.

```
$ aws ec2 describe-snapshots --filters "Name=volume-id, Values=volume_id"
```

예를 들어 다음 명령은 볼륨 vol-0f3e2c292c52b85c3에서 생성된 모든 스냅샷을 반환합니다.

```
$ aws ec2 describe-snapshots --filters "Name=volume-id,
 Values=vol-0f3e2c292c52b85c3"
```

다음은 명령 출력이며 3개의 스냅샷이 볼륨 vol-0f3e2c292c52b85c3에서 생성되었음을 나타냅니다.

```
{
```

```
    "Snapshots": [
        {
            "Description": "",
            "Tags": [],
            "Encrypted": false,
            "VolumeId": "vol-0f3e2c292c52b85c3",
            "State": "completed",
            "VolumeSize": 8,
            "StartTime": "2021-11-14T08:57:39.300Z",
            "Progress": "100%",
            "OwnerId": "123456789012",
            "SnapshotId": "snap-08ca60083f86816b0"
        },
        {
            "Description": "",
            "Tags": [],
            "Encrypted": false,
            "VolumeId": "vol-0f3e2c292c52b85c3",
            "State": "completed",
            "VolumeSize": 8,
            "StartTime": "2021-11-15T08:29:49.840Z",
            "Progress": "100%",
            "OwnerId": "123456789012",
            "SnapshotId": "snap-09c9114207084f0d9"
        },
        {
            "Description": "01",
            "Tags": [],
            "Encrypted": false,
            "VolumeId": "vol-0f3e2c292c52b85c3",
            "State": "completed",
            "VolumeSize": 8,
            "StartTime": "2021-11-16T07:50:08.042Z",
            "Progress": "100%",
            "OwnerId": "123456789012",
            "SnapshotId": "snap-024f49fe8dd853fa8"
        }
    ]
}
```


3. 이전 명령의 출력을 사용하여 스냅샷을 생성 시간별로 가장 오래된 항목에서 최신 항목 순으로 정렬합니다. 각 스냅샷에 대한 `StartTime` 응답 파라미터는 UTC 시간 형식으로 생성 시간을 나타냅니다.

예를 들어 생성 시간별로 가장 오래된 항목에서 최신 항목 순으로 정렬된 이전 단계에서 반환된 스냅샷은 다음과 같습니다.

1. snap-08ca60083f86816b0(가장 오래된 항목 - 아카이빙할 스냅샷 전에 생성됨)
2. snap-09c9114207084f0d9(아카이빙할 스냅샷)
3. snap-024f49fe8dd853fa8(최신 - 아카이빙하려는 스냅샷 후에 생성됨)

4. 아카이빙할 스냅샷 직전과 직후에 생성된 스냅샷을 식별합니다. 이 경우 3개의 스냅샷 세트에서 생성된 것 중 두 번째 증분 스냅샷인 snap-09c9114207084f0d9를 아카이빙하려고 합니다. 스냅샷 snap-08ca60083f86816b0은 직전에 생성되었으며 스냅샷 snap-024f49fe8dd853fa8은 직후에 생성되었습니다.

5. 아카이빙할 스냅샷에서 참조되지 않은 데이터를 찾습니다. 먼저 아카이빙할 스냅샷 직전에 생성된 스냅샷과 아카이빙할 스냅샷 간에 다른 블록을 찾습니다. list-changed-blocks 명령을 사용합니다. --first-snapshot-id에서 아카이빙할 스냅샷 직전에 생성된 스냅샷의 ID를 지정합니다. --second-snapshot-id에서 아카이빙할 스냅샷의 ID를 지정합니다.

```
$ aws ebs list-changed-blocks --first-snapshot-id snapshot_created_before --second-snapshot-id snapshot_to_archive
```

예를 들어 다음 명령은 스냅샷 snap-08ca60083f86816b0(아카이빙할 스냅샷 전에 생성된 스냅샷)과 스냅샷 snap-09c9114207084f0d9(아카이빙할 스냅샷) 간에 다른 블록에 대한 블록 인덱스를 보여줍니다.

```
$ aws ebs list-changed-blocks --first-snapshot-id snap-08ca60083f86816b0 --second-snapshot-id snap-09c9114207084f0d9
```

다음은 일부 블록이 생략된 명령 출력을 보여줍니다.

```
{
    "BlockSize": 524288,
    "ChangedBlocks": [
        {
            "FirstBlockToken": "ABgBAX6y
+WH6Rm9y5zq1VyeTCmEzGmTT0jNZG1cDirFq1rOVeFbWXsH3W4z/",
            "SecondBlockToken": "ABgBASyx0bHHBnTERu
+9USLxYK/81UT0dbHIUFqUjQUkwTwK5qkjP8NSGyNB",
            "BlockIndex": 4
        },
```

```
        {
            "FirstBlockToken": "ABgBAcfL
+EfmQmlNgstqrFnYgsAxR4SDSO4LkNLYOOChGBWcfJnpn90E9XX1",
            "SecondBlockToken": "ABgBAdX0mtX6aBAt3EBy
+8jFCESMpig7csKjbO2Ocd08m2iNJV2Ue+cRwUqF",
            "BlockIndex": 5
        },
        {
            "FirstBlockToken": "ABgBAVBaFJmbP/eRHGh7vnJlAwyiyNUi3MKZmEMxs2wC3AmM/
fc6yCOAMb65",
            "SecondBlockToken":
 "ABgBAdewWkHKTcrhZmsfM7GbaHyXD1Ctcn2nppz4wYItZRmAo1M72fpXU0Yv",
            "BlockIndex": 13
        },
        {
            "FirstBlockToken": "ABgBAQGxwuf6z095L6DpRoVRVnOqPxmx9r7Wf6O+i
+ltZ0dwPpGN39ijztLn",
            "SecondBlockToken": "ABgBAUdlitCVI7c6hGsT4ckkKCw6bMRclnV
+bKjViu/9UESTcW7CD9w4J2td",
            "BlockIndex": 14
        },
        {
            "FirstBlockToken":
 "ABgBAZBfEv4EHS1aSXTXxSE3mBZG6CNeIkwxpljzmgSHICGlFmZCyJXzE4r3",
            "SecondBlockToken":
 "ABgBAVWR7QuQQB0AP2TtmNkgS4Aec5KAQVCldnpc91zBiNmSfW9ouIlbeXWy",
            "BlockIndex": 15
        },
        .....
        {
            "SecondBlockToken": "ABgBAeHwXPL+z3DBLjDhwjdAM9+CPGV5VO5Q3rEEA
+ku50P498hjnTAgMhLG",
            "BlockIndex": 13171
        },
        {
            "SecondBlockToken":
 "ABgBAbZcPiVtLx6U3Fb4lAjRdrkJMwW5M2tiCgIp6ZZpcZ8AwXxkjVUUHADq",
            "BlockIndex": 13172
        },
        {
            "SecondBlockToken": "ABgBAVmEd/pQ9VW9hWiOujOAKcauOnUFCO
+eZ5ASVdWLXWWC04ijfoDTpTVZ",
            "BlockIndex": 13173
        },
```

```
        {
            "SecondBlockToken": "ABgBAT/jeN7w
+8ALuNdaiwXmsSfM6tOvMoLBLJ14LKvavw4IiB1d0iykWe6b",
            "BlockIndex": 13174
        },
        {
            "SecondBlockToken": "ABgBAXtGvUhTjjUqkwKXfXzyR2GpQei/
+pJSG/19ESwvt7Hd8GHaUqVs6Zf3",
            "BlockIndex": 13175
        }
    ],
    "ExpiryTime": 1637648751.813,
    "VolumeSize": 8
}
```

그런 다음 동일한 명령을 사용하여 아카이빙할 스냅샷과 그 직후에 생성된 스냅샷 간에 다른 블록을 찾습니다. `--first-snapshot-id`에서 아카이빙할 스냅샷의 ID를 지정합니다. `--second-snapshot-id`에서 아카이빙할 스냅샷 직후에 생성된 스냅샷의 ID를 지정합니다.

```
$ aws ebs list-changed-blocks --first-snapshot-id snapshot_to_archive --second-
snapshot-id snapshot_created_after
```

예를 들어 다음 명령은 스냅샷 `snap-09c9114207084f0d9`(아카이빙할 스냅샷)와 스냅샷 `snap-024f49fe8dd853fa8`(아카이빙할 스냅샷 후에 생성된 스냅샷) 간에 다른 블록의 블록 인덱스를 보여줍니다.

```
$ aws ebs list-changed-blocks --first-snapshot-id snap-09c9114207084f0d9 --second-
snapshot-id snap-024f49fe8dd853fa8
```

다음은 일부 블록이 생략된 명령 출력을 보여줍니다.

```
{
    "BlockSize": 524288,
    "ChangedBlocks": [
        {
            "FirstBlockToken": "ABgBAVax0bHHBnTERu
+9USLxYK/81UT0dbSnkDk0gqwRFSFGWA7HYbkkAy5Y",
            "SecondBlockToken":
 "ABgBASEvi9x8Om7Htp37cKG2NT9XUzEbLHpGcayelomSoHpGy8LGyvG0yYfK",
            "BlockIndex": 4
```

```
        },
        {
            "FirstBlockToken": "ABgBAeL0mtX6aBAt3EBy+8jFCESMpig7csfMrI4ufnQJT3XBm/
pwJZ1n2Uec",
            "SecondBlockToken": "ABgBAXmUTg6rAI
+v0LvekshbxCVpJjWILvxgC0AG0GQBEUNRVHkNABBwXLkO",
            "BlockIndex": 5
        },
        {
            "FirstBlockToken":
 "ABgBATKwWkHKTcrhZmsfM7GbaHyXD1CtcnjIZv9YzisYsQTMHfTfh4AhS0s2",
            "SecondBlockToken": "ABgBAcmiPFovWgXQio
+VBrxOqGy4PKZ9SAAHaZ2HQBM9fQQU0+EXxQjVGv37",
            "BlockIndex": 13
        },
        {
            "FirstBlockToken":
 "ABgBAbRlitCVI7c6hGsT4ckkKCw6bMRclnARrMt1hUbIhFnfz8kmUaZOP2ZE",
            "SecondBlockToken": "ABgBAXe935n544+rxhJ0INB8q7pAeoPZkkD27vkspE/
qKyvOwpozYII6UNCT",
            "BlockIndex": 14
        },
        {
            "FirstBlockToken": "ABgBAd+yxCO26I
+1Nm2KmuKfrhjCkuaP6LXuol3opCNk6+XRGcct4suBHje1",
            "SecondBlockToken": "ABgBAcPpnXz821NtTvWBPTz8uUFXnS8jXubvghEjZulIjHgc
+7saWys77shb",
            "BlockIndex": 18
        },
        .....
        {
            "SecondBlockToken": "ABgBATni4sDE5rS8/a9pqV03lU/lKCW
+CTxFl3cQ5p2f2h1njpuUiGbqKGUa",
            "BlockIndex": 13190
        },
        {
            "SecondBlockToken": "ABgBARbXo7zFhu7IEQ/9VMYFCTCtCuQ
+iSlWVpBIshmeyeS5FD/M0i64U+a9",
            "BlockIndex": 13191
        },
        {
            "SecondBlockToken": "ABgBAZ8DhMk+rROXa4dZlNK45rMYnVIGGSyTeiMli/sp/
JXUVZKJ9sMKIsGF",
            "BlockIndex": 13192
```

```
        },
        {
            "SecondBlockToken":
 "ABgBATh6MBVE904l6sqOC27s1nVntFUpDwiMcRWGyJHy8sIgGL5yuYXHAVty",
            "BlockIndex": 13193
        },
        {
            "SecondBlockToken":
 "ABgBARuZykaFBWpCWrJPXaPCneQMbyVgnITJqj4c1kJWPIj5Gn61OQyy+giN",
            "BlockIndex": 13194
        }
    ],
    "ExpiryTime": 1637692677.286,
    "VolumeSize": 8
}
```

6. 이전 단계의 두 명령에서 반환되는 출력을 비교합니다. 두 명령 출력 모두에 동일한 블록 인덱스가 나타나면 블록에 참조되지 않은 데이터가 포함되어 있는 것입니다.

   예를 들어 이전 단계의 명령 출력은 블록 4, 5, 13 및 14가 스냅샷 `snap-09c9114207084f0d9`에 대해 고유하며 스냅샷 계보의 다른 스냅샷에서 참조하지 않음을 나타냅니다.

   표준 계층 스토리지의 감소를 확인하려면 두 명령 출력에 나타나는 블록 수에 스냅샷 블록 크기인 512KiB를 곱합니다.

   예를 들어 두 명령 출력에 9,950개의 블록 인덱스가 나타나면 표준 계층 스토리지가 약 4.85GiB(9,950개 블록* 512KiB = 4.85GiB) 감소한다는 것을 나타냅니다.

7. 참조되지 않은 블록을 표준 계층에 90일 동안 저장하는 데 드는 스토리지 비용을 결정합니다. 이 값을 1단계에서 설명한 전체 스냅샷을 아카이브 계층에 저장하는 비용과 비교합니다. 최소 90일 동안 아카이브 계층에서 전체 스냅샷을 복원하지 않는다고 가정하면 값을 비교하여 비용 절감액을 확인할 수 있습니다. 자세한 내용은 Amazon EBS 스냅샷 아카이빙 요금 및 결제 단원을 참조하십시오.

Amazon EBS 스냅샷 아카이빙에 필요한 IAM 권한

기본적으로 사용자에게는 스냅샷 아카이빙을 사용할 수 있는 권한이 없습니다. 사용자가 스냅샷 아카이브를 사용하도록 허용하려면 특정 리소스 및 API 작업을 사용할 권한을 부여하는 IAM 정책을 생성해야 합니다. 자세한 내용은 IAM 사용 설명서의 IAM 정책 생성을 참조하세요.

스냅샷 아카이빙을 사용하려면 사용자에게 다음 권한이 필요합니다.

- ec2:DescribeSnapshotTierStatus
- ec2:ModifySnapshotTier
- ec2:RestoreSnapshotTier

콘솔 사용자에게는 ec2:DescribeSnapshots 등의 추가 권한이 필요할 수 있습니다.

암호화된 스냅샷을 아카이브하고 복원하려면 다음과 같은 추가 AWS KMS 권한이 필요합니다.

- kms:CreateGrant
- kms:Decrypt
- kms:DescribeKey

다음은 암호화된 스냅샷과 암호화되지 않은 스냅샷을 보관 및 복원하고 볼 수 있는 권한을 IAM 사용자에게 부여하는 IAM 정책의 예제입니다. 여기에는 콘솔 사용자에 대한 ec2:DescribeSnapshots 권한이 포함됩니다. 일부 권한이 필요하지 않은 경우 정책에서 권한을 제거할 수 있습니다.

> **Tip**
>
> 최소 권한의 원칙을 따르려면 kms:CreateGrant에 대한 전체 액세스 권한을 허용하지 마세요. 대신 다음 예제와 AWS 같이 서비스에 의해 사용자를 대신하여 권한 부여가 생성되는 경우에만 kms:GrantIsForAWSResource 조건 키를 사용하여 사용자가 KMS 키에 대한 권한 부여를 생성하도록 허용합니다.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [{
        "Effect": "Allow",
        "Action": [
            "ec2:DescribeSnapshotTierStatus",
            "ec2:ModifySnapshotTier",
            "ec2:RestoreSnapshotTier",
            "ec2:DescribeSnapshots",
            "kms:CreateGrant",
            "kms:Decrypt",
            "kms:DescribeKey"
```

```
        ],
        "Resource": "*",
        "Condition": {
                "Bool": {
                    "kms:GrantIsForAWSResource": true
                }
            }
    }]
}
```

액세스 권한을 제공하려면 사용자, 그룹 또는 역할에 권한을 추가하세요.

- 의 사용자 및 그룹 AWS IAM Identity Center:

  권한 세트를 생성합니다. AWS IAM Identity Center 사용자 안내서에서 권한 세트 생성의 지침을 따릅니다.
- ID 제공업체를 통해 IAM에서 관리되는 사용자:

  ID 페더레이션을 위한 역할을 생성합니다. IAM 사용자 설명서의 Create a role for a third-party identity provider (federation)의 지침을 따릅니다.
- IAM 사용자:
  - 사용자가 맡을 수 있는 역할을 생성합니다. IAM 사용자 설명서에서 Create a role for an IAM user의 지침을 따릅니다.
  - (권장되지 않음) 정책을 사용자에게 직접 연결하거나 사용자를 사용자 그룹에 추가합니다. IAM 사용 설명서에서 사용자(콘솔)에 권한 추가의 지침을 따릅니다.

Amazon EBS 스냅샷 아카이빙

`completed` 상태이고 계정에서 소유하는 모든 스냅샷을 아카이빙할 수 있습니다. `pending` 또는 `error` 상태의 스냅샷이나 본인과 공유 중인 스냅샷은 아카이빙할 수 없습니다. 자세한 내용은 Amazon EBS 스냅샷 아카이빙 시 고려 사항 및 제한 사항 단원을 참조하십시오.

스냅샷이 하나 이상의 AMI와 연결된 경우 스냅샷을 보관하기 전에 연결된 AMI를 비활성화해야 합니다. 자세한 내용은 AMI 비활성화를 참조하세요.

보관된 스냅샷은 스냅샷 ID, 암호화 상태, AWS Identity and Access Management (IAM) 권한, 소유자 정보 및 리소스 태그를 유지합니다. 그러나 스냅샷이 아카이빙된 후에는 빠른 스냅샷 복원 및 스냅샷 공유가 자동으로 비활성화됩니다.

아카이브가 진행되는 동안 스냅샷을 계속 사용할 수 있습니다. 스냅샷 계층화 상태가 `archival-complete` 상태에 도달하는 즉시 더 이상 스냅샷을 사용할 수 없습니다.

Console

스냅샷 아카이빙

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [스냅샷(Snapshots)]을 선택합니다.
3. 스냅샷 목록에서 아카이빙할 스냅샷을 선택하고 작업(Actions), 스냅샷 아카이빙(Archive snapshot)을 선택합니다.
4. 확인하려면 스냅샷 아카이빙(Archive snapshot)을 선택합니다.

AWS CLI

스냅샷을 아카이브하려면

modify-snapshot-tier AWS CLI 명령을 사용합니다. `--snapshot-id`에서 아카이빙할 스냅샷의 ID를 지정합니다. `--storage-tier`에서 `archive`를 지정합니다.

```
aws ec2 modify-snapshot-tier \
    --snapshot-id snap-0abcdef1234567890 \
    --storage-tier archive
```

다음은 예제 출력입니다. `TieringStartTime` 응답 파라미터는 아카이브 프로세스가 시작된 날짜 및 시간을 UTC 시간 형식(YYYY-MM-DDTHH:MM:SSZ)으로 나타냅니다.

```
{
    "SnapshotId": "snap-0abcdef1234567890",
    "TieringStartTime": "2021-09-15T16:44:37.574Z"
}
```

PowerShell

스냅샷을 아카이브하려면

Edit-EC2SnapshotTier cmdlet을 사용합니다. 다음 예시에서는 지정된 스냅샷을 아카이브합니다.

```
Edit-EC2SnapshotTier `
    -SnapshotId snap-0abcdef1234567890 `
```

```
    -StorageTier "archive"
```

아카이브된 Amazon EBS 스냅샷 복원

아카이빙된 스냅샷을 사용하려면 먼저 표준 계층으로 복원해야 합니다. 복원된 스냅샷은 아카이빙되기 전과 동일한 스냅샷 ID, 암호화 상태, IAM 권한, 소유자 정보 및 리소스 태그를 가집니다. 복원된 후에는 계정의 다른 스냅샷을 사용하는 것과 동일한 방식으로 사용할 수 있습니다. 복원된 스냅샷은 항상 전체 스냅샷입니다.

스냅샷을 복원할 때 영구적으로 또는 임시로 복원하도록 선택할 수 있습니다.

스냅샷을 영구적으로 복원하면 스냅샷이 아카이브 계층에서 표준 계층으로 영구적으로 이동됩니다. 스냅샷은 수동으로 다시 아카이빙하거나 삭제할 때까지 복원된 상태로 유지되며 사용할 준비가 되어 있습니다. 스냅샷을 영구적으로 복원하면 아카이브 계층에서 스냅샷이 제거됩니다.

스냅샷을 임시로 복원하면 지정한 복원 기간 동안 스냅샷이 아카이브 계층에서 표준 계층으로 복사됩니다. 스냅샷은 복원된 상태로 유지되며 복원 기간 동안만 사용할 수 있습니다. 복원 기간 동안 스냅샷의 사본은 아카이브 계층에 남아 있습니다. 기간이 만료되면 스냅샷이 표준 계층에서 자동으로 제거됩니다. 복원 기간 동안 언제든지 복원 기간을 늘리고 줄이거나 복원 유형을 영구로 변경할 수 있습니다. 자세한 내용은 임시로 복원된 Amazon EBS 스냅샷의 복원 기간 수정 단원을 참조하십시오.

비활성화된 AMI와 연결된 스냅샷을 복원하고 해당 AMI를 사용하려는 경우 먼저 연결된 모든 스냅샷을 영구적으로 복원한 다음에 비활성화된 AMI를 다시 활성화해야 사용할 수 있습니다. 연결된 스냅샷이 일시적으로 복원된 경우 AMI를 활성화할 수 없습니다. 다음 명령을 사용하여 AMI와 연결된 모든 스냅샷을 찾을 수 있습니다.

```
aws ec2 describe-images --image-id ami_id \
  --query Images[*].BlockDeviceMappings[*].Ebs[].SnapshotId[]
```

Console

아카이브에서 스냅샷 복원

https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.

1. 탐색 창에서 [스냅샷(Snapshots)]을 선택합니다.
2. 스냅샷 목록에서 복원할 아카이빙된 스냅샷을 선택한 다음 작업(Actions), 아카이브에서 스냅샷 복원(Restore snapshot from archive)을 선택합니다.
3. 수행할 복원의 유형을 지정합니다. 복원 유형(Restore type)에서 다음 중 하나를 수행합니다.

- 스냅샷을 영구적으로 복원하려면 영구(Permanent)를 선택합니다.
- 스냅샷을 임시로 복원하려면 임시(Temporary)를 선택한 다음 임시 복원 기간(Temporary restore period)에서 스냅샷을 복원할 일수를 입력합니다.

4. 확인하려면 스냅샷 복원(Restore snapshot)을 선택합니다.

AWS CLI

아카이빙된 스냅샷 영구 복원

restore-snapshot-tier AWS CLI 명령을 `--permanent-restore` 옵션과 함께 사용합니다. `--snapshot-id`에 대해 복원할 스냅샷의 ID를 지정합니다.

```
aws ec2 restore-snapshot-tier \
    --snapshot-id snap-0abcdef1234567890 \
    --permanent-restore
```

다음은 예제 출력입니다.

```
{
    "SnapshotId": "snap-0abcdef1234567890",
    "IsPermanentRestore": true
}
```

아카이빙된 스냅샷 임시

restore-snapshot-tier AWS CLI 명령을 사용합니다. `--permanent-restore` 옵션을 생략합니다. `--temporary-restore-days`에서 스냅샷을 복원할 일수를 정합니다. 유효한 범위는 1~180일입니다. 값을 지정하지 않으면 기본값은 1일입니다.

다음 예제는 복원 기간 5일 동안 지정된 스냅샷을 임시로 복원합니다.

```
aws ec2 restore-snapshot-tier \
    --snapshot-id snap-0abcdef1234567890 \
    --temporary-restore-days 5
```

다음은 예제 출력입니다.

```
{
    "SnapshotId": "snap-0abcdef1234567890",
```

```
    "RestoreDuration": 5,
    "IsPermanentRestore": false
}
```

PowerShell

아카이빙된 스냅샷 영구 복원

Restore-EC2SnapshotTier cmdlet을 사용합니다.

```
Restore-EC2SnapshotTier `
    -SnapshotId snap-0abcdef1234567890 `
    -PermanentRestore $true
```

아카이빙된 스냅샷 임시

Restore-EC2SnapshotTier cmdlet을 사용합니다.

```
Restore-EC2SnapshotTier `
    -SnapshotId snap-0abcdef1234567890 `
    -TemporaryRestoreDays 5
```

임시로 복원된 Amazon EBS 스냅샷의 복원 기간 수정

스냅샷을 임시로 복원할 때 계정에서 스냅샷이 복원된 상태로 유지되는 일수를 지정해야 합니다. 복원 기간이 만료되면 스냅샷이 표준 계층에서 자동으로 제거됩니다.

임시로 복원된 스냅샷의 복원 기간은 언제든지 변경할 수 있습니다.

복원 기간을 늘리거나 줄이도록 선택하거나 복원 유형을 임시에서 영구로 변경할 수 있습니다.

복원 기간을 변경하면 새 복원 기간이 현재 날짜부터 유효합니다. 예를 들어 새 복원 기간을 5일로 지정하면 스냅샷은 현재 날짜로부터 5일 동안 복원된 상태로 유지됩니다.

> **Note**
>
> 복원 기간을 1일로 설정하여 임시 복원을 조기에 종료할 수 있습니다.

복원 유형을 임시에서 영구로 변경하면 스냅샷 사본이 아카이브 계층에서 삭제되고 스냅샷은 수동으로 다시 아카이빙하거나 삭제할 때까지 계정에서 계속 사용할 수 있습니다.

Console

복원 기간 또는 복원 유형 수정

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [스냅샷(Snapshots)]을 선택합니다.
3. 스냅샷 목록에서 이전에 임시로 복원한 스냅샷을 선택한 다음 작업(Actions), 아카이브에서 스냅샷 복원(Restore snapshot from archive)을 선택합니다.
4. 복원 유형(Restore type)에서 다음 중 하나를 수행합니다.
   - 복원 유형을 임시에서 영구로 변경하려면 영구(Permanent)를 선택합니다.
   - 복원 기간을 늘리거나 줄이려면 임시(Temporary)를 유지한 다음 임시 복원 기간(Temporary restore period)에 새 복원 기간을 일 단위로 입력합니다.
5. 확인하려면 스냅샷 복원(Restore snapshot)을 선택합니다.

AWS CLI

복원 기간 수정 또는 복원 유형 변경

restore-snapshot-tier 명령을 사용합니다. `--snapshot-id`에 대해 이전에 임시로 복원한 스냅샷의 ID를 지정합니다. 복원 유형을 임시에서 영구로 변경하려면 `--permanent-restore`를 지정하고 `--temporary-restore-days`를 생략합니다. 복원 기간을 늘리거나 줄이려면 `--permanent-restore`를 생략하고 `--temporary-restore-days`에 대해 새 복원 기간을 일 단위로 지정합니다.

예: 복원 기간 증가 또는 감소

다음 예제는 지정한 스냅샷의 복원 기간을 10일로 변경합니다.

```
aws ec2 restore-snapshot-tier \
    --snapshot-id snap-0abcdef1234567890 \
    --temporary-restore-days 10
```

다음은 예제 출력입니다.

```
{
    "SnapshotId": "snap-0abcdef1234567890",
    "RestoreDuration": 10,
    "IsPermanentRestore": false
```

```
}
```

예: 복원 유형을 영구로 변경

다음 예제는 지정된 스냅샷의 복원 유형을 임시에서 영구로 변경합니다.

```
aws ec2 restore-snapshot-tier \
    --snapshot-id snap-0abcdef1234567890 \
    --permanent-restore
```

다음은 예제 출력입니다.

```
{
    "SnapshotId": "snap-0abcdef1234567890",
    "IsPermanentRestore": true
}
```

PowerShell

복원 기간 수정 또는 복원 유형 변경

Restore-EC2SnapshotTier cmdlet을 사용합니다. -SnapshotId에 대해 이전에 임시로 복원한 스냅샷의 ID를 지정합니다. 복원 유형을 임시에서 영구로 변경하려면 -PermanentRestore를 지정하고 -TemporaryRestoreDays를 생략합니다. 복원 기간을 늘리거나 줄이려면 -PermanentRestore를 생략하고 -TemporaryRestoreDays에 대해 새 복원 기간을 일 단위로 지정합니다.

예: 복원 기간 증가 또는 감소

다음 예제는 지정한 스냅샷의 복원 기간을 10일로 변경합니다.

```
Restore-EC2SnapshotTier `
    -SnapshotId snap-0abcdef1234567890 `
    -TemporaryRestoreDays 10
```

예: 복원 유형을 영구로 변경

다음 예제는 지정된 스냅샷의 복원 유형을 임시에서 영구로 변경합니다.

```
Restore-EC2SnapshotTier `
    -SnapshotId snap-0abcdef1234567890 `
    -PermanentRestore $true
```

아카이브된 Amazon EBS 스냅샷 보기

Console

스냅샷에 대한 스토리지 계층 정보 보기

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [스냅샷(Snapshots)]을 선택합니다.
3. 스냅샷 목록에서 스냅샷을 선택하고 스토리지 계층(Storage tier) 탭을 선택합니다.

   이 탭은 다음 정보를 제공합니다.

   - 마지막 계층 변경이 시작된 일자(Last tier change started on) - 마지막 아카이브 또는 복원이 시작된 날짜 및 시간입니다.
   - 계층 변경 진행률(Tier change progress) - 백분율로 나타낸 마지막 아카이브 또는 복원 작업의 진행률입니다.
   - 스토리지 계층(Storage tier) - 스냅샷의 스토리지 계층입니다. 임시로 복원된 스냅샷을 포함하여 아카이빙된 스냅샷의 경우 항상 `archive`이고 표준 계층에 저장된 스냅샷의 경우 `standard`입니다.
   - 상태 계층화(Tiering status) - 마지막 아카이브 또는 복원 작업의 상태입니다.
   - 아카이빙 완료 일자(Archive completed on) - 아카이빙이 완료된 날짜 및 시간입니다.
   - 임시 복원 만료 일자(Temporary restore expires on) - 임시로 복원된 스냅샷이 만료되도록 설정된 날짜 및 시간입니다.

AWS CLI

아카이빙된 스냅샷에 대한 아카이브 정보 보기

describe-snapshot-tier-status 명령을 사용합니다. `snapshot-id` 필터를 지정하고 필터 값에 대해 스냅샷 ID를 지정합니다. 또는 아카이빙된 모든 스냅샷을 보려면 필터를 생략합니다.

```
--filters "Name=snapshot-id, Values=snapshot_id"
```

출력에는 다음 응답 파라미터가 포함됩니다.

- `Status` - 스냅샷의 상태입니다. 아카이빙된 스냅샷의 경우 항상 `completed`입니다. `completed` 상태의 스냅샷만 아카이빙할 수 있습니다.

- `LastTieringStartTime` - 아카이브 프로세스가 시작된 날짜 및 시간으로 UTC 시간 형식(YYYY-MM-DDTHH:MM:SSZ)입니다.
- `LastTieringOperationState` - 아카이브 프로세스의 현재 상태입니다. 가능한 상태: `archival-in-progress` | `archival-completed` | `archival-failed` | `permanent-restore-in-progress` | `permanent-restore-completed` | `permanent-restore-failed` | `temporary-restore-in-progress` | `temporary-restore-completed` | `temporary-restore-failed`
- `LastTieringProgress` - 스냅샷 아키이빙 프로세스의 진행률(%)입니다.
- `StorageTier` - 스냅샷의 스토리지 계층입니다. 임시로 복원된 스냅샷을 포함하여 아카이빙된 스냅샷의 경우 항상 `archive`이고 표준 계층에 저장된 스냅샷의 경우 `standard`입니다.
- `ArchivalCompleteTime` - 아카이브 프로세스가 완료된 날짜 및 시간으로 UTC 시간 형식(YYYY-MM-DDTHH:MM:SSZ)입니다.

예제: 스냅샷 설명

다음 예제는 지정된 스냅샷에 대한 정보를 표시합니다.

```
aws ec2 describe-snapshot-tier-status \
    --filters "Name=snapshot-id, Values=snap-0abcdef1234567890"
```

다음은 예제 출력입니다.

```
{
    "SnapshotTierStatuses": [
        {
            "Status": "completed",
            "ArchivalCompleteTime": "2021-09-15T17:33:16.147Z",
            "LastTieringProgress": 100,
            "Tags": [],
            "VolumeId": "vol-01234567890abcdef",
            "LastTieringOperationState": "archival-completed",
            "StorageTier": "archive",
            "OwnerId": "123456789012",
            "SnapshotId": "snap-0abcdef1234567890",
            "LastTieringStartTime": "2021-09-15T16:44:37.574Z"
        }
    ]
}
```

아카이빙된 표준 계층 스냅샷 보기

describe-snapshots 명령을 사용합니다. --snapshot-ids에 대해 스냅샷의 ID를 지정합니다.

```
aws ec2 describe-snapshots --snapshot-ids snap-0abcdef1234567890
```

다음은 예제 출력입니다. StorageTier 응답 파라미터는 스냅샷이 현재 아카이빙되어 있는지 여부를 나타냅니다. archive는 스냅샷이 현재 아카이빙되었으며 아카이브 계층에 저장되어 있음을 나타내고, standard는 스냅샷이 현재 아카이빙되지 않았으며 표준 계층에 저장되어 있음을 나타냅니다.

다음 예제 출력에서는 Snap A만 아카이빙되었으며, Snap B와 Snap C는 아카이빙되지 않았습니다.

또한 RestoreExpiryTime 응답 파라미터는 아카이브에서 임시로 복원된 스냅샷에 대해서만 반환됩니다. 임시로 복원된 스냅샷이 표준 계층에서 자동으로 제거되는 시기를 나타냅니다. 영구적으로 복원된 스냅샷에 대해서는 반환되지 않습니다.

다음 예제 출력에서 Snap C는 임시로 복원되며 2021-09-19T21:00:00.000Z(2021년 9월 19일 21:00 UTC)에 표준 계층에서 자동으로 제거됩니다.

```
{
    "Snapshots": [
        {
            "Description": "Snap A",
            "Encrypted": false,
            "VolumeId": "vol-01234567890aaaaaa",
            "State": "completed",
            "VolumeSize": 8,
            "StartTime": "2021-09-07T21:00:00.000Z",
            "Progress": "100%",
            "OwnerId": "123456789012",
            "SnapshotId": "snap-01234567890aaaaaa",
            "StorageTier": "archive",
            "Tags": []
        },
        {
            "Description": "Snap B",
            "Encrypted": false,
            "VolumeId": "vol-09876543210bbbbbb",
            "State": "completed",
            "VolumeSize": 10,
```

```
            "StartTime": "2021-09-14T21:00:00.000Z",
            "Progress": "100%",
            "OwnerId": "123456789012",
            "SnapshotId": "snap-09876543210bbbbbb",
            "StorageTier": "standard",
            "RestoreExpiryTime": "2019-09-19T21:00:00.000Z",
            "Tags": []
        },
        {
            "Description": "Snap C",
            "Encrypted": false,
            "VolumeId": "vol-054321543210cccccc",
            "State": "completed",
            "VolumeSize": 12,
            "StartTime": "2021-08-01T21:00:00.000Z",
            "Progress": "100%",
            "OwnerId": "123456789012",
            "SnapshotId": "snap-054321543210cccccc",
            "StorageTier": "standard",
            "Tags": []
        }
    ]
}
```

아카이브 계층 또는 표준 계층에 저장된 스냅샷만 보기

describe-snapshots 명령을 사용합니다. 필터 이름 `storage-tier`와 함께 `--filter` 옵션을 포함합니다. 필터 값에 `archive` 또는 `standard`를 지정합니다. 다음 예제는 아카이빙된 스냅샷만 표시합니다.

```
aws ec2 describe-snapshots --filters "Name=storage-tier,Values=archive"
```

PowerShell

아카이빙된 스냅샷에 대한 아카이브 정보 보기

Get-EC2SnapshotTierStatus cmdlet을 사용합니다.

```
Get-EC2SnapshotTierStatus `
    -Filter @{Name="snapshot-id"; Values="snap-0abcdef1234567890"}
```

아카이빙된 표준 계층 스냅샷 보기

Get-EC2Snapshot cmdlet을 사용합니다.

```
Get-EC2Snapshot `
    -SnapshotId snap-0abcdef1234567890
```

아카이브 계층 또는 표준 계층에 저장된 스냅샷만 보기

Get-EC2Snapshot cmdlet을 사용합니다. 다음 예제는 아카이빙된 스냅샷만 표시합니다.

```
Get-EC2Snapshot `
    -Filter @{Name="storage-tier"; Values="archive"}
```

CloudWatch Events를 사용하여 Amazon EBS 스냅샷 아카이빙 모니터링

Amazon EBS는 스냅샷 아카이빙 작업과 관련된 이벤트를 내보냅니다. AWS Lambda 및 Amazon CloudWatch Events를 사용하여 이벤트 알림을 프로그래밍 방식으로 처리할 수 있습니다. 이벤트는 최선의 작업을 기반으로 발생됩니다. 자세한 내용은 Amazon EventBridge 사용 설명서를 참조하세요.

다음과 같은 이벤트를 사용할 수 있습니다.

- `archiveSnapshot` - 스냅샷 아카이빙 작업이 성공하거나 실패할 때 내보내집니다.

  다음은 스냅샷 아카이브 작업이 성공할 때 내보내지는 이벤트의 예입니다.

```
{
   "version": "0",
   "id": "01234567-0123-0123-0123-012345678901",
   "detail-type": "EBS Snapshot Notification",
   "source": "aws.ec2",
   "account": "123456789012",
   "time": "2021-05-25T13:12:22Z",
   "region": "us-east-1",
   "resources": [
     "arn:aws:ec2:us-east-1::snapshot/snap-01234567890abcdef"
   ],
   "detail": {
     "event": "archiveSnapshot",
     "result": "succeeded",
     "cause": "",
     "request-id": "123456789",
     "snapshot_id": "arn:aws:ec2:us-east-1::snapshot/snap-01234567890abcdef",
```

```
    "startTime": "2021-05-25T13:12:22Z",
    "endTime": "2021-05-45T15:30:00Z",
    "recycleBinExitTime": "2021-10-45T15:30:00Z"
  }
```

다음은 스냅샷 아카이브 작업이 실패할 때 내보내지는 이벤트의 예입니다.

```
{
   "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "EBS Snapshot Notification",
  "source": "aws.ec2",
  "account": "123456789012",
  "time": "2021-05-25T13:12:22Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:ec2:us-east-1::snapshot/snap-01234567890abcdef"
  ],
  "detail": {
    "event": "archiveSnapshot",
    "result": "failed",
    "cause": "Source snapshot ID is not valid",
    "request-id": "1234567890",
    "snapshot_id": "arn:aws:ec2:us-east-1::snapshot/snap-01234567890abcdef",
    "startTime": "2021-05-25T13:12:22Z",
    "endTime": "2021-05-45T15:30:00Z",
    "recycleBinExitTime": "2021-10-45T15:30:00Z"
  }
}
```

- `permanentRestoreSnapshot` - 영구 복원 작업이 성공하거나 실패할 때 내보내집니다.

  다음은 영구적 복원 작업이 성공할 때 내보내지는 이벤트의 예입니다.

```
{
   "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "EBS Snapshot Notification",
  "source": "aws.ec2",
  "account": "123456789012",
  "time": "2021-05-25T13:12:22Z",
  "region": "us-east-1",
  "resources": [
```

```
    "arn:aws:ec2:us-east-1::snapshot/snap-01234567890abcdef"
  ],
  "detail": {
    "event": "permanentRestoreSnapshot",
    "result": "succeeded",
    "cause": "",
    "request-id": "1234567890",
    "snapshot_id": "arn:aws:ec2:us-east-1::snapshot/snap-01234567890abcdef",
    "startTime": "2021-05-25T13:12:22Z",
    "endTime": "2021-10-45T15:30:00Z"
  }
}
```

다음은 영구적 복원 작업이 실패할 때 내보내지는 이벤트의 예입니다.

```
{
   "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "EBS Snapshot Notification",
  "source": "aws.ec2",
  "account": "123456789012",
  "time": "2021-05-25T13:12:22Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:ec2:us-east-1::snapshot/snap-01234567890abcdef"
  ],
  "detail": {
    "event": "permanentRestoreSnapshot",
    "result": "failed",
    "cause": "Source snapshot ID is not valid",
    "request-id": "1234567890",
    "snapshot_id": "arn:aws:ec2:us-east-1::snapshot/snap-01234567890abcdef",
    "startTime": "2021-05-25T13:12:22Z",
    "endTime": "2021-05-45T15:30:00Z",
    "recycleBinExitTime": "2021-10-45T15:30:00Z"
  }
}
```

- `temporaryRestoreSnapshot` - 임시 복원 작업이 성공하거나 실패할 때 내보내집니다.

  다음은 임시 복원 작업이 성공할 때 내보내지는 이벤트의 예입니다.

```
{
```

```
   "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "EBS Snapshot Notification",
  "source": "aws.ec2",
  "account": "123456789012",
  "time": "2021-05-25T13:12:22Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:ec2:us-east-1::snapshot/snap-01234567890abcdef"
  ],
  "detail": {
    "event": "temporaryRestoreSnapshot",
    "result": "succeeded",
    "cause": "",
    "request-id": "1234567890",
    "snapshot_id": "arn:aws:ec2:us-us-east-1::snapshot/snap-01234567890abcdef",
    "startTime": "2021-05-25T13:12:22Z",
    "endTime": "2021-05-45T15:30:00Z",
    "restoreExpiryTime": "2021-06-45T15:30:00Z",
    "recycleBinExitTime": "2021-10-45T15:30:00Z"
  }
}
```

다음은 임시 복원 작업이 실패할 때 내보내지는 이벤트의 예입니다.

```
{
   "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "EBS Snapshot Notification",
  "source": "aws.ec2",
  "account": "123456789012",
  "time": "2021-05-25T13:12:22Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:ec2:us-east-1::snapshot/snap-01234567890abcdef"
  ],
  "detail": {
    "event": "temporaryRestoreSnapshot",
    "result": "failed",
    "cause": "Source snapshot ID is not valid",
    "request-id": "1234567890",
    "snapshot_id": "arn:aws:ec2:us-east-1::snapshot/snap-01234567890abcdef",
    "startTime": "2021-05-25T13:12:22Z",
```

```
    "endTime": "2021-05-45T15:30:00Z",
    "recycleBinExitTime": "2021-10-45T15:30:00Z"
  }
}
```

- `restoreExpiry` - 임시로 복원된 스냅샷의 복원 기간이 만료될 때 내보내집니다.

  다음은 예입니다.

```
{
   "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "EBS Snapshot Notification",
  "source": "aws.ec2",
  "account": "123456789012",
  "time": "2021-05-25T13:12:22Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:ec2:us-east-1::snapshot/snap-01234567890abcdef"
  ],
  "detail": {
    "event": "restoryExpiry",
    "result": "succeeded",
    "cause": "",
    "request-id": "1234567890",
    "snapshot_id": "arn:aws:ec2:us-east-1::snapshot/snap-01234567890abcdef",
    "startTime": "2021-05-25T13:12:22Z",
    "endTime": "2021-05-45T15:30:00Z",
    "recycleBinExitTime": "2021-10-45T15:30:00Z"
  }
}
```

Amazon EBS 스냅샷 삭제

볼륨의 Amazon EBS 스냅샷이 더 이상 필요하지 않으면 삭제할 수 있습니다. 스냅샷을 삭제해도 볼륨에는 영향을 주지 않습니다. 볼륨을 삭제해도 해당 볼륨에서 만들어진 스냅샷에는 아무런 영향을 미치지 않습니다.

주제

- 스냅샷 삭제 시 고려 사항
- 증분 스냅샷 삭제 작동 방식

- 스냅샷 삭제
- 다중 볼륨 스냅샷 삭제

스냅샷 삭제 시 고려 사항

다음은 스냅샷을 삭제할 때 고려할 사항입니다.

- 등록된 AMI에서 사용된 EBS 볼륨의 루트 디바이스에 대한 스냅샷을 삭제할 수 없습니다. 이 고려 사항은 등록된 AMI가 더 이상 사용되지 않거나 비활성화된 경우에도 적용됩니다. 스냅샷을 삭제하기 전 우선 AMI를 등록해야 합니다. 자세한 내용은 AMI 등록 취소를 참조하세요.
- Amazon EC2를 사용하여 AWS Backup 서비스에서 관리하는 스냅샷은 삭제할 수 없습니다. 대신 AWS Backup 를 사용하여 백업 볼트에서 해당 복구 시점을 삭제합니다. 자세한 내용은AWS Backup 개발자 안내서의 백업 삭제를 참조하세요.
- 스냅샷을 수동으로 생성, 보존 및 삭제하거나 Amazon Data Lifecycle Manager를 사용하여 스냅샷을 관리할 수 있습니다. 자세한 내용은 Amazon Data Lifecycle Manager 섹션을 참조하세요.
- 진행 중인 스냅샷을 삭제할 수는 있지만, 삭제가 적용되려면 해당 스냅샷이 완전해야 합니다. 이 작업은 시간이 오래 걸릴 수 있습니다. 동시 스냅샷 제한 상태에서 스냅샷을 추가로 만들려고 하면 `ConcurrentSnapshotLimitExceeded` 오류를 받을 수 있습니다. 자세한 내용은 Amazon Web Services 일반 참조에서 Amazon EBS의 Service Quotas을 참조하세요.
- 휴지통 보존 규칙과 일치하는 스냅샷을 삭제하면 스냅샷이 즉시 삭제되는 대신 휴지통에 보관됩니다. 자세한 내용은 휴지통을 참조하세요.
- EBS 지원 AMI와 연결된 스냅샷은 삭제할 수 있습니다. 자세한 내용은 AMI 비활성화를 참조하세요.
- 공유된 스냅샷은 삭제할 수 없습니다.
- 소유한 공유 스냅샷을 삭제하면 해당 스냅샷이 공유된 모든 계정이 스냅샷에 대한 액세스 권한을 상실합니다.

증분 스냅샷 삭제 작동 방식

정기적으로 볼륨의 스냅샷을 만드는 경우, 스냅샷은 증분식으로 늘어납니다. 다시 말해 새 스냅샷에는 최신 스냅샷 이후로 변경된 디바이스 블록만 저장됩니다. 스냅샷은 증분식으로 저장되지만 스냅샷 삭제 프로세스는 볼륨을 생성하기 위해 가장 최근의 스냅샷만을 유지할 수 있도록 설계됩니다.

이전 스냅샷 또는 일련의 스냅샷에 보관된 볼륨에 데이터가 있는 경우 나중에 데이터가 볼륨에서 삭제되어도 해당 데이터는 이전 스냅샷의 고유한 데이터로 간주됩니다. 고유 데이터는 고유한 데이터를 참조하는 모든 스냅샷을 삭제하는 경우에만 스냅샷 시퀀스에서 삭제됩니다.

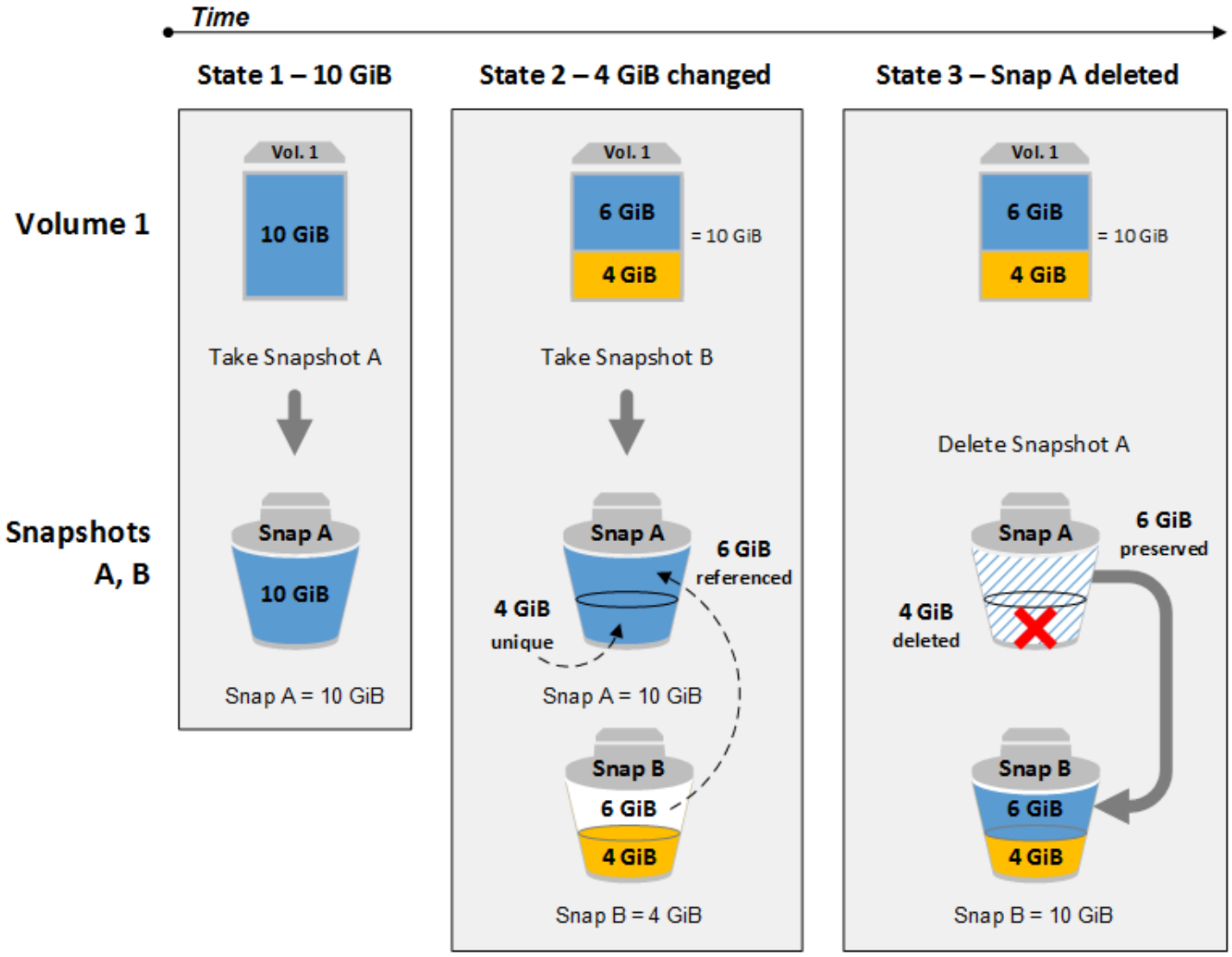
스냅샷을 삭제하면 해당 스냅샷에서 참조하는 데이터만 제거됩니다. 고유 데이터는 해당 데이터를 참조하는 모든 스냅샷이 삭제되는 경우에만 삭제됩니다. 볼륨의 이전 스냅샷을 삭제해도 해당 볼륨의 이후 스냅샷에서 볼륨을 생성하는 기능에는 영향을 주지 않습니다.

스냅샷을 삭제해도 조직의 데이터 스토리지 비용이 줄어들지 않을 수 있습니다. 다른 스냅샷은 해당 스냅샷의 데이터를 참조할 수 있으며, 참조된 데이터는 항상 보존됩니다. 이후의 스냅샷에서 사용 중인 데이터가 포함된 스냅샷을 삭제하는 경우, 참조된 데이터와 관련된 비용이 이후의 스냅샷에 할당됩니다. 스냅샷이 데이터를 저장하는 방법에 대한 자세한 내용은 Amazon EBS 스냅샷 작동 방식 및 다음 예를 참조하세요.

다음 다이어그램에서 볼륨 1은 세 가지 시점에 표시됩니다. 스냅샷이 첫 두 상태를 각각 캡처했으며, 세 번째에서는 스냅샷이 삭제되었습니다.

- 상태 1의 볼륨에는 10GiB의 데이터가 있습니다. 스냅 A는 이 볼륨의 첫 번째 스냅샷이므로 10GiB 데이터 전체를 복사해야 합니다. 이 상태에서는 10GiB의 스냅샷 데이터를 저장하는 데 대한 요금이 부과됩니다.
- 상태 2의 볼륨에는 여전히 10GiB의 데이터가 있지만 4GiB가 변경되었습니다. 스냅 B는 스냅 A를 생성한 후 변경된 4GiB만 저장하고 변경되지 않은 6GiB의 데이터는 스냅 A에 이미 저장된 데이터를 참조합니다.이 상태에서는 스냅샷 데이터 14GiB(스냅 A의 10Gib + 스냅 B의 4GiB)를 저장하는 데 대한 요금이 부과됩니다.
- 상태 3에서는 볼륨이 변경되지는 않지만 스냅 A가 삭제됩니다. 스냅 A의 변경되지 않은 데이터 6GiB는 여전히 스냅 B에서 참조하고 있으므로 해당 데이터는 보존되고 스냅 B와 연결됩니다. 스냅 A의 고유 데이터 4GiB는 더 이상 다른 스냅샷에서 참조하지 않으므로 삭제됩니다. 이 상태에서는 10GiB의 스냅샷 데이터(스냅 A에 보존된 6GiB 데이터 + 스냅 B의 4GiB 데이터)를 저장하는 데 대한 요금이 부과됩니다.

다른 스냅샷에서 참조된 데이터가 포함된 스냅샷 삭제



스냅샷 삭제

Console

스냅샷 삭제

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [Snapshots]를 선택합니다.
3. 스냅샷을 선택합니다.
4. 작업(Action), 스냅샷 삭제(Delete snapshot)를 선택합니다.
5. 확인 메시지가 나타나면 **delete**을 입력한 다음 삭제를 선택합니다.

AWS CLI

스냅샷 삭제

delete-snapshot 명령을 사용합니다.

```
aws ec2 delete-snapshot --snapshot-id snap-0abcdef1234567890
```

PowerShell

스냅샷 삭제

Remove-EC2Snapshot cmdlet을 사용합니다.

```
Remove-EC2Snapshot -SnapshotId snap-0abcdef1234567890
```

문제 해결 도움말

현재 AMI에서 스냅샷을 사용 중이라는 `Failed to delete snapshot` 오류가 표시되면 먼저 연결된 AMI를 등록 취소해야 스냅샷을 삭제할 수 있습니다. AMI에 연결된 스냅샷은 삭제할 수 없습니다.
콘솔을 사용 중이고 연결된 AMI가 비활성화된 경우 비활성화된 AMI를 보려면 AMI 화면에서 비활성화된 이미지 필터를 선택해야 합니다.

다중 볼륨 스냅샷 삭제

다중 볼륨 스냅샷을 삭제하려면 스냅샷을 생성할 때 세트에 적용한 태그를 사용하여 다중 볼륨 스냅샷 세트에 대한 모든 스냅샷을 검색합니다. 그런 다음 스냅샷을 개별적으로 삭제합니다.

다중 볼륨 스냅샷 세트의 개별 스냅샷을 삭제할 수 있습니다. `pending state` 상태에 있는 스냅샷을 삭제하는 경우 해당 스냅샷만 삭제됩니다. 다중 볼륨 스냅샷 세트의 다른 스냅샷은 여전히 성공적으로 완료됩니다.

Amazon EBS 빠른 스냅샷 복원

Amazon EBS 빠른 스냅샷 복원(FSR)을 사용하면 생성 시 완전히 초기화된 스냅샷에서 볼륨을 생성할 수 있습니다. 이렇게 하면 처음 액세스할 때 블록에 대한 I/O 작업 지연 시간이 없어집니다. 빠른 스냅샷 복원을 사용하여 생성된 볼륨은 프로비저닝된 모든 성능을 즉시 제공합니다.

시작하려면 특정 가용 영역의 특정 스냅샷에 대해 빠른 스냅샷 복원을 활성화합니다. 각 스냅샷 및 가용 영역 페어는 하나의 빠른 스냅샷 복원을 나타냅니다. 활성화된 가용 영역 중 하나에 있는 이러한 스냅샷 중 하나에서 볼륨을 생성하면 빠른 스냅샷 복원을 사용하여 볼륨이 복원됩니다.

각 스냅샷에 대해 빠른 스냅샷 복원을 명시적으로 활성화해야 합니다. 예를 들어 빠른 스냅샷 복원 지원 스냅샷에서 복원된 볼륨으로 새 스냅샷을 생성하는 경우 새 스냅샷에 대해 빠른 스냅샷 복원이 자동으로 활성화되지 않습니다. 빠른 스냅샷 복원이 활성화된 스냅샷을 복사하는 경우 스냅샷 복사본은 빠른 스냅샷 복원이 자동으로 활성화되지 않습니다.

빠른 스냅샷 복원의 전체 성능 이점으로 복원할 수 있는 볼륨 수는 스냅샷에 대한 볼륨 생성 크레딧에 따라 결정됩니다. 자세한 내용은 Amazon EBS 빠른 스냅샷 복원 볼륨 생성 크레딧 섹션을 참조하세요.

본인이 소유한 스냅샷과 본인에게 공유된 퍼블릭 및 프라이빗 스냅샷에 대해 빠른 스냅샷 복원을 활성화할 수 있습니다.

내용

- 고려 사항
- 요금 및 결제
- Amazon EBS 빠른 스냅샷 복원 볼륨 생성 크레딧
- Amazon EBS 스냅샷에 대한 빠른 스냅샷 복원 구성
- Amazon EBS 스냅샷에 대한 빠른 스냅샷 복원 상태 확인
- 빠른 스냅샷 복원을 사용하여 복원된 Amazon EBS 볼륨 보기

고려 사항

- 빠른 스냅샷 복원은 AWS Outposts로컬 영역 및 Wavelength 영역에서 지원되지 않습니다.
- 크기가 16TiB 이하인 스냅샷에서 빠른 스냅샷 복원을 활성화할 수 있습니다.
- 최대 64,000 IOPS 및 1,000MiB/s 처리량의 성능으로 프로비저닝된 볼륨은 빠른 스냅샷 복원의 성능상 이점을 최대한 활용합니다. 64,000 IOPS 또는 1,000MiB/s 처리량 이상의 성능으로 프로비저닝된 볼륨의 경우 전체 성능을 발휘하도록 볼륨을 초기화하는 것이 좋습니다.
- 리전별로 빠른 스냅샷 복원을 위해 최대 5개의 스냅샷을 활성화할 수 있습니다. 할당량은 본인이 소유한 스냅샷과 본인에게 공유된 스냅샷에 적용됩니다. 본인에게 공유된 스냅샷에 대해 빠른 스냅샷 복원을 활성화하면 빠른 스냅샷 복원 할당량에 포함됩니다. 스냅샷 소유자의 빠른 스냅샷 복원 할당량에 포함되지 않습니다.
- Amazon EBS는 스냅샷에 대한 빠른 스냅샷 복원 상태가 변경되면 Amazon CloudWatch Events를 보냅니다. 자세한 내용은 EBS 빠른 스냅샷 복원 이벤트 섹션을 참조하세요.

요금 및 결제

특정 가용 영역의 스냅샷에 대해 빠른 스냅샷 복원이 활성화된 1분마다 요금이 청구됩니다. 요금은 최소 1시간으로 비례 청구됩니다.

예를 들어 한 달(30일) 동안 US-East-1a에서 한 스냅샷에 대해 빠른 스냅샷 복원을 사용하면 540 USD(스냅샷 1개 x AZ 1개 x 720시간 x 시간당 $0.75)가 청구됩니다. 동일한 기간 동안 us-east-1a, us-east-1b, us-east-1c에서 두 스냅샷에 대해 빠른 스냅샷 복원을 활성화하면 3240 USD(스냅샷 2개 x AZ 3개 x 720시간 x 시간당 $0.75)가 청구됩니다.

본인에게 공유된 퍼블릭 또는 프라이빗 스냅샷에 대해 빠른 스냅샷 복원을 활성화하는 경우 계정에 요금이 청구되며 스냅샷 소유자에게는 요금이 청구되지 않습니다. 본인에게 공유된 스냅샷이 스냅샷 소유자에 의해 삭제되거나 공유 해제되면 계정의 스냅샷에 대해 빠른 스냅샷 복원이 비활성화되고 청구가 중지됩니다.

자세한 내용은 Amazon EBS 요금을 참조하세요.

Amazon EBS 빠른 스냅샷 복원 볼륨 생성 크레딧

빠른 스냅샷 복원의 최대 성능 이점을 얻는 볼륨 수는 스냅샷에 대한 볼륨 생성 크레딧에 의해 결정됩니다. 가용 영역당 스냅샷별로 하나의 크레딧 버킷이 있습니다. 빠른 스냅샷 복원이 활성화된 스냅샷에서 생성하는 각 볼륨은 크레딧 버킷에서 하나의 크레딧을 사용합니다. 스냅샷에서 초기화된 볼륨을 생성하려면 버킷에 하나 이상의 크레딧이 있어야 합니다. 볼륨을 생성하지만 버킷에 크레딧이 하나 미만인 경우 빠른 스냅샷 복원의 이점 없이 볼륨이 생성됩니다.

본인에게 공유된 스냅샷에 대해 빠른 스냅샷 복원을 활성화하면 계정의 공유 스냅샷에 대해 별도의 크레딧 버킷이 생성됩니다. 공유 스냅샷에서 볼륨을 생성하는 경우 크레딧은 크레딧 버킷에서 소비되고 스냅샷 소유자의 크레딧 버킷에서 소비되지 않습니다.

크레딧 버킷 크기와 리필 속도는 스냅샷 데이터의 크기가 아닌 스냅샷의 크기(즉, 소스 볼륨의 크기)를 기반으로 합니다. 예를 들어 150GiB의 데이터가 있는 200GiB 볼륨에서 스냅샷을 생성하고 빠른 스냅샷 복원을 활성화하는 경우 크레딧 버킷 크기와 리필 속도는 200GiB를 기반으로 합니다.

스냅샷에 대해 빠른 스냅샷 복원을 활성화하면 크레딧 버킷이 0크레딧으로 시작하고 최대 크레딧 용량에 도달할 때까지 설정된 속도로 채워집니다. 또한 크레딧을 사용하면 최대 크레딧 용량에 도달할 때까지 시간 경과에 따라 크레딧 버킷이 다시 채워집니다.

크레딧 버킷의 채우기 속도는 다음과 같이 계산됩니다.

```
MIN (10, (1024 ÷ snapshot_size_gib))
```

그리고 크레딧 버킷의 크기는 다음과 같이 계산됩니다.

```
MAX (1, MIN (10, (1024 ÷ snapshot_size_gib)))
```

예를 들어 크기가 128 GiB인 스냅샷에 대해 빠른 스냅샷 복원을 활성화하는 경우 채우기 비율은 분당 0.1333크레딧입니다.

```
MIN (10, (1024 ÷ 128))
 = MIN (10, 8)
 = 8 credits per hour
 = 0.1333 credits per minute
```

그리고 크레딧 버킷의 최대 크기는 8크레딧입니다.

```
MAX (1, MIN (10, (1024 ÷ 128)))
 = MAX (1, MIN (10, 8))
 = MAX (1, 8)
 = 8 credits
```

이 예에서 빠른 스냅샷 복원을 활성화하면 크레딧 버킷은 0크레딧으로 시작합니다. 8분이 지나면 크레딧 버킷은 하나의 초기화된 볼륨을 생성하기에 충분한 크레딧을 갖게 됩니다(0.1333 credits × 8 minutes = 1.066 credits). 크레딧 버킷이 가득 차면 8개의 초기화된 볼륨을 동시에 생성할 수 있습니다(8크레딧). 버킷이 최대 용량 미만이면 분당 0.1333크레딧으로 다시 채워집니다.

CloudWatch 지표를 사용하여 크레딧 버킷의 크기와 각 버킷에서 사용할 수 있는 크레딧의 수를 모니터링할 수 있습니다. 자세한 내용은 빠른 스냅샷 복원 관련 지표 섹션을 참조하세요.

빠른 스냅샷 복원이 활성화된 스냅샷에서 볼륨을 생성한 후, describe-volumes를 사용하여 볼륨을 설명하고 출력에서 fastRestored 필드를 확인하여 볼륨이 빠른 스냅샷 복원을 사용하여 초기화된 볼륨으로 생성되었는지 여부를 확인할 수 있습니다.

Amazon EBS 스냅샷에 대한 빠른 스냅샷 복원 구성

스냅샷에 대해 빠른 스냅샷 복원은 기본적으로 비활성화되어 있습니다. 본인이 소유한 스냅샷과 본인에게 공유된 스냅샷에 대해 빠른 스냅샷 복원을 활성화하거나 비활성화할 수 있습니다. 스냅샷에 대해 빠른 스냅샷 복원을 활성화하거나 비활성화하면 변경 사항이 계정에만 적용됩니다.

> **Note**
>
> 스냅샷에 대해 빠른 스냅샷 복원을 활성화하면 특정 가용 영역에서 빠른 스냅샷 복원이 활성화된 1분마다 계정에 요금이 청구됩니다. 요금은 최소 1시간으로 비례 청구됩니다.

본인이 소유한 스냅샷을 삭제하면 계정의 해당 스냅샷에 대해 빠른 스냅샷 복원이 자동으로 비활성화됩니다. 본인에게 공유된 스냅샷에 대해 빠른 스냅샷 복원을 활성화하고 스냅샷 소유자가 해당 스냅샷을 삭제하거나 공유 해제하는 경우 계정의 공유 스냅샷에 대해 빠른 스냅샷 복원이 자동으로 비활성화됩니다.

본인에게 공유된 스냅샷에 대해 빠른 스냅샷 복원을 활성화하고 사용자 정의 CMK로 암호화된 경우 스냅샷 소유자가 사용자 정의 CMK에 대한 액세스를 취소해도 스냅샷에 대해 빠른 스냅샷 복원이 자동으로 비활성화되지 않습니다. 해당 스냅샷에 대해 빠른 스냅샷 복원을 수동으로 비활성화해야 합니다.

스냅샷에 대해 빠른 스냅샷 복원 기능을 사용하면 `optimizing` 상태가 됩니다. `optimizing` 상태인 스냅샷은 볼륨을 복원할 때 사용하면 어느 정도의 성능 이점이 있습니다. `enabled` 상태로 전환된 후에야 빠른 스냅샷 복원의 완전한 성능 이점이 생기게 됩니다.

Console

빠른 스냅샷 복원을 구성하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [스냅샷(Snapshots)]을 선택합니다.
3. 스냅샷을 선택하고 작업(Actions), 빠른 스냅샷 복원 관리(Manage fast snapshot restore)를 선택합니다.
4. 빠른 스냅샷 복원 설정 섹션에는 선택한 스냅샷에 대해 빠른 스냅샷 복원을 활성화할 수 있는 가용 영역이 모두 나열됩니다. 현재 상태(Current status) 볼륨은 각 영역에 대해 빠른 스냅샷 복원이 현재 활성화되어 있는지 아니면 비활성화되어 있는지 나타냅니다.

   현재 빠른 스냅샷 복원이 비활성화된 영역에서 빠른 스냅샷 복원을 사용하려면 해당 영역을 선택하고 활성화(Enable)를 선택한 다음 확인을 위해 활성화(Enable)를 선택합니다.

   현재 빠른 스냅샷 복원이 활성화된 영역에서 빠른 스냅샷 복원을 비활성화하려면 해당 영역을 선택하고 비활성화(Disable)를 선택합니다.
5. 필요한 변경을 수행한 후 닫기(Close)를 선택합니다.

AWS CLI

빠른 스냅샷 복원 활성화

enable-fast-snapshot-restores 명령을 사용합니다.

```
aws ec2 enable-fast-snapshot-restores \
    --availability-zones us-east-1a us-east-1b \
    --source-snapshot-ids snap-0abcdef1234567890
```

빠른 스냅샷 복원 비활성화

disable-fast-snapshot-restores 명령을 사용합니다.

```
aws ec2 disable-fast-snapshot-restores \
    --availability-zones us-east-1a \
    --source-snapshot-ids snap-0abcdef1234567890
```

다음 예제는 describe-fast-snapshot-restores 명령을 사용하여 비활성화된 빠른 스냅샷 복원을 설명합니다.

```
aws ec2 describe-fast-snapshot-restores \
    --filters Name=state,Values=disabled
```

PowerShell

빠른 스냅샷 복원 활성화

Enable-EC2FastSnapshotRestore cmdlet을 사용합니다.

```
Enable-EC2FastSnapshotRestore `
    -AvailabilityZone us-east-1a us-east-1b `
    -SourceSnapshotId snap-0abcdef1234567890
```

빠른 스냅샷 복원 비활성화

Disable-EC2FastSnapshotRestore cmdlet을 사용합니다.

```
Disable-EC2FastSnapshotRestore `
    -AvailabilityZone us-east-1a `
    -SourceSnapshotId snap-0abcdef1234567890
```

다음 예제는 Get-EC2FastSnapshotRestore cmdlet을 사용하여 비활성화된 빠른 스냅샷 복원을 설명합니다.

```
Get-EC2FastSnapshotRestore `
    -Filter @{Name="state"; Values="disabled"}
```

Amazon EBS 스냅샷에 대한 빠른 스냅샷 복원 상태 확인

스냅샷에 대한 빠른 스냅샷 복원이 다음 상태 중 하나일 수 있습니다.

- `enabling` — 빠른 스냅샷 복원 활성화를 요청했습니다.
- `optimizing` — 빠른 스냅샷 복원이 활성화되고 있습니다. 스냅샷을 최적화하려면 TiB당 60분이 소요됩니다. 이 상태의 스냅샷은 볼륨을 복원할 때 어느 정도의 성능 이점이 있습니다.
- `enabled` — 빠른 스냅샷 복원이 활성화되었습니다. 이 상태의 스냅샷은 충분한 볼륨 생성 크레딧이 있으며 볼륨을 복원할 때 완전한 성능 이점이 있습니다.
- `disabling` — 빠른 스냅샷 복원을 비활성화하도록 요청했거나 빠른 스냅샷 복원을 활성화하는 요청이 실패했습니다.
- `disabled` — 빠른 스냅샷 복원이 비활성화되었습니다. 필요에 따라 빠른 스냅샷 복원을 다시 활성화할 수 있습니다.

본인이 소유한 스냅샷 또는 본인에게 공유된 스냅샷에 대한 빠른 스냅샷 복원의 상태를 볼 수 있습니다.

Console

빠른 스냅샷 복원의 상태를 보려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [Snapshots]를 선택합니다.
3. 스냅샷을 선택합니다.
4. 세부 정보(Details) 탭에서 빠른 스냅샷 복원(Fast snapshot restore)은 빠른 스냅샷 복원의 상태를 표시합니다.

AWS CLI

빠른 스냅샷 복원이 활성화된 스냅샷을 보려면

describe-fast-snapshot-restores 명령을 사용합니다.

```
aws ec2 describe-fast-snapshot-restores --filters Name=state,Values=enabled
```

다음은 예제 출력입니다.

```
{
    "FastSnapshotRestores": [
        {
            "SnapshotId": "snap-0e946653493cb0447",
            "AvailabilityZone": "us-east-2a",
            "State": "enabled",
            "StateTransitionReason": "Client.UserInitiated - Lifecycle state
 transition",
            "OwnerId": "123456789012",
            "EnablingTime": "2020-01-25T23:57:49.596Z",
            "OptimizingTime": "2020-01-25T23:58:25.573Z",
            "EnabledTime": "2020-01-25T23:59:29.852Z"
        },
        {
            "SnapshotId": "snap-0e946653493cb0447",
            "AvailabilityZone": "us-east-2b",
            "State": "enabled",
            "StateTransitionReason": "Client.UserInitiated - Lifecycle state
 transition",
            "OwnerId": "123456789012",
            "EnablingTime": "2020-01-25T23:57:49.596Z",
            "OptimizingTime": "2020-01-25T23:58:25.573Z",
            "EnabledTime": "2020-01-25T23:59:29.852Z"
        }
    ]
}
```

PowerShell

빠른 스냅샷 복원이 활성화된 스냅샷을 보려면

Get-EC2FastSnapshotRestore cmdlet을 사용합니다.

```
Get-EC2FastSnapshotRestore `
    -Filter @{Name="state"; Values="enabled"}
```

빠른 스냅샷 복원을 사용하여 복원된 Amazon EBS 볼륨 보기

볼륨의 가용 영역에서 빠른 스냅샷 복원이 활성화된 스냅샷을 기반으로 볼륨을 생성하면 빠른 스냅샷 복원을 사용하여 볼륨이 복원됩니다.

AWS CLI

빠른 스냅샷 복원이 활성화된 스냅샷을 기반으로 생성된 볼륨을 확인하려면

describe-volumes 명령을 사용합니다.

```
aws ec2 describe-volumes --filters Name=fast-restored,Values=true
```

다음은 예제 출력입니다.

```
{
    "Volumes": [
        {
            "Attachments": [],
            "AvailabilityZone": "us-east-2a",
            "CreateTime": "2020-01-26T00:34:11.093Z",
            "Encrypted": true,
            "KmsKeyId": "arn:aws:kms:us-west-2:123456789012:key/8c5b2c63-b9bc-45a3-
a87a-5513e232e843",
            "Size": 20,
            "SnapshotId": "snap-0abcdef1234567890",
            "State": "available",
            "VolumeId": "vol-01234567890abcdef",
            "Iops": 100,
            "VolumeType": "gp2",
            "FastRestored": true
        }
    ]
}
```

PowerShell

빠른 스냅샷 복원이 활성화된 스냅샷을 기반으로 생성된 볼륨을 확인하려면

Get-EC2Volume cmdlet을 사용합니다.

```
Get-EC2Volume -Filter @{Name="fast-restored"; Values="true"}
```

Amazon EBS 스냅샷 잠금

Amazon EBS 스냅샷을 잠가서 우발적이거나 악의적인 삭제로부터 스냅샷을 보호하거나 특정 기간 동안 WORM(Write-Once-Read-Many) 형식으로 저장할 수 있습니다. 스냅샷이 잠겨 있는 동안에는 IAM 권한에 관계없이 어떤 사용자도 스냅샷을 삭제할 수 없습니다. 다른 스냅샷을 사용하는 것과 동일한 방식으로 잠긴 스냅샷을 계속 사용할 수 있습니다.

> **Note**
>
> 스냅샷 잠금은 SEC 17a-4, CFTC 및 FINRA 규정의 적용을 받는 환경에서 Cohasset Associates에 의해 평가되었습니다. 스냅샷 잠금과 이러한 규정의 관계에 대한 자세한 내용은 Cohasset Associates Compliance Assessment를 참조하세요.

규정 준수 모드 또는 거버넌스 모드의 두 가지 모드 중 하나로 스냅샷을 특정 기간 동안 또는 특정 날짜까지 잠글 수 있습니다. 자세한 내용은 잠금 모드 및 잠금 지속 시간 섹션을 참조하세요.

가격 책정

추가 비용 없이 스냅샷을 잠그고 잠금 해제할 수 있습니다. 잠긴 스냅샷에 대한 표준 Amazon EBS 스냅샷 스토리지 비용을 지불합니다.

주제

- Amazon EBS 스냅샷 잠금 개념
- Amazon EBS 스냅샷 잠금 고려 사항
- Amazon EBS 스냅샷 잠금에 대한 액세스 제어
- Amazon EBS 스냅샷 잠금
- Amazon EBS 스냅샷 잠금 해제
- Amazon EBS 스냅샷 잠금 설정 업데이트
- Amazon EBS 스냅샷 잠금 모니터링

Amazon EBS 스냅샷 잠금 개념

다음은 스냅샷 잠금 사용을 시작하려면 알아야 하는 중요한 개념입니다.

목차

- 잠금 모드

- 잠금 지속 시간
- 쿨링 오프 기간
- 잠금 상태

잠금 모드

다음 두 가지 모드 중 하나로 스냅샷을 잠글 수 있습니다.

거버넌스 모드

스냅샷이 잠긴 후 적절한 IAM 권한을 가진 사용자는 언제든지 스냅샷을 잠금 해제하고 잠금 모드, 잠금 기간 또는 만료 날짜를 수정할 수 있습니다. 거버넌스 모드에서 스냅샷을 잠그면 스냅샷이 즉시 잠기므로 쿨링 오프 기간이 없습니다. 거버넌스 모드에서 잠긴 후 스냅샷을 삭제하려면 먼저 스냅샷을 잠금 해제하거나 잠금이 만료될 때까지 기다려야 합니다.

거버넌스 모드를 사용하면 특정 사용자에게만 스냅샷을 잠금 해제하고 스냅샷 잠금 구성을 수정할 수 있는 권한이 부여되어 조직의 데이터 거버넌스 요구 사항을 충족할 수 있습니다. 또한 거버넌스 모드를 사용하여 규정 준수 모드에서 스냅샷을 잠그기 전에 잠금 구성을 테스트할 수 있습니다.

규정 준수 모드

규정 준수 모드에서 스냅샷을 잠그는 경우 스냅샷을 잠근 후 즉시 시작되는 쿨링 오프 기간을 선택적으로 지정할 수 있습니다. 쿨링 오프 기간 동안 적절한 권한이 있는 사용자는 스냅샷을 잠금 해제하고, 잠금 모드를 변경하고, 쿨링 오프 기간, 잠금 기간 또는 만료 날짜를 늘리거나 줄일 수 있습니다. 쿨링 오프 기간이 만료된 후에는 스냅샷을 잠금 해제하거나, 잠금 모드를 변경하거나, 잠금 기간 또는 만료 날짜를 줄일 수 없으며 잠금 기간이나 만료 날짜를 늘릴 수만 있습니다. 규정 준수 상태에서 스냅샷이 잠겼다가 쿨링 오프 기간이 만료된 후 스냅샷을 삭제하려면 잠금이 만료될 때까지 기다려야 합니다.

> **Note**
>
> 요청에서 쿨링 오프 기간을 생략하여 쿨링 오프 기간 없이 규정 준수 모드에서 스냅샷을 잠글 수 있습니다. 이렇게 하면 잠금이 즉시 적용되고 스냅샷을 잠금 해제하거나, 잠금 모드를 변경하거나, 잠금 기간 또는 만료 날짜를 줄일 수 없으며 잠금 기간이나 만료 날짜를 늘릴 수만 있습니다.

규정 준수 모드를 사용하면 규정 준수를 위해 특정 기간 동안 삭제해서는 안 되는 스냅샷을 보호할 수 있습니다. 규정 준수 모드는 다음과 같은 이점을 제공합니다.

- 스냅샷에 대한 WORM(Write-Once, Read-Many) 구성을 활성화합니다.
- 이는 우발적이거나 악의적인 삭제로부터 스냅샷을 보호하는 추가 방어 계층을 제공합니다.
- 조직의 데이터 보호 정책 및 절차를 충족하기 위해 권한 있는 사용자가 조기 삭제하는 것을 방지하는 보존 기간을 적용합니다.

> **Note**
>
> 잠금이 만료되기 전에 규정 준수 모드에서 잠긴 스냅샷을 삭제하는 유일한 방법은 연결된 AWS 계정을 닫는 것입니다.

잠금 지속 시간

잠금 기간은 스냅샷이 잠긴 상태로 유지되는 기간입니다. 잠금 기간을 다음 중 하나로 지정할 수 있지만 둘 다 지정할 수는 없습니다.

일수

잠금 기간은 스냅샷이 잠긴 상태로 유지되는 일수로 지정됩니다. 지정된 일수가 경과하면 스냅샷이 자동으로 잠금 해제됩니다. 기간은 1일에서 3만 6,500일(100년) 사이일 수 있습니다.

잠금 만료 날짜

잠금 기간은 미래의 만료 날짜에 따라 결정됩니다. 스냅샷은 잠금 만료 날짜에 도달할 때까지 잠긴 상태로 유지됩니다. 잠금 만료 날짜에 도달하면 스냅샷이 자동으로 잠금 해제됩니다.

쿨링 오프 기간

쿨링 오프 기간은 규정 준수 모드에서 스냅샷을 잠글 때 지정할 수 있는 선택적 기간입니다. 쿨링 오프 기간 동안 적절한 권한이 있는 사용자는 스냅샷을 잠금 해제하고, 잠금 모드를 변경하고, 쿨링 오프 기간과 잠금 기간을 늘리거나 줄일 수 있습니다. 쿨링 오프 기간이 만료되면 사용자는 권한에 관계없이 스냅샷을 잠금 해제하거나, 잠금 모드를 변경하거나, 쿨링 오프 기간을 복구하거나, 잠금 기간을 줄일 수 없습니다.

쿨링 오프 기간 중에는 스냅샷을 삭제할 수 없습니다.

지정된 경우 쿨링 오프 기간은 스냅샷을 잠근 직후에 시작됩니다. 생략하면 스냅샷은 쿨링 오프 기간 없이 즉시 규정 준수 모드에서 잠깁니다.

쿨링 오프 기간은 1~72시간일 수 있습니다. 쿨링 오프 기간 없이 즉시 규정 준수 모드에서 스냅샷을 잠그려면 요청에 쿨링 오프 기간을 지정하지 마세요.

잠금 상태

스냅샷 잠금은 다음 상태 중 하나일 수 있습니다.

- `compliance-cooloff` - 스냅샷이 규정 준수 모드에서 잠겼지만 아직 쿨링 오프 기간 내에 있습니다. 스냅샷은 삭제할 수 없지만 잠금 해제할 수 있으며 적절한 권한이 있는 사용자가 잠금 설정을 수정할 수 있습니다.
- `governance` - 스냅샷은 거버넌스 모드에서 잠깁니다. 스냅샷은 삭제할 수 없지만 잠금 해제할 수 있으며 적절한 권한이 있는 사용자가 잠금 설정을 수정할 수 있습니다.
- `compliance` - 스냅샷이 쿨링 오프 기간 없이 규정 준수 모드에서 잠기거나 쿨링 오프 기간이 만료되었습니다. 스냅샷을 잠금을 해제하거나 삭제할 수 없습니다. 적절한 권한을 가진 사용자만 잠금 기간을 늘릴 수 있습니다.
- `expired` - 스냅샷이 규정 준수 또는 거버넌스 모드에서 잠겼지만 잠금이 만료되었습니다. 스냅샷은 잠기지 않으므로 삭제할 수 있습니다.

Amazon EBS 스냅샷 잠금 고려 사항

Amazon EBS 스냅샷을 잠글 때는 다음 사항에 유의해야 합니다.

- 스냅샷이 `pending` 또는 `completed` 상태인 경우에만 스냅샷을 잠글 수 있습니다.
  - `pending` 상태에 있는 스냅샷을 잠그고 특정 기간 동안 잠그면 스냅샷이 `completed` 상태에 도달할 때만 잠금 기간이 시작됩니다. 스냅샷이 `pending` 상태에 있는 동안에는 삭제할 수 없습니다.
  - `pending` 상태에서 스냅샷을 잠그고 어떤 이유로든 스냅샷 생성이 실패하면 잠금이 취소됩니다.
- 쿨링 오프 기간이 만료된 후 규정 준수 모드에서 잠긴 스냅샷의 잠금 기간을 연장하는 경우 다른 쿨링 오프 기간을 지정할 수 없습니다. 쿨링 오프 기간을 지정하면 요청이 실패합니다.
- 아카이빙된 스냅샷을 잠글 수 있습니다. 그리고 잠긴 스냅샷을 아카이브할 수 있습니다.
- AMI와 연결된 스냅샷은 잠글 수 있습니다.
- 잠금 상태인 연결된 스냅샷이 있는 AMI를 등록 취소할 수 있습니다.
- 잠긴 스냅샷을 암호화하는 데 사용되는 KMS 키를 삭제할 수 있습니다.
- 에서 생성한 스냅샷을 잠그지 않는 것이 좋습니다 AWS Backup. AWS Backup 보존 기간이 만료되기 전에 스냅샷이 삭제되지 않도록 해야 합니다. 에서 관리하는 스냅샷에 대한 보안 계층을 추가하

려면 AWS Backup 볼트 잠금을 사용하는 AWS Backup것이 좋습니다. 자세한 내용은 AWS Backup Vault Lock을 참조하세요.

- 생성 중이나 AMI 등록 중에는 스냅샷을 잠글 수 없습니다.
- AWS Outposts에서는 로컬 Amazon EBS 스냅샷을 잠글 수 없습니다.
- 잠금이 만료되기 전에 규정 준수 모드에서 잠긴 스냅샷을 삭제하는 유일한 방법은 연결된 AWS 계정을 닫는 것입니다.

  스냅샷이 잠긴 상태에서 AWS 계정을 닫으면는 스냅샷이 그대로 유지된 상태에서 90일 동안 계정을 일시 AWS 중지합니다. 90일 이내에 계정을 다시 열지 않으면는 스냅샷이 잠긴 경우에도 스냅샷을 AWS 삭제합니다.

Amazon EBS 스냅샷 잠금에 대한 액세스 제어

기본적으로 사용자에게는 스냅샷 잠금 사용 권한이 없습니다. 사용자가 스냅샷 잠금을 사용하도록 허용하려면 특정 리소스 및 API 작업을 사용할 권한을 부여하는 IAM 정책을 생성해야 합니다. 자세한 내용은 IAM 사용 설명서의 IAM 정책 생성을 참조하세요.

주제

- 필수 권한
- 조건 키로 액세스 제한

필수 권한

스냅샷 잠금을 사용하려면 사용자에게 다음 권한이 필요합니다.

- `ec2:LockSnapshot` - 스냅샷 잠금
- `ec2:UnlockSnapshot` - 스냅샷 잠금 해제
- `ec2:DescribeLockedSnapshots` - 스냅샷 잠금 설정 보기

다음은 스냅샷을 잠금 및 잠금 해제하고 스냅샷 잠금 설정을 볼 수 있는 권한을 사용자에게 부여하는 IAM 정책의 예제입니다. 여기에는 콘솔 사용자에 대한 `ec2:DescribeSnapshots` 권한이 포함됩니다. 일부 권한이 필요하지 않은 경우 정책에서 권한을 제거할 수 있습니다.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "AllowSnapshotLockOperations",
      "Effect": "Allow",
      "Action": [
        "ec2:LockSnapshot",
        "ec2:UnlockSnapshot",
        "ec2:DescribeLockedSnapshots",
        "ec2:DescribeSnapshots"
      ],
      "Resource": [
        "arn:aws:ec2:*::snapshot/*",
        "arn:aws:ec2:*:111122223333:volume/*"
      ]
    }
  ]
}
```

액세스 권한을 제공하려면 사용자, 그룹 또는 역할에 권한을 추가하세요.

- 의 사용자 및 그룹 AWS IAM Identity Center:

  권한 세트를 생성합니다. AWS IAM Identity Center 사용자 안내서에서 권한 세트 생성의 지침을 따릅니다.
- ID 제공업체를 통해 IAM에서 관리되는 사용자:

  ID 페더레이션을 위한 역할을 생성합니다. IAM 사용자 설명서의 Create a role for a third-party identity provider (federation)의 지침을 따릅니다.
- IAM 사용자:
  - 사용자가 맡을 수 있는 역할을 생성합니다. IAM 사용자 설명서에서 Create a role for an IAM user의 지침을 따릅니다.
  - (권장되지 않음) 정책을 사용자에게 직접 연결하거나 사용자를 사용자 그룹에 추가합니다. IAM 사용 설명서에서 사용자(콘솔)에 권한 추가의 지침을 따릅니다.

조건 키로 액세스 제한

조건 키를 사용하여 사용자가 스냅샷을 잠그는 방법을 제한할 수 있습니다.

주제

- ec2:SnapshotLockDuration
- ec2:CoolOffPeriod

ec2:SnapshotLockDuration

`ec2:SnapshotLockDuration` 조건 키를 사용하여 스냅샷을 잠글 때 특정 잠금 기간으로 사용자를 제한할 수 있습니다.

다음 예제 정책은 사용자가 잠금 기간을 `10~50`일로 지정하도록 제한합니다.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "AllowSnapshotLockWithDurationCondition",
      "Effect": "Allow",
      "Action": "ec2:LockSnapshot",
      "Resource": "arn:aws:ec2:*::snapshot/*",
      "Condition": {
        "NumericGreaterThan": {
          "ec2:SnapshotLockDuration": 10
        },
        "NumericLessThan": {
          "ec2:SnapshotLockDuration": 50
        }
      }
    }
  ]
}
```

ec2:CoolOffPeriod

`ec2:CoolOffPeriod` 조건 키를 사용하여 쿨링 오프 기간 없이 사용자가 규정 준수 모드에서 스냅샷을 잠그지 못하도록 할 수 있습니다.

다음 예제 정책은 사용자가 규정 준수 모드에서 스냅샷을 잠글 때 쿨링 오프 기간을 48 시간 이상으로 지정하도록 제한합니다.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "AllowSnapshotLockWithCondition",
      "Effect": "Allow",
      "Action": "ec2:LockSnapshot",
      "Resource": "arn:aws:ec2:*::snapshot/*",
      "Condition": {
        "NumericGreaterThan": {
          "ec2:SnapshotTime": 48
        }
      }
    }
  ]
}
```

Amazon EBS 스냅샷 잠금

스냅샷이 `pending` 또는 `completed` 상태인 경우 스냅샷을 잠글 수 있습니다. 자세한 내용은 Amazon EBS 스냅샷 잠금 고려 사항 단원을 참조하십시오.

Console

스냅샷 잠금

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [스냅샷(Snapshots)]을 선택합니다.

3. 잠글 스냅샷을 선택하고 작업, 스냅샷 설정, 스냅샷 잠금 관리를 선택합니다.
4. 스냅샷 잠금을 선택합니다.
5. 잠금 모드에서 거버넌스 모드 또는 규정 준수 모드를 선택합니다. 자세한 내용은 잠금 모드 단원을 참조하십시오.
6. 잠금 기간에서 다음 중 하나를 수행합니다.

   - 특정 기간 동안 스냅샷을 잠그려면 스냅샷 잠금 대상을 선택한 다음 기간을 일 또는 연 단위로 입력합니다.
   - 특정 날짜 및 시간까지 스냅샷을 잠그려면 스냅샷 잠금 기한을 선택한 다음 만료 날짜 및 시간을 선택합니다.

   자세한 내용은 잠금 지속 시간 단원을 참조하십시오.
7. (규정 준수 모드만 해당) 쿨링 오프 기간에는 스냅샷을 잠금 해제하고 잠금 구성을 수정할 수 있는 쿨링 오프 기간을 지정합니다. 자세한 내용은 쿨링 오프 기간 단원을 참조하십시오.
8. (규정 준수 모드만 해당) 규정 준수 모드에서 스냅샷을 잠그고 쿨링 오프 기간이 만료된 후에는 스냅샷을 잠금 해제할 수 없음을 확인하려면 승인을 선택합니다.
9. 잠금 설정 저장을 선택합니다.

AWS CLI

거버넌스 모드에서 스냅샷 잠금

lock-snapshot 명령을 사용합니다. `--lock-mode`에서 `governance`를 지정합니다. 특정 기간(`--lock-duration`) 동안 스냅샷을 잠그려면 기간을 일 단위로 지정합니다.

```
aws ec2 lock-snapshot \
    --snapshot-id snap-0abcdef1234567890 \
    --lock-mode governance \
    --lock-duration 30
```

특정 날짜까지 스냅샷을 잠그려면 `--expiration-date`에 대해 잠금이 만료되어야 하는 날짜와 시간을 UTC 시간대로 지정합니다.

```
aws ec2 lock-snapshot \
    --snapshot-id snap-0abcdef1234567890 \
    --lock-mode governance \
```

```
    --expiration-date YYYY-MM-DDThh:mm:ss.sssZ
```

규정 준수 모드에서 스냅샷 잠금

lock-snapshot 명령을 사용합니다. --lock-mode에서 compliance를 지정합니다. --cool-off-period에 대해 선택적으로 쿨링 오프 기간을 시간 단위로 지정합니다. 특정 기간(--lock-duration) 동안 스냅샷을 잠그려면 스냅샷을 잠글 일 수를 지정합니다.

```
aws ec2 lock-snapshot \
    --snapshot-id snap-0abcdef1234567890 \
    --lock-mode compliance \
    --cool-off-period 24 \
    --lock-duration 30
```

특정 날짜까지 스냅샷을 잠그려면 --expiration-date에 대해 잠금이 만료되어야 하는 날짜와 시간을 UTC 시간대로 지정합니다.

```
aws ec2 lock-snapshot \
    --snapshot-id snap-0abcdef1234567890 \
    --lock-mode compliance \
    --expiration-date YYYY-MM-DDThh:mm:ss.sssZ
```

PowerShell

거버넌스 모드에서 스냅샷 잠금

Lock-EC2Snapshot cmdlet을 사용합니다. 선택적으로 Snapshot Lock 기간을 일 단위로 지정할 수 있습니다.

```
Lock-EC2Snapshot `
    -SnapshotId snap-0abcdef1234567890 `
    -LockMode "governance" `
    -LockDuration 30
```

또는 UTC 시간대의 특정 날짜까지 스냅샷을 잠글 수 있습니다.

```
Lock-EC2Snapshot `
    -SnapshotId snap-0abcdef1234567890 `
    -LockMode "governance" `
    -ExpirationDate YYYY-MM-DDThh:mm:ss.sssZ
```

규정 준수 모드에서 스냅샷 잠금

Lock-EC2Snapshot cmdlet을 사용합니다. 선택적으로 쿨링 오프 기간을 시간 단위로 지정합니다. 또한 선택적으로 Snapshot Lock 기간을 일 단위로 지정할 수 있습니다.

```
Lock-EC2Snapshot `
    -SnapshotId snap-0abcdef1234567890 `
    -LockMode "compliance" `
    -CoolOffPeriod 24 `
    -LockDuration 30
```

또는 UTC 시간대의 특정 날짜까지 스냅샷을 잠글 수 있습니다.

```
Lock-EC2Snapshot `
    -SnapshotId snap-0abcdef1234567890 `
    -LockMode "compliance" `
    -ExpirationDate YYYY-MM-DDThh:mm:ss.sssZ
```

Amazon EBS 스냅샷 잠금 해제

스냅샷이 거버넌스 모드에서 잠겨 있거나 규정 준수 모드에서 잠겨 있고 아직 쿨링 오프 기간 내에 있는 경우에만 스냅샷을 잠금 해제할 수 있습니다.

Console

스냅샷 잠금 해제

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [스냅샷(Snapshots)]을 선택합니다.
3. 잠금 해제할 스냅샷을 선택하고 작업, 스냅샷 설정, 스냅샷 잠금 관리를 선택합니다.
4. 스냅샷 잠금 해제를 선택한 다음 스냅샷 잠금 해제를 다시 선택하여 확인합니다.

AWS CLI

스냅샷 잠금 해제

unlock-snapshot 명령을 사용합니다.

```
aws ec2 unlock-snapshot --snapshot-id snap-0abcdef1234567890
```

PowerShell

스냅샷 잠금 해제

Unlock-EC2Snapshot cmdlet을 사용합니다.

```
Unlock-EC2Snapshot -SnapshotId snap-0abcdef1234567890
```

Amazon EBS 스냅샷 잠금 설정 업데이트

허용되는 업데이트는 잠금 상태에 따라 달라집니다.

- `governance` - 잠금 모드를 변경하고 잠금 기간 또는 만료 날짜를 늘리거나 줄일 수 있습니다.
- `compliance-cooloff` - 잠금 모드를 변경하고, 쿨링 오프 기간, 잠금 기간 또는 만료 날짜를 늘리거나 줄일 수 있습니다.
- `compliance` - 잠금 기간 또는 만료 날짜를 늘릴 수 있습니다.

Console

스냅샷 잠금 설정 업데이트

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [스냅샷(Snapshots)]을 선택합니다.
3. 잠금 설정을 수정할 스냅샷을 선택하고 작업, 스냅샷 설정, 스냅샷 잠금 관리를 선택합니다.
4. 필요에 따라 설정을 업데이트한 다음 잠금 설정 저장을 선택합니다.

AWS CLI

스냅샷 잠금 설정 업데이트

lock-snapshot 명령을 사용합니다. 스냅샷의 ID와 수정할 옵션을 지정합니다. 다음 예제는 만료 날짜를 변경합니다.

```
aws ec2 lock-snapshot \
    --snapshot-id snap-0abcdef1234567890 \
    --lock-mode governance \
    --expiration-date YYYY-MM-DDThh:mm:ss.sssZ
```

PowerShell

스냅샷 잠금 설정 업데이트

Lock-EC2Snapshot cmdlet을 사용합니다. 스냅샷의 ID와 수정할 옵션을 지정합니다. 다음 예제는 만료 날짜를 변경합니다.

```
Lock-EC2Snapshot `
    -SnapshoId snap-0abcdef1234567890 `
    -LockMode "governance" `
    -ExpirationDate YYYY-MM-DDThh:mm:ss.sssZ
```

Amazon EBS 스냅샷 잠금 모니터링

다음 도구를 사용하여 Amazon EBS 스냅샷 잠금과 관련된 작업을 모니터링할 수 있습니다.

주제

- 를 사용하여 Amazon EBS 스냅샷 잠금 모니터링 AWS CloudTrail
- Amazon EventBridge를 사용하여 Amazon EBS 스냅샷 잠금 모니터링

를 사용하여 Amazon EBS 스냅샷 잠금 모니터링 AWS CloudTrail

콘솔의 직접 호출 및 API에 대한 코드 직접 호출을 포함하여 스냅샷 잠금에 대한 API 직접 호출을 이벤트로 모니터링할 수 있습니다. CloudTrail에서 수집한 정보를 사용하여 수행된 요청, 요청이 수행된 IP 주소, 요청을 수행한 사람, 요청이 수행된 시간 및 추가 세부 정보를 확인할 수 있습니다.

자세한 내용은 AWS CloudTrail을 사용하여 API 직접 호출 로깅을 참조하세요.

Amazon EventBridge를 사용하여 Amazon EBS 스냅샷 잠금 모니터링

Amazon EBS는 스냅샷 잠금 작업과 관련된 이벤트를 발생시킵니다. AWS Lambda 및 Amazon EventBridge를 사용하여 이벤트 알림을 프로그래밍 방식으로 처리할 수 있습니다. 이벤트는 최선의 작업을 기반으로 발생됩니다. 자세한 내용은 Amazon EventBridge 사용 설명서를 참조하세요.

다음과 같은 이벤트가 발생합니다.

- 거버넌스 또는 규정 준수 모드에서 성공적으로 잠긴 스냅샷

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

```
  "id": "01234567-01234-0123-0123-012345678901",
  "detail-type": "EBS Snapshot Notification",
  "source": "aws.ec2",
  "account": "012345678901",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "region": "us-east-1",
  "resources": [
    "arn:aws:ec2::us-west-2:snapshot/snap-01234567890abcdef"
  ],
  "detail": {
    "event": "lockSnapshot",
    "result": "succeeded",
    "snapshot_id": "arn:aws:ec2::us-west-2:snapshot/snap-01234567890abcdef",
    "source": 012345678901,
    "lockState": "compliance-cooloff",
    "lockCreatedOn": "yyyy-mm-ddThh:mm:ssZ",
    "lockExpiresOn": "yyyy-mm-ddThh:mm:ssZ",
    "lockDuration": 123,
    "lockStartDurationTime": "yyyy-mm-ddThh:mm:ssZ",
    "cooOffPeriod": 24,
    "coolOffPeriodExpiresOn": "yyyy-mm-ddThh:mm:ssZ"
  }
}
```

- 스냅샷이 pending 상태에서 잠기고 completed 상태에 도달하지 못할 때 실패한 잠금 이벤트

```
{
  "version": "0",
  "id": "01234567-01234-0123-0123-012345678901",
  "detail-type": "EBS Snapshot Notification",
  "source": "aws.ec2",
  "account": "012345678901",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "region": "us-east-1",
  "resources": [
    "arn:aws:ec2::us-west-2:snapshot/snap-01234567890abcdef"
  ],
  "detail": {
    "event": "lockSnapshot",
    "result": "failed",
    "cause": "snapshot failed",
    "snapshot_id": "arn:aws:ec2::us-west-2:snapshot/snap-01234567890abcdef",
    "lockState": "pending-compliance",
    "lockCreatedOn": "yyyy-mm-ddThh:mm:ssZ",
```

```
    "lockDuration": 123,
    "lockStartDurationTime": "yyyy-mm-ddThh:mm:ssZ",
    "cooOffPeriod": 24,
    "coolOffPeriodExpiresOn": "yyyy-mm-ddThh:mm:ssZ"
  }
}
```

- 잠금 만료

```
{
  "version": "0",
  "id": "01234567-01234-0123-0123-012345678901",
  "detail-type": "EBS Snapshot Notification",
  "source": "aws.ec2",
  "account": "012345678901",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "region": "us-east-1",
  "resources": [
    "arn:aws:ec2::us-west-2:snapshot/snap-01234567890abcdef"
  ],
  "detail": {
    "event": "lockDurationExpiry",
    "result": "succeeded",
    "snapshot_id": "arn:aws:ec2::us-west-2:snapshot/snap-01234567890abcdef",
    "lockState": "expired",
    "lockCreatedOn": "yyyy-mm-ddThh:mm:ssZ",
    "lockExpiresOn": "yyyy-mm-ddThh:mm:ssZ",
    "lockDuration": 123
  }
}
```

- 규정 준수 모드에서 잠긴 후 쿨링 오프 기간 만료됨

```
{
  "version": "0",
  "id": "01234567-01234-0123-0123-012345678901",
  "detail-type": "EBS Snapshot Notification",
  "source": "aws.ec2",
  "account": "012345678901",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "region": "us-east-1",
  "resources": [
    "arn:aws:ec2::us-west-2:snapshot/snap-01234567890abcdef"
```

```
  ],
  "detail": {
    "event": "cooloffperiodExpiry",
    "result": "succeeded",
    "snapshot_id": "arn:aws:ec2::us-west-2:snapshot/snap-01234567890abcdef",
    "lockState": "compliance",
    "lockCreatedOn": "yyyy-mm-ddThh:mm:ssZ",
    "lockExpiresOn": "yyyy-mm-ddThh:mm:ssZ",
    "lockDuration": 123,
    "lockStartDurationTime": "yyyy-mm-ddThh:mm:ssZ",
    "cooOffPeriod": 24,
    "coolOffPeriodExpiresOn": "yyyy-mm-ddThh:mm:ssZ"
  }
}
```

Amazon EBS 스냅샷에 대한 퍼블릭 액세스 차단

스냅샷의 퍼블릭 공유를 방지하려면 스냅샷에 대한 퍼블릭 액세스 차단을 활성화합니다. 리전에서 스냅샷에 대한 퍼블릭 액세스 차단을 활성화하면 해당 리전에서 스냅샷을 공개적으로 공유하려는 모든 시도가 자동으로 차단됩니다. 이를 통해 스냅샷의 보안을 강화하고 무단 액세스나 의도하지 않은 액세스로부터 스냅샷 데이터를 보호할 수 있습니다.

다음 두 가지 모드 중 하나에서 스냅샷에 대한 퍼블릭 액세스 차단을 활성화할 수 있습니다.

- 모든 공유 차단 - 스냅샷의 모든 퍼블릭 공유를 차단합니다. 계정의 사용자는 새 퍼블릭 공유를 요청할 수 없습니다. 또한 이미 공개적으로 공유된 스냅샷은 비공개로 취급되어 더 이상 공개적으로 사용할 수 없습니다.
- 새 공유 차단 - 스냅샷의 새로운 퍼블릭 공유만 차단합니다. 계정의 사용자는 새 퍼블릭 공유를 요청할 수 없습니다. 그러나 이미 공개적으로 공유된 스냅샷은 계속 공개적으로 사용할 수 있습니다.

고려 사항

스냅샷에 대한 퍼블릭 액세스 차단을 사용할 때 다음 사항에 유의해야 합니다.

- 스냅샷에 대한 퍼블릭 액세스를 차단해도 프라이빗 스냅샷 공유가 차단되지는 않습니다.
- 모든 공유 차단 모드에서 스냅샷에 대한 퍼블릭 액세스 차단을 활성화해도 이미 공개적으로 공유된 스냅샷에 대한 권한은 변경되지 않습니다. 대신 이러한 스냅샷이 공개적으로 표시되거나 액세스할 수 없게 됩니다. 따라서 이러한 스냅샷의 속성은 스냅샷이 공개적으로 사용 불가능하더라도 여전히 공개적으로 공유됨을 나타냅니다.

나중에 퍼블릭 액세스 차단을 비활성화하거나 모드를 새 공유 차단으로 변경하면 이러한 스냅샷을 다시 공개적으로 사용할 수 있습니다.

- 스냅샷에 대한 퍼블릭 액세스 차단은 리전 설정입니다. 이 설정이 활성화된 리전의 모든 스냅샷에 적용됩니다. 스냅샷의 퍼블릭 공유를 방지하려는 각 리전에서 스냅샷에 대한 퍼블릭 액세스 차단을 활성화해야 합니다.
- 퍼블릭 액세스 차단은 계정 수준 설정입니다. 이 설정은 관리자 사용자를 포함한 계정의 모든 사용자에게 적용됩니다. 조직 수준에서는 스냅샷에 대한 퍼블릭 액세스 차단을 활성화할 수 없습니다.
- 퍼블릭 액세스 차단 설정은 계정 수준에서 직접 구성되거나 선언적 정책을 사용하여 구성됩니다. 선언적 정책을 사용하면 여러 리전과 여러 계정에 동시에 설정을 적용할 수 있습니다. 선언적 정책을 사용 중인 경우 계정 내에서 직접 설정을 수정할 수 없습니다. 이 주제에서는 계정 내에서 직접 설정을 구성하는 방법을 설명합니다. 선언적 정책 사용에 대한 자세한 내용은 AWS Organizations 사용 설명서의 선언적 정책을 참조하세요.
- 스냅샷에 대한 퍼블릭 액세스를 차단해도 EBS 지원 AMI의 퍼블릭 공유가 차단되지는 않습니다. 스냅샷에 대한 퍼블릭 액세스 차단을 활성화해도 사용자는 여전히 EBS 지원 AMI를 공개적으로 공유할 수 있습니다. EBS 지원 AMI가 공개적으로 공유되는 경우 해당 AMI에 대한 액세스 권한이 있는 사용자는 연결된 스냅샷에서 볼륨을 생성할 수 있습니다. AMI의 퍼블릭 공유를 방지하려면 AMI에 대한 퍼블릭 액세스 차단을 사용합니다.
- 의 로컬 스냅샷에서는 스냅샷에 대한 퍼블릭 액세스 차단이 지원되지 않습니다 AWS Outposts.

가격 책정

스냅샷에 대한 퍼블릭 액세스 차단을 추가 비용 없이 활성화할 수 있습니다.

목차

- Amazon EBS 스냅샷에 대한 퍼블릭 액세스를 차단하기 위한 IAM 권한
- Amazon EBS 스냅샷에 대한 퍼블릭 액세스 차단 구성
- Amazon EBS 스냅샷에 대한 퍼블릭 액세스 차단 설정 보기
- Amazon EBS 스냅샷에 대한 퍼블릭 액세스 차단 비활성화
- EventBridge를 사용하여 Amazon EBS 스냅샷에 대한 퍼블릭 액세스 차단 모니터링

Amazon EBS 스냅샷에 대한 퍼블릭 액세스를 차단하기 위한 IAM 권한

기본적으로 사용자에게는 스냅샷에 대한 퍼블릭 액세스 차단 사용 권한이 없습니다. 사용자가 스냅샷에 대한 퍼블릭 액세스 차단을 사용하도록 허용하려면 특정 API 작업을 사용할 권한을 부여하는 IAM 정책을 생성해야 합니다. 정책이 생성되면 사용자, 그룹 또는 역할에 권한을 추가해야 합니다.

스냅샷에 대한 퍼블릭 액세스 차단을 사용하려면 사용자에게 다음 권한이 필요합니다.

- `ec2:EnableSnapshotBlockPublicAccess` - 스냅샷에 대한 퍼블릭 액세스 차단을 활성화하고 모드를 수정합니다.
- `ec2:DisableSnapshotBlockPublicAccess` - 스냅샷에 대한 퍼블릭 액세스 차단을 비활성화합니다.
- `ec2:GetSnapshotBlockPublicAccessState` - 리전의 스냅샷에 대한 퍼블릭 액세스 차단 설정을 봅니다.

다음은 예시 IAM 정책입니다. 일부 권한이 필요하지 않은 경우 정책에서 권한을 제거할 수 있습니다.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [{
        "Effect": "Allow",
        "Action": [
            "ec2:EnableSnapshotBlockPublicAccess",
            "ec2:DisableSnapshotBlockPublicAccess",
            "ec2:GetSnapshotBlockPublicAccessState"
        ],
        "Resource": "*"
    }]
}
```

액세스 권한을 제공하려면 사용자, 그룹 또는 역할에 권한을 추가하세요.

- 의 사용자 및 그룹 AWS IAM Identity Center:

  권한 세트를 생성합니다. AWS IAM Identity Center 사용자 안내서에서 권한 세트 생성의 지침을 따릅니다.

- ID 제공업체를 통해 IAM에서 관리되는 사용자:

  ID 페더레이션을 위한 역할을 생성합니다. IAM 사용자 설명서의 Create a role for a third-party identity provider (federation)의 지침을 따릅니다.

- IAM 사용자:
  - 사용자가 맡을 수 있는 역할을 생성합니다. IAM 사용자 설명서에서 Create a role for an IAM user의 지침을 따릅니다.
  - (권장되지 않음) 정책을 사용자에게 직접 연결하거나 사용자를 사용자 그룹에 추가합니다. IAM 사용 설명서에서 사용자(콘솔)에 권한 추가의 지침을 따릅니다.

Amazon EBS 스냅샷에 대한 퍼블릭 액세스 차단 구성

리전에서 스냅샷이 공개적으로 공유되는 것을 방지하려면 스냅샷에 대한 퍼블릭 액세스 차단을 활성화하세요. 이 기능이 활성화되면 해당 리전에서 스냅샷을 공개적으로 공유하려는 요청이 차단됩니다.

> **Important**
>
> 모든 공유 차단 모드에서 스냅샷에 대한 퍼블릭 액세스 차단을 활성화해도 이미 공개적으로 공유된 스냅샷에 대한 권한은 변경되지 않습니다. 대신 이러한 스냅샷이 공개적으로 표시되거나 액세스할 수 없게 됩니다. 따라서 이러한 스냅샷의 속성은 스냅샷이 공개적으로 사용 불가능하더라도 여전히 공개적으로 공유됨을 나타냅니다.
> 나중에 퍼블릭 액세스 차단을 비활성화하거나 모드를 새 공유 차단으로 변경하면 이러한 스냅샷을 다시 공개적으로 사용할 수 있습니다.

> **Note**
>
> 이 설정은 계정 수준에서 직접 구성되거나 선언적 정책을 사용하여 구성됩니다. 스냅샷의 퍼블릭 공유를 방지 AWS 리전 하려는 각에서 구성해야 합니다. 선언적 정책을 사용하면 여러 리전과 여러 계정에 동시에 설정을 적용할 수 있습니다. 선언적 정책을 사용 중인 경우 계정 내에서 직접 설정을 수정할 수 없습니다. 이 주제에서는 계정 내에서 직접 설정을 구성하는 방법을 설명합니다. 선언적 정책 사용에 대한 자세한 내용은 AWS Organizations 사용 설명서의 선언적 정책을 참조하세요.

Console

스냅샷에 대한 퍼블릭 액세스 차단 구성

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 EC2 대시보드를 선택한 다음 계정 속성(오른쪽)에서 데이터 보호 및 보안을 선택합니다.
3. EBS 스냅샷에 대한 퍼블릭 액세스 차단 섹션에서 관리를 선택합니다.
4. 퍼블릭 액세스 차단을 선택한 후 다음 옵션 중 하나를 선택합니다.
   - 모든 퍼블릭 액세스 차단 - 스냅샷의 모든 퍼블릭 공유를 차단합니다. 계정의 사용자는 새 퍼블릭 공유를 요청할 수 없습니다. 또한 이미 공개적으로 공유된 스냅샷은 비공개로 취급되어 더 이상 공개적으로 사용할 수 없습니다.
   - 새 퍼블릭 공유 차단 - 스냅샷의 새 퍼블릭 공유만 차단합니다. 계정의 사용자는 새 퍼블릭 공유를 요청할 수 없습니다. 그러나 이미 공개적으로 공유된 스냅샷은 계속 공개적으로 사용할 수 있습니다.
5. 업데이트를 선택합니다.

AWS CLI

스냅샷에 대한 퍼블릭 액세스 차단 활성화 또는 수정

enable-snapshot-block-public-access 명령을 사용합니다. `--state`에 대해 다음 값 중 하나를 지정합니다.

- `block-all-sharing` - 스냅샷의 모든 퍼블릭 공유를 차단합니다. 계정의 사용자는 새 퍼블릭 공유를 요청할 수 없습니다. 또한 이미 공개적으로 공유된 스냅샷은 비공개로 취급되어 더 이상 공개적으로 사용할 수 없습니다.
- `block-new-sharing` - 스냅샷의 모든 퍼블릭 공유만 차단합니다. 계정의 사용자는 새 퍼블릭 공유를 요청할 수 없습니다. 그러나 이미 공개적으로 공유된 스냅샷은 계속 공개적으로 사용할 수 있습니다.

특정 리전의 스냅샷에 대한 퍼블릭 액세스 차단을 활성화 또는 수정하려면

```
aws ec2 enable-snapshot-block-public-access \
    --state block-new-sharing \
    --region us-east-1
```

다음은 예제 출력입니다.

```
{
    "State": "block-new-sharing"
}
```

모든 리전에서 스냅샷에 대한 퍼블릭 액세스 차단을 활성화 또는 수정하려면

```
echo -e "Region   \t Public Access State" ; \
echo -e "-------------- \t ----------------------" ; \
for region in $(
    aws ec2 describe-regions \
        --region us-east-1 \
        --query "Regions[*].[RegionName]" \
        --output text
    );
    do (output=$(
        aws ec2 enable-snapshot-block-public-access \
            --region $region \
            --state block-new-sharing \
            --output text)
        echo -e "$region \t $output"
    );
done
```

다음은 예제 출력입니다.

```
Region           Public Access State
--------------   ----------------------
ap-south-1       block-new-sharing
eu-north-1       block-new-sharing
eu-west-3        block-new-sharing
...
```

PowerShell

스냅샷에 대한 퍼블릭 액세스 차단 활성화 또는 수정

Enable-EC2SnapshotBlockPublicAccess 명령을 사용합니다. `-State`에 대해 다음 값 중 하나를 지정합니다.

- `block-all-sharing` - 스냅샷의 모든 퍼블릭 공유를 차단합니다. 계정의 사용자는 새 퍼블릭 공유를 요청할 수 없습니다. 또한 이미 공개적으로 공유된 스냅샷은 비공개로 취급되어 더 이상 공개적으로 사용할 수 없습니다.
- `block-new-sharing` - 스냅샷의 모든 퍼블릭 공유만 차단합니다. 계정의 사용자는 새 퍼블릭 공유를 요청할 수 없습니다. 그러나 이미 공개적으로 공유된 스냅샷은 계속 공개적으로 사용할 수 있습니다.

특정 리전의 스냅샷에 대한 퍼블릭 액세스 차단을 활성화 또는 수정하려면

```
Enable-EC2SnapshotBlockPublicAccess `
    -Region us-east-1 `
    -State block-new-sharing
```

다음은 예제 출력입니다.

```
Value
-----
block-new-sharing
```

모든 리전에서 스냅샷에 대한 퍼블릭 액세스 차단을 활성화 또는 수정하려면

```
(Get-EC2Region -Region us-east-1).RegionName | `
    ForEach-Object {
    [PSCustomObject]@{
        Region            = $_
        PublicAccessState = (
            Enable-EC2SnapshotBlockPublicAccess `
                -Region $_ `
                -State block-new-sharing)
    }
} | Format-Table -AutoSize
```

다음은 예제 출력입니다.

```
Region         PublicAccessState
------         -----------------
ap-south-1     block-new-sharing
eu-north-1     block-new-sharing
eu-west-3      block-new-sharing
...
```

Amazon EBS 스냅샷에 대한 퍼블릭 액세스 차단 설정 보기

퍼블릭 액세스 차단은 계정의 각 리전에 대해 다음 상태 중 하나일 수 있습니다.

- 모든 공유 차단 - 스냅샷의 모든 퍼블릭 공유가 차단됩니다. 계정의 사용자는 새 퍼블릭 공유를 요청할 수 없습니다. 또한 이미 공개적으로 공유된 스냅샷은 비공개로 취급되어 더 이상 공개적으로 사용할 수 없습니다.
- 새 공유 차단 - 스냅샷의 새로운 퍼블릭 공유만 차단됩니다. 계정의 사용자는 새 퍼블릭 공유를 요청할 수 없습니다. 그러나 이미 공개적으로 공유된 스냅샷은 계속 공개적으로 사용할 수 있습니다.
- 차단 해제됨 - 퍼블릭 공유가 차단되지 않습니다. 사용자가 공개적으로 스냅샷을 공유할 수 있습니다.

Console

스냅샷에 대한 퍼블릭 액세스 차단 설정 보기

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 EC2 대시보드를 선택한 다음 계정 속성(오른쪽)에서 데이터 보호 및 보안을 선택합니다.
3. EBS 스냅샷에 대한 퍼블릭 액세스 차단 섹션에 현재 설정이 표시됩니다.

AWS CLI

스냅샷에 대한 퍼블릭 액세스 차단 설정 보기

get-snapshot-block-public-access-state 명령을 사용합니다.

- 특정 리전

```
aws ec2 get-snapshot-block-public-access-state
```

이 예제 출력에서 `ManagedBy` 필드는 설정을 구성한 엔터티를 나타내고 `account`는 설정이 계정에서 직접 구성되었음을 나타냅니다. 값이 `declarative-policy`이면 설정이 선언적 정책에 의해 구성되었음을 의미합니다. 자세한 내용은 AWS Organizations 사용 설명서의 선언적 정책을 참조하세요.

```
{
```

```
    "State": "unblocked",
    "ManagedBy": "account"
}
```

- 모든 리전

```
echo -e "Region   \t Public Access State" ; \
echo -e "-------------- \t ----------------------" ; \
for region in $(
    aws ec2 describe-regions \
        --region us-east-1 \
        --query "Regions[*].[RegionName]" \
        --output text
    );
    do (output=$(
        aws ec2 get-snapshot-block-public-access-state \
            --region $region \
            --output text)
        echo -e "$region \t $output"
    );
done
```

다음은 예제 출력입니다.

```
Region           Public Access State
--------------   ----------------------
ap-south-1       unblocked
eu-north-1       unblocked
eu-west-3        unblocked
```

PowerShell

스냅샷에 대한 퍼블릭 액세스 차단 설정 보기

Get-EC2SnapshotBlockPublicAccessState cmdlet을 사용합니다.

- 특정 리전

```
Get-EC2SnapshotBlockPublicAccessState -Region us-east-1
```

다음은 예제 출력입니다.

```
Value
-----
block-new-sharing
```

- 모든 리전

```
(Get-EC2Region -Region us-east-1).RegionName | `
    ForEach-Object {
    [PSCustomObject]@{
        Region            = $_
        PublicAccessState = (Get-EC2SnapshotBlockPublicAccessState -Region $_)
    }
} | Format-Table -AutoSize
```

다음은 예제 출력입니다.

```
Region           Public Access State
--------------   ----------------------
ap-south-1       unblocked
eu-north-1       unblocked
eu-west-3        unblocked
...
```

Amazon EBS 스냅샷에 대한 퍼블릭 액세스 차단 비활성화

리전에서 스냅샷을 공개적으로 공유할 수 있도록 하려면 스냅샷에 대한 퍼블릭 액세스 차단을 비활성화하세요. 이 기능을 비활성화하면 사용자가 해당 리전에서 스냅샷을 공개적으로 공유할 수 있습니다.

> **Important**
>
> 모든 공유 차단 모드에서 스냅샷에 대한 퍼블릭 액세스 차단을 활성화해도 이미 공개적으로 공유된 스냅샷에 대한 권한은 변경되지 않습니다. 대신 이러한 스냅샷이 공개적으로 표시되거나 액세스할 수 없게 됩니다. 따라서 이러한 스냅샷의 속성은 스냅샷이 공개적으로 사용 불가능하더라도 여전히 공개적으로 공유됨을 나타냅니다.
> 퍼블릭 액세스 차단을 비활성화하면 이러한 스냅샷을 다시 공개적으로 사용할 수 있습니다.

> **Note**
>
> 이 설정은 계정 수준에서 직접 구성되거나 선언적 정책을 사용하여 구성됩니다. 스냅샷의 퍼블릭 공유를 허용 AWS 리전 하려는 각에서 구성해야 합니다. 선언적 정책을 사용하면 여러 리전과 여러 계정에 동시에 설정을 적용할 수 있습니다. 선언적 정책을 사용 중인 경우 계정 내에서 직접 설정을 수정할 수 없습니다. 이 주제에서는 계정 내에서 직접 설정을 구성하는 방법을 설명합니다. 선언적 정책 사용에 대한 자세한 내용은 AWS Organizations 사용 설명서의 선언적 정책을 참조하세요.

Console

스냅샷에 대한 퍼블릭 액세스 차단 비활성화

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 EC2 대시보드를 선택한 다음 계정 속성(오른쪽)에서 데이터 보호 및 보안을 선택합니다.
3. EBS 스냅샷에 대한 퍼블릭 액세스 차단 섹션에서 관리를 선택합니다.
4. 퍼블릭 액세스 차단을 선택 취소하고 업데이트를 선택합니다.

AWS CLI

스냅샷에 대한 퍼블릭 액세스 차단 비활성화

disable-snapshot-block-public-access 명령을 사용합니다.

- 특정 리전

```
aws ec2 disable-snapshot-block-public-access --region us-east-1
```

다음은 예제 출력입니다.

```
{
    "State": "unblocked"
}
```

- 모든 리전

```
echo -e "Region   \t Public Access State" ; \
```

```
echo -e "-------------- \t ----------------------" ; \
for region in $(
    aws ec2 describe-regions \
        --region us-east-1 \
        --query "Regions[*].[RegionName]" \
        --output text
    );
    do (output=$(
        aws ec2 disable-snapshot-block-public-access \
            --region $region \
            --output text)
        echo -e "$region \t $output"
    );
done
```

다음은 예제 출력입니다.

```
Region           Public Access State
--------------   ----------------------
ap-south-1       unblocked
eu-north-1       unblocked
eu-west-3        unblocked
```

PowerShell

스냅샷에 대한 퍼블릭 액세스 차단 비활성화

Disable-EC2SnapshotBlockPublicAccess cmdlet을 사용합니다.

- 특정 리전

```
Disable-EC2SnapshotBlockPublicAccess -Region us-east-1
```

다음은 예제 출력입니다.

```
Value
-----
unblocked
```

- 모든 리전

```
(Get-EC2Region -Region us-east-1).RegionName | `
    ForEach-Object {
    [PSCustomObject]@{
        Region            = $_
        PublicAccessState = (Disable-EC2SnapshotBlockPublicAccess -Region $_)
    }
} | `
Format-Table -AutoSize
```

다음은 예제 출력입니다.

```
Region         PublicAccessState
------         -----------------
ap-south-1     unblocked
eu-north-1     unblocked
eu-west-3      unblocked
...
```

EventBridge를 사용하여 Amazon EBS 스냅샷에 대한 퍼블릭 액세스 차단 모니터링

Amazon EBS는 스냅샷에 대한 퍼블릭 액세스 차단과 관련된 이벤트를 발생시킵니다. AWS Lambda 및 Amazon EventBridge를 사용하여 이벤트 알림을 프로그래밍 방식으로 처리할 수 있습니다. 이벤트는 최선의 작업을 기반으로 발생됩니다. 자세한 내용은 Amazon EventBridge 사용 설명서를 참조하세요.

다음과 같은 이벤트가 발생합니다.

- 모든 공유 차단 모드에서 스냅샷에 대한 퍼블릭 액세스 차단 활성화

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "EBS Snapshot Block Public Access Enabled",
  "source": "aws.ec2",
  "account": "123456789012",
  "time": "2019-05-31T21:49:54Z",
  "region": "us-east-1",
  "detail": {
```

```
    "SnapshotBlockPublicAccessState": "block-all-sharing",
    "message": "Block Public Access was successfully enabled in 'block-all-sharing'
 mode"
  }
}
```

- 새 공유 차단 모드에서 스냅샷에 대한 퍼블릭 액세스 차단 활성화

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "EBS Snapshot Block Public Access Enabled",
  "source": "aws.ec2",
  "account": "123456789012",
  "time": "2019-05-31T21:49:54Z",
  "region": "us-east-1",
  "detail": {
    "SnapshotBlockPublicAccessState": "block-new-sharing",
    "message": "Block Public Access was successfully enabled in 'block-new-sharing'
 mode"
  }
}
```

- 스냅샷에 대한 퍼블릭 액세스 차단 비활성화

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "EBS Snapshot Block Public Access Disabled",
  "source": "aws.ec2",
  "account": "123456789012",
  "time": "2019-05-31T21:49:54Z",
  "region": "us-east-1",
  "detail": {
    "SnapshotBlockPublicAccessState": "unblocked",
    "message": "Block Public Access was successfully disabled"
  }
}
```

Outposts의 Amazon EBS 로컬 스냅샷

Amazon EBS 스냅샷은 EBS 볼륨의 특정 시점 복사본입니다.

기본적으로 AWS Outpost의 EBS 볼륨에 대한 스냅샷은 Outpost의 리전에 있는 Amazon S3에 저장됩니다. 또한 Outpost의 Amazon EBS 로컬 스냅샷을 사용하여 Outpost의 볼륨 스냅샷을 Outpost 자체의 Amazon S3에 로컬로 저장할 수 있습니다. 이를 통해 스냅샷 데이터가 Outpost와 프레미스에 상주합니다. 또한 AWS Identity and Access Management (IAM) 정책 및 권한을 사용하여 데이터 레지던시 적용 정책을 설정하여 스냅샷 데이터가를 벗어나지 않도록 할 수 있습니다Outpost. 이는 아직 리전에서 서비스를 제공하지 않고 데이터 레지던시 요구 사항이 있는 국가 또는 AWS 리전에 상주하는 경우에 특히 유용합니다.

이 주제는 Outposts의 Amazon EBS 로컬 스냅샷 사용에 관한 정보를 제공합니다. Amazon EBS 스냅샷 및 AWS 리전의 스냅샷 작업에 대한 자세한 내용은 섹션을 참조하세요Amazon EBS 스냅샷.

자세한 내용은 AWS Outposts 패밀리 및 AWS Outposts 패밀리 설명서를 참조하세요.

주제

- 자주 묻는 질문(FAQ)
- 사전 조건
- 고려 사항
- IAM을 통한 액세스 제어
- 로컬 스냅샷 작업

자주 묻는 질문(FAQ)

1. 로컬 스냅샷은 무엇입니까?

   기본적으로 Outpost의 볼륨에 대한 Amazon EBS 스냅샷은 Outpost의 리전에 있는 Amazon S3에 저장됩니다. Outpost가 S3 on Outposts와 함께 프로비저닝된 경우 Outpost 자체에 로컬로 스냅샷을 저장하도록 선택할 수 있습니다. 로컬 스냅샷은 최근 스냅샷이 저장된 이후 변경된 볼륨의 블록만 저장되는 증분 백업입니다. 이러한 스냅샷을 사용하여 언제든지 스냅샷과 동일한 Outpost로 볼륨을 복원할 수 있습니다. Amazon EBS 스냅샷 복사에 대한 자세한 내용은 Amazon EBS 스냅샷 섹션을 참조하세요.

2. 왜 로컬 스냅샷을 사용해야 합니까?

   스냅샷은 편리한 데이터 백업 방법입니다. 로컬 스냅샷을 사용하면 모든 스냅샷 데이터가 Outpost에 로컬로 저장됩니다. 따라서 프레미스를 벗어나지 않습니다. 이는 아직 리전에서 서비스를 제공하지 않고 레지던시 요구 사항이 있는 국가 또는 AWS 리전에 상주하는 경우에 특히 유용합니다.

   추가로 로컬 스냅샷을 사용하면 대역폭 제한 환경에서 리전과 Outpost 사이의 통신에 사용되는 대역폭을 줄일 수 있습니다.

3. Outpost에 스냅샷 데이터 레지던시를 어떻게 적용합니까?

   AWS Identity and Access Management (IAM) 정책을 사용하여 로컬 스냅샷으로 작업할 때 보안 주체(AWS 계정, IAM 사용자 및 IAM 역할)가 갖는 권한을 제어하고 데이터 레지던시를 적용할 수 있습니다. 보안 주체가 Outpost 볼륨 및 인스턴스에서 스냅샷을 생성하고 AWS 리전에 스냅샷을 저장하지 못하도록 하는 정책을 생성할 수 있습니다. 현재 Outpost에서 리전으로 스냅샷 및 이미지를 복사하는 것은 지원되지 않습니다. 자세한 내용은 IAM을 통한 액세스 제어 섹션을 참조하세요.

4. 장애 시 일관성이 있는 다중 볼륨 로컬 스냅샷이 지원됩니까?

   예, Outpost의 인스턴스에서 장애 시 일관성이 있는 다중 볼륨 로컬 스냅샷 생성이 가능합니다.

5. 로컬 스냅샷을 어떻게 생성합니까?

   AWS Command Line Interface (AWS CLI) 또는 Amazon EC2 콘솔을 사용하여 스냅샷을 수동으로 생성할 수 있습니다. 자세한 내용은 단원을 참조하십시오로컬 스냅샷 작업. Amazon Data Lifecycle Manager 사용을 통해 로컬 스냅샷의 수명 주기를 자동화할 수도 있습니다. 자세한 내용은 Outpost에서 스냅샷 자동화 섹션을 참조하세요.

6. Outpost와 리전과의 연결이 끊어지면 로컬 스냅샷을 생성, 사용 또는 삭제할 수 있습니까?

   아니요, 리전에서 액세스, 권한 부여, 로깅 및 스냅샷 상태에 있어 중요한 모니터링 서비스를 제공하므로 Outpost는 리전에 연결되어야 합니다. 연결이 끊어진 경우 새 로컬 스냅샷 생성, 기존 로컬 스냅샷에서 볼륨 생성이나 인스턴시 시작, 또는 로컬 스냅샷 삭제를 수행할 수 없습니다.

7. 로컬 스냅샷 삭제 후 Amazon S3 스토리지 용량은 얼마나 빠르게 사용 가능합니까?

   Amazon S3 스토리지 용량은 로컬 스냅샷과 이를 참조하는 볼륨 삭제 후 72시간 내에 사용 가능합니다.

8. 에서 Amazon S3 용량이 부족하지 않도록 하려면 어떻게 해야 하나요Outpost?

   Amazon CloudWatch 경보를 사용하여 Amazon S3 스토리지 용량을 모니터링하고, 스토리지 용량이 부족하지 않도록 더 이상 필요하지 않은 스냅샷과 볼륨을 삭제하는 것이 좋습니다. Amazon Data Lifecycle Manager를 사용하여 로컬 스냅샷의 수명 주기를 자동화하는 경우 스냅샷 보존 정책이 스냅샷을 필요 이상으로 오래 보존하지 않도록 해야 합니다.

9. Outpost에서 로컬 Amazon S3 용량이 부족하면 어떻게 됩니까?

   Outpost에서 로컬 Amazon S3 용량이 부족하면 Amazon Data Lifecycle Manager는 Outpost에서 로컬 스냅샷을 성공적으로 생성할 수 없습니다. Amazon Data Lifecycle Manager는 Outpost에서 로컬 스냅샷을 생성하려고 시도하지만 스냅샷은 즉시 `error` 상태로 전환되고 결국 Amazon Data Lifecycle Manager에 의해 삭제됩니다. 스냅샷 생성 실패에 대한 스냅샷 수명 주기 정책을 모니터링

하는 SnapshotsCreateFailed Amazon CloudWatch 지표를 사용하는 것이 좋습니다. 자세한 내용은 CloudWatch를 사용하여 Data Lifecycle Manager 정책 모니터링 섹션을 참조하세요.

10. 스팟 인스턴스 및 스팟 플릿에 로컬 스냅샷 및 로컬 스냅샷 기반 AMI를 사용할 수 있습니까?

    아니요 로컬 스냅샷 또는 로컬 스냅샷 기반 AMI를 사용하여 스팟 인스턴스나 스팟 플릿을 시작할 수 없습니다.

11. Amazon EC2 Auto Scaling에 로컬 스냅샷 및 로컬 스냅샷 기반 AMI를 사용할 수 있습니까?

    예,로컬 스냅샷 및 로컬 스냅샷 기반 AMI를 사용하여 스냅샷과 동일한 Outpost에 있는 서브넷의 Auto Scaling 그룹을 시작할 수 있습니다. Amazon EC2 Auto Scaling 그룹 서비스 연결 역할에는 스냅샷을 암호화하는 데 사용되는 KMS 키 암호화 키를 사용할 권한이 있어야 합니다.

    로컬 스냅샷 또는 로컬 스냅샷이 지원하는 AMIs 사용하여 AWS 리전에서 Auto Scaling 그룹을 시작할 수 없습니다.

사전 조건

Outpost에 스냅샷을 저장하려면 Outpost의 S3가 프로비저닝된 Outpost가 있어야 합니다. S3 on Outposts에 대한 자세한 내용은 Amazon S3 on Outposts 사용 설명서의 S3 on Outposts를 참조하세요.

고려 사항

로컬 스냅샷 작업 시 다음 사항에 유의하세요.

- 로컬 스냅샷을 사용하려면가 해당 AWS 리전에 연결되어 Outpost 있어야 합니다.
- 스냅샷 메타데이터는와 연결된 AWS 리전에 저장됩니다Outpost. 여기에는 스냅샷 데이터가 포함되지 않습니다.
- Outpost에 저장된 스냅샷은 기본적으로 암호화됩니다. 암호화되지 않은 스냅샷은 지원되지 않습니다. 로컬 스냅샷Outpost에서에 볼륨을 생성하면 볼륨은 소스 스냅샷과 동일한 KMS 키를 사용하여 암호화됩니다. 그러나 로컬 스냅샷을 복사할 때 암호화 키를 변경할 수 있습니다.
- Outpost에서 로컬 스냅샷을 삭제한 이후 삭제된 스냅샷에서 사용하는 Amazon S3 스토리지 용량은 72시간 내에 사용 가능하게 됩니다. 자세한 내용은 로컬 스냅샷 삭제 섹션을 참조하세요.
- Outpost에서 로컬 스냅샷을 내보낼 수 없습니다.
- 로컬 스냅샷에 대해 빠른 스냅샷 복원을 활성화할 수 없습니다.
- EBS 다이렉트 API의 경우 로컬 스냅샷에서 지원되지 않습니다.

- 에서 리전으로, 한 리전에서 다른 AWS 리전Outpost으로 또는 내에서 로컬 스냅샷 또는 AMIs를 복사 Outpost할 수 없습니다Outpost. 하지만 AWS 리전에서 Outpost로 스냅샷을 복사할 수 있습니다. 자세한 내용은 AWS 리전에서 로 스냅샷 복사 Outpost 섹션을 참조하세요.
- AWS 리전에서 로 스냅샷을 복사할 때 Outpost데이터는 서비스 링크를 통해 전송됩니다. 여러 스냅샷을 동시에 복사하면 Outpost에서 실행 중인 다른 서비스에 영향을 미칠 수 있습니다.
- 로컬 스냅샷을 공유할 수 없습니다.
- IAM 정책을 사용하여 데이터 레지던시 요구 사항이 충족되는지 확인해야 합니다. 자세한 내용은 IAM을 통한 액세스 제어 섹션을 참조하세요.
- 로컬 스냅샷은 증분 백업입니다. 가장 최근의 스냅샷 이후에 변경된 볼륨의 블록만 저장됩니다. 각 로컬 스냅샷에는 (스냅샷을 만든 시점의) 데이터를 새 EBS 볼륨에 복원하는 데 필요한 모든 정보가 들어 있습니다. 자세한 내용은 Amazon EBS 스냅샷 작동 방식 섹션을 참조하세요.
- IAM 정책을 사용하여 CopySnapshot 및 CopyImage 작업에 대한 데이터 레지던시를 적용할 수 없습니다.

IAM을 통한 액세스 제어

AWS Identity and Access Management (IAM) 정책을 사용하여 로컬 스냅샷으로 작업할 때 보안 주체(AWS 계정, IAM 사용자 및 IAM 역할)가 갖는 권한을 제어할 수 있습니다. 다음은 로컬 스냅샷에 대한 특정 작업을 수행할 권한을 부여 또는 거부하는 데 사용할 수 있는 정책의 예시입니다.

> **Important**
>
> Outpost에서 리전으로 스냅샷 및 이미지를 복사하는 것은 현재 지원되지 않습니다. 따라서 현재 IAM 정책을 사용하여 CopySnapshot 및 CopyImage 작업에 대한 데이터 레지던시를 적용할 수 없습니다.

주제

- 스냅샷에 대한 데이터 레지던시 적용
- 보안 주체의 로컬 스냅샷 삭제 방지

스냅샷에 대한 데이터 레지던시 적용

다음 예제 정책은 모든 보안 주체가의 볼륨 및 인스턴스에서 스냅샷을 생성하고 AWS 리전에 스냅샷 데이터를 Outpost `arn:aws:outposts:us-east-1:123456789012:outpost/`

op-1234567890abcdef 저장하지 못하도록 합니다. 보안 주체는 로컬 스냅샷을 생성할 수 있습니다. 이 정책을 통해 모든 스냅샷이 Outpost에 유지됩니다.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": [
                "ec2:CreateSnapshot",
                "ec2:CreateSnapshots"
            ],
            "Resource": "arn:aws:ec2:us-east-1::snapshot/*",
            "Condition": {
                "StringEquals": {
                    "ec2:SourceOutpostArn": "arn:aws:outposts:us-
east-1:123456789012:outpost/op-1234567890abcdef0"
                },
                "Null": {
                    "ec2:OutpostArn": "true"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateSnapshot",
                "ec2:CreateSnapshots"
            ],
            "Resource": "*"
        }
    ]
}
```

보안 주체의 로컬 스냅샷 삭제 방지

다음 정책 예제는 모든 위탁자가 Outpost `arn:aws:outposts:us-east-1:123456789012:outpost/op-1234567890abcdef0`에 저장된 로컬 스냅샷을 삭제하지 못하도록 합니다.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": [
                "ec2:DeleteSnapshot"
            ],
            "Resource": "arn:aws:ec2:us-east-1::snapshot/*",
            "Condition": {
                "StringEquals": {
                    "ec2:OutpostArn": "arn:aws:outposts:us-east-1:123456789012:outpost/op-1234567890abcdef0"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DeleteSnapshot"
            ],
            "Resource": "*"
        }
    ]
}
```

로컬 스냅샷 작업

다음 섹션에서는 로컬 스냅샷 사용 방법을 설명합니다.

주제

- 스냅샷 저장 규칙

- Outpost의 볼륨에서 로컬 스냅샷 생성
- 로컬 스냅샷에서 AMI 생성
- AWS 리전에서 로 스냅샷 복사 Outpost
- AWS 리전에서 로 AMIs 복사 Outpost
- 로컬 스냅샷에서 볼륨 생성
- 로컬 스냅샷 기반 AMI에서 인스턴스 시작
- 로컬 스냅샷 삭제
- Outpost에서 스냅샷 자동화

스냅샷 저장 규칙

스냅샷 스토리지에 다음 규칙이 적용됩니다.

- 볼륨의 최신 스냅샷이 Outpost에 저장된 경우 모든 후속 스냅샷은 동일한 Outpost에 저장되어야 합니다.
- 볼륨의 최신 스냅샷이 AWS 리전에 저장되는 경우 모든 연속 스냅샷은 동일한 리전에 저장되어야 합니다. 해당 볼륨에서 로컬 스냅샷 생성을 시작하려면 다음을 수행합니다.

  1. AWS 리전에서 볼륨의 스냅샷을 생성합니다.
  2. AWS 리전Outpost에서 스냅샷을에 복사합니다.
  3. 로컬 스냅샷에서 새 볼륨을 생성합니다.
  4. Outpost의 인스턴스에 볼륨을 연결합니다.

  Outpost의 새 볼륨의 경우 다음 스냅샷은 Outpost 또는 AWS 리전에 저장할 수 있습니다. 이후 모든 후속 스냅샷은 동일한 위치에 저장되어야 합니다.
- 에서 생성된 스냅샷Outpost과 AWS 리전Outpost에서 로 복사된 스냅샷을 포함한 로컬 스냅샷은 동일한에서 볼륨을 생성하는 데에만 사용할 수 있습니다Outpost.
- 한 리전의 스냅샷에 있는 Outpost에서 볼륨을 생성한 경우 새 볼륨의 모든 후속 스냅샷은 동일한 리전에 있어야 합니다.
- 로컬 스냅샷의 Outpost에서 볼륨을 생성한 경우 새 볼륨의 모든 후속 스냅샷은 동일한 Outpost에 있어야 합니다.

Outpost의 볼륨에서 로컬 스냅샷 생성

Outpost의 볼륨에서 로컬 스냅샷을 생성할 수 있습니다. 동일한 Outpost의 스냅샷을 소스 볼륨으로 저장하거나 Outpost에 대한 리전에 저장할 수 있습니다.

로컬 스냅샷은 동일한 Outpost에서 볼륨을 생성하는 데에만 사용할 수 있습니다.

자세한 내용은 Amazon EBS 스냅샷 생성 섹션을 참조하세요.

로컬 스냅샷에서 AMI 생성

Outpost의 리전에 저장된 로컬 스냅샷과 스냅샷의 조합을 사용하여 Amazon Machine Image(AMI)를 생성할 수 있습니다. 예를 들어, `us-east-1`에 Outpost가 있는 경우 해당 Outpost의 로컬 스냅샷 기반, 그리고 `us-east-1` 리전의 스냅샷 기반의 데이터 볼륨이 포함된 AMI를 생성할 수 있습니다.

로컬 스냅샷을 Outpost 지원하는의 인스턴스에서 직접 AMI를 생성할 수도 있습니다. AMI에 대해 생성된 백업 스냅샷은 인스턴스Outpost와 동일한 또는 상위 리전에 저장할 수 있습니다.

> **Note**
>
> - 여러 Outposts에서 저장된 백업 스냅샷을 포함한 AMI를 생성할 수 있습니다.
> - 로컬 스냅샷 기반 AMI를 사용하여 동일한 Outpost에서만 인스턴스를 시작할 수 있습니다.

리전의 스냅샷에서 Outpost에 있는 AMI를 생성하려면

1. 리전에서 Outpost로 스냅샷을 복사합니다. 자세한 내용은 AWS 리전에서 로 스냅샷 복사 Outpost 섹션을 참조하세요.
2. Amazon EC2 콘솔 또는 register-image 명령을 사용하여 Outpost의 스냅샷 복사본을 사용하는 AMI를 생성합니다. 자세한 내용은 스냅샷에서 AMI 생성을 참조하십시오.

의 인스턴스에서 직접 AMI를 생성하려면 Outpost

create-image 명령을 `--snapshot-location` 파라미터와 함께 사용하여 백업 스냅샷이 저장되는 위치를 제어합니다.

- 스냅샷을 인스턴스Outpost와 동일한에 로컬로 저장하려면를 지정합니다`local`.
- 의 상위 리전에 스냅샷을 저장하려면를 Outpost지정합니다`regional`.

소스 인스턴스가 로컬 스냅샷을 지원하는에 Outpost 있는 경우 `--snapshot-location` 파라미터가 필요합니다. 생략하면 요청이 실패하고 `InvalidParameterValue` 오류가 발생합니다.

다음 예제에서는 AMI를 생성하고 Outpost 인스턴스와 동일한에 백업 스냅샷을 저장합니다.

```
aws ec2 create-image \
    --instance-id i-1234567890abcdef0 \
    --name "My Outpost image" \
    --snapshot-location local
```

Outpost의 인스턴스에서 리전에 있는 AMI를 생성하려면

create-image 명령을 사용하고 `--snapshot-location` 파라미터`regional`에를 지정합니다.

다음 예제에서는 AMI를 생성하고 백업 스냅샷을의 상위 리전에 저장합니다Outpost.

```
aws ec2 create-image \
    --instance-id i-1234567890abcdef0 \
    --name "My Regional image" \
    --snapshot-location regional
```

AWS 리전에서 로 스냅샷 복사 Outpost

AWS 리전에서 로 스냅샷을 복사할 수 있습니다Outpost. 스냅샷이 Outpost의 리전에 있는 경우에만 이를 수행할 수 있습니다. 스냅샷이 다른 리전에 있는 경우 우선 Outpost에 대한 리전으로 스냅샷을 복사한 다음 해당 리전에서 Outpost로 복사해야 합니다.

> **Note**
>
> Outpost에서 리전으로, 하나의 Outpost에서 다른 Outpost로, 또는 동일한 Outpost 내에서 로컬 스냅샷을 복사할 수 없습니다.

자세한 내용은 Amazon EBS 스냅샷 복사 섹션을 참조하세요.

AWS 리전에서 로 AMIs 복사 Outpost

AWS 리전에서 로 AMIs를 복사할 수 있습니다Outpost. 리전에서 Outpost로 AMI를 복사하면 AMI에 연결된 모든 스냅샷이 리전에서 Outpost로 복사됩니다.

AMI와 연결된 스냅샷이 Outpost에 대한 리전에 있는 경우에만 리전에서 Outpost로 AMI를 복사할 수 있습니다. 스냅샷이 다른 리전에 있는 경우 우선 Outpost에 대한 리전으로 AMI를 복사한 다음 해당 리전에서 Outpost로 복사해야 합니다.

> **Note**
>
> Outpost에서 리전으로, 하나의 Outpost에서 다른 Outpost로, 또는 Outpost 내에서 AMI를 복사할 수 없습니다.

copy-image AWS CLI 명령만 Outpost 사용하여 리전에서 로 AMIs를 복사할 수 있습니다.

로컬 스냅샷에서 볼륨 생성

로컬 스냅샷에서 Outpost에 있는 볼륨을 생성할 수 있습니다. 볼륨은 소스 스냅샷과 동일한 Outpost에 생성되어야 합니다. 로컬 스냅샷을 사용하여 Outpost에 대한 리전에서 볼륨을 생성할 수 없습니다.

로컬 스냅샷에서 볼륨을 생성할 때 다른 KMS 키를 사용하여 볼륨을 다시 암호화할 수 없습니다. 로컬 스냅샷에서 생성된 볼륨은 소스 스냅샷과 동일한 KMS 키를 사용하여 암호화되어야 합니다.

자세한 내용은 Amazon EBS 볼륨 생성 섹션을 참조하세요.

로컬 스냅샷 기반 AMI에서 인스턴스 시작

로컬 스냅샷 기반 AMI에서 인스턴스를 시작할 수 있습니다. 소스 AMI와 동일한 Outpost에 있는 인스턴스를 시작해야 합니다. 자세한 내용은 AWS Outposts 사용 설명서의 Outpost에서 인스턴스 시작을 참조하세요.

로컬 스냅샷 삭제

Outpost에서 로컬 스냅샷을 삭제할 수 있습니다. Outpost에서 스냅샷을 삭제한 이후 삭제된 스냅샷에서 사용한 Amazon S3 스토리지 용량은 스냅샷과 해당 스냅샷을 참조하는 볼륨 삭제 후 72시간 내에 사용 가능하게 됩니다.

Amazon S3 스토리지 용량을 즉시 사용할 수 없게 된 경우 Amazon CloudWatch 경보를 사용하여 Amazon S3 스토리지 용량을 모니터링하는 것이 좋습니다. 스토리지 용량이 부족하지 않도록 더 이상 필요하지 않은 스냅샷과 볼륨을 삭제하세요.

스냅샷 삭제에 대한 자세한 내용은 스냅샷 삭제 섹션을 참조하세요.

Outpost에서 스냅샷 자동화

Outpost에서 볼륨 및 인스턴스의 스냅샷을 자동으로 생성, 복사, 유지 및 삭제하는 Amazon Data Lifecycle Manager 스냅샷 수명 주기 정책을 생성할 수 있습니다. 스냅샷을 리전에 저장할지 여부 또는 Outpost에 로컬로 저장할지 여부를 선택할 수 있습니다. 또한 AWS 리전에서 생성 및 저장된 스냅샷을 에 자동으로 복사할 수 있습니다Outpost.

다음 표는 지원되는 기능의 개요를 제공합니다.

리소스 위치	스냅샷 대상	리전 간 리전 복사	에 리전 간 복사 Outpost	빠른 스냅샷 복원	교차 계정 공유
리전	리전	✓	✓	✓	✓
Outpost	리전	✓	✓	✓	✓
Outpost	Outpost	✗	✗	✗	✗

고려 사항

- 현재 Amazon EBS 스냅샷 수명 주기 정책만 지원됩니다. EBS 기반 AMI 정책 및 교차 계정 공유 이벤트 정책은 지원되지 않습니다.
- 정책이 리전의 볼륨 또는 인스턴스에 대한 스냅샷을 관리하는 경우 스냅샷은 소스 리소스와 동일한 리전에 생성됩니다.
- 정책이 Outpost의 볼륨 또는 인스턴스에 대한 스냅샷을 관리하는 경우 소스 Outpost, 또는 해당 Outpost의 리전에서 스냅샷을 생성할 수 있습니다.
- 단일 정책은 한 리전의 스냅샷과 Outpost의 스냅샷을 모두 관리할 수 없습니다. 리전과 Outpost의 스냅샷을 자동화해야 하는 경우 별도의 정책을 생성해야 합니다.
- 빠른 스냅샷 복원은 Outpost에서 생성된 스냅샷이나 Outpost에 복사된 스냅샷에 대해 지원되지 않습니다.
- Outpost에서 생성된 스냅샷에 대해 교차 계정 공유는 지원되지 않습니다.
- 에서 생성된 스냅샷에는 사전 스크립트와 사후 스크립트가 지원되지 않습니다Outpost.

로컬 스냅샷을 관리하는 스냅샷 수명 주기 생성에 대한 자세한 내용은 스냅샷 수명 주기 자동화를 참조하십시오.

로컬 영역의 로컬 스냅샷

Amazon EBS 스냅샷은 EBS 볼륨의 특점 시점 복사본입니다.

AWS 로컬 영역의 EBS 볼륨 스냅샷은 동일한 로컬 영역의 Amazon S3 또는 해당 로컬 영역의 상위 리전에 저장할 수 있습니다. 로컬 영역에 스냅샷을 저장하면 스냅샷 데이터가 특정 국가, 주 또는 지방 자치제에서 처리 및 저장되도록 하여 데이터 레지던시 요구 사항을 충족하는 데 도움이 됩니다. 또한 IAM을 사용하여 데이터 레지던시 적용 정책을 설정하여 스냅샷 데이터가 로컬 영역을 벗어나지 않도록 할 수 있습니다.

로컬 영역은 AWS 인프라를 최종 사용자 및 비즈니스 센터에 더 가깝게 가져와서 10밀리초 미만의 지연 시간 또는 로컬 데이터 처리가 필요한 애플리케이션에 적합합니다. 또한 규제 및 규정 준수에 민감한 워크로드에 대한 데이터 레지던시 요구 사항을 충족할 수 있습니다. 자세한 내용은 AWS 로컬 영역이란 무엇입니까?를 참조하십시오.

현재 로컬 스냅샷은 Amazon S3를 지원하는 로컬 영역에서 지원됩니다. 자세한 내용은 AWS 로컬 영역을 참조하세요.

주제

- 자주 묻는 질문(FAQ)
- 고려 사항
- IAM을 통한 액세스 제어

자주 묻는 질문(FAQ)

1. 로컬 영역의 로컬 스냅샷이란 무엇입니까?

   로컬 영역의 로컬 스냅샷은 로컬 영역의 Amazon S3에 저장된 스냅샷입니다. 기본적으로 로컬 영역에 있는 Amazon EBS 볼륨의 스냅샷은 상위 리전에 있는 Amazon S3에 저장됩니다. 로컬 영역이 Amazon S3를 지원하는 경우 스냅샷을 로컬 영역에 로컬로 저장하도록 선택할 수 있습니다. AWS 리전의 스냅샷과 마찬가지로 로컬 영역의 로컬 스냅샷은 증분식입니다. 즉, 가장 최근 스냅샷 이후에 변경된 볼륨의 블록만 저장됩니다. 이러한 스냅샷을 사용하여 언제든지 동일한 로컬 영역에서 Amazon EBS 볼륨을 복원할 수 있습니다.

2. 왜 로컬 스냅샷을 사용해야 합니까?

   로컬 영역의 로컬 스냅샷을 사용하면 스냅샷 데이터가 국가, 주 또는 지방 자치제와 같은 특정 지리적 위치에 상주하도록 하여 데이터 레지던시 또는 데이터 격리 요구 사항을 충족할 수 있습니다.

3. 로컬 영역에 스냅샷 데이터 레지던시를 어떻게 적용합니까?

   AWS Identity and Access Management (IAM) 정책을 사용하여 로컬 영역의 로컬 스냅샷으로 작업할 때 보안 주체(AWS 계정, IAM 사용자 및 IAM 역할)가 갖는 권한을 제어하고 데이터 레지던시를 적용할 수 있습니다. 예를 들어 사용자가 로컬 영역의 볼륨에서 스냅샷을 생성하고 해당 스냅샷을 AWS 리전에 저장하지 못하도록 하는 정책을 생성할 수 있습니다. 자세한 내용은 IAM을 통한 액세스 제어 섹션을 참조하세요.

4. 장애 시 일관성이 있는 다중 볼륨 로컬 스냅샷이 지원됩니까?

   예, 로컬 영역의 인스턴스에서 장애 시 일관성이 있는 다중 볼륨 로컬 스냅샷을 생성할 수 있습니다.

5. 로컬 영역의 로컬 스냅샷을 어떻게 생성합니까?

   AWS CLI 또는 Amazon EC2 콘솔을 사용하여 로컬 영역에서 로컬 스냅샷을 수동으로 생성할 수 있습니다. 자세한 내용은 단원을 참조하십시오EBS 볼륨의 스냅샷 생성. Amazon Data Lifecycle Manager를 사용하여 로컬 영역의 로컬 스냅샷의 수명 주기를 자동화할 수도 있습니다. 자세한 내용은 단원을 참조하십시오EBS 스냅샷에 대한 Amazon Data Lifecycle Manager 사용자 지정 정책 생성.

6. 로컬 영역의 로컬 스냅샷을 복사할 수 있습니까?

   예, 리전에서 동일한 리전의 로컬 영역으로, 로컬 영역에서 해당 리전으로, 한 로컬 영역에서 동일한 로컬 영역 그룹의 다른 로컬 영역으로 스냅샷을 복사할 수 있습니다.

7. 로컬 영역의 로컬 스냅샷에서 데이터를 복원하려면 어떻게 해야 합니까?

   로컬 영역의 로컬 스냅샷을 사용하여 동일한 로컬 영역에만 Amazon EBS 볼륨을 생성할 수 있습니다.

8. 로컬 영역의 로컬 스냅샷은 어떻게 암호화됩니까?

   로컬 영역별 암호화 동작로컬 영역, Outpost 및 Wavelength 영역의 Amazon EBS 암호화은 섹션을 참조하세요. Amazon EBS 리소스가 기본적으로 암호화되는 로컬 영역에서 로컬 스냅샷에서 생성된 볼륨은 항상 스냅샷과 동일한 AWS KMS 키를 사용하여 암호화됩니다. 그러나 로컬 스냅샷을 복사할 때 암호화 키를 변경할 수 있습니다.

9. 로컬 영역의 로컬 스냅샷을 사용하여 EBS 지원 AMI를 생성할 수 있습니까?

   예, 스냅샷 대상을 로컬 영역으로 지정하여 EBS 지원 AMI를 생성할 때 로컬 영역의 로컬 스냅샷을 사용할 수 있습니다. 자세한 내용을 알아보려면 Amazon EBS 지원 Linux AMI 생성을 참조하세요.

10. 로컬 영역의 로컬 스냅샷을 공유할 수 있습니까?

    예, 로컬 영역의 로컬 스냅샷을 AWS 해당 계정에서 사용할 수 있도록 로컬 영역을 활성화한 다른 계정과 공유할 수 있습니다.

11. 볼륨의 로컬 스냅샷을 생성한 다음 상위 리전의 스냅샷 생성으로 전환할 수 있습니까?

    아니요, 볼륨의 로컬 스냅샷을 생성한 후에는 상위 리전에서 해당 볼륨의 후속 스냅샷을 생성할 수 없습니다. 모든 스냅샷은 증분식이므로 볼륨의 최신 스냅샷이 로컬 스냅샷인 경우 해당 볼륨의 모든 후속 스냅샷은 로컬 스냅샷이어야 합니다.

고려 사항

로컬 영역의 로컬 스냅샷 작업 시 다음 사항에 유의하세요.

- 현재 로컬 스냅샷은 Amazon S3를 지원하는 로컬 영역에서 지원됩니다.
- 다음 기능은 로컬 영역의 로컬 스냅샷에 사용할 수 없습니다.
  - VM Import/Export 작업
  - 빠른 스냅샷 복원
  - EBS 다이렉트 API
  - 휴지통
  - 스냅샷 아카이브
- IAM 정책을 사용하여 데이터 레지던시 요구 사항을 충족해야 합니다. 자세한 내용은 IAM을 통한 액세스 제어 섹션을 참조하세요.
- 볼륨의 최신 스냅샷이 로컬 스냅샷인 경우 모든 후속 스냅샷은 로컬 스냅샷이어야 합니다. 마찬가지로 볼륨의 최신 스냅샷이 AWS 리전에 저장되는 경우 모든 연속 스냅샷은 동일한 리전에 저장되어야 합니다.

IAM을 통한 액세스 제어

AWS Identity and Access Management (IAM) 정책을 사용하여 로컬 영역에서 로컬 스냅샷으로 작업할 때 보안 주체(AWS 계정, IAM 사용자 및 IAM 역할)가 갖는 권한을 제어할 수 있습니다. 다음은 로컬 영역의 로컬 스냅샷에 대한 특정 작업을 수행할 권한을 부여 또는 거부하는 데 사용할 수 있는 정책의 예제입니다.

주제

- 로컬 영역의 로컬 스냅샷에 대한 데이터 레지던시 적용
- 로컬 영역의 로컬 스냅샷 공유 방지
- 위탁자의 로컬 영역의 로컬 스냅샷 삭제 방지

로컬 영역의 로컬 스냅샷에 대한 데이터 레지던시 적용

다음 예제 정책은 사용자가 로컬 영역의 볼륨 및 인스턴스에서 로컬 영역의 로컬 스냅샷만 생성하도록 제한합니다. 이는 사용자가 로컬 영역의 볼륨 및 인스턴스에서 리전의 스냅샷을 생성하는 것을 방지합니다.

JSON

```
{
    "Version":"2012-10-17",
    "Statement":
    [
        {
            "Effect": "Deny",
            "Action":
            [
                "ec2:CreateSnapshot",
                "ec2:CreateSnapshots"
            ],
            "Resource": "*",
            "Condition":
            {
                "StringEquals":
                {
                    "ec2:Location": "regional",
                    "ec2:SourceAvailabilityZone": "local_zone"
                }
            }
        },
        {
            "Effect": "Deny",
            "Action":
            [
                "ec2:CreateSnapshot",
                "ec2:CreateSnapshots"
            ],
            "Resource": "*",
            "Condition":
            {
                "StringEquals":
                {
                    "ec2:SourceAvailabilityZone": "local_zone"
```

```
                },
                "Null":
                {
                    "ec2:Location": "true"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action":
            [
                "ec2:CreateSnapshot",
                "ec2:CreateSnapshots"
            ],
            "Resource": "*"
        }
    ]
}
```

로컬 영역의 로컬 스냅샷 공유 방지

다음 예제 정책은 모든 사용자가 로컬 영역의 로컬 스냅샷을 공유하지 못하도록 합니다.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "DenySnapshotModifyInLocalZone",
      "Effect": "Deny",
      "Action": [
        "ec2:ModifySnapshotAttribute"
      ],
      "Resource": "arn:aws:ec2:*::snapshot/*",
      "Condition": {
        "StringEquals": {
        "ec2:AvailabilityZone": "use1-atl2-az1"
        }
      }
    },
```

```
    {
      "Sid": "AllowSnapshotModifyElsewhere",
      "Effect": "Allow",
      "Action": [
        "ec2:ModifySnapshotAttribute"
      ],
      "Resource": "arn:aws:ec2:*::snapshot/*"
    }
  ]
}
```

위탁자의 로컬 영역의 로컬 스냅샷 삭제 방지

다음 예제 정책은 모든 사용자가 로컬 영역의 로컬 스냅샷을 삭제하지 못하도록 합니다.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "DenySnapshotDeleteInLocalZone",
      "Effect": "Deny",
      "Action": [
        "ec2:DeleteSnapshot"
      ],
      "Resource": "arn:aws:ec2:*::snapshot/*",
      "Condition": {
        "StringEquals": {
        "ec2:AvailabilityZone": "use1-atl2-az1"
        }
      }
    },
    {
      "Sid": "AllowSnapshotDeleteElsewhere",
      "Effect": "Allow",
      "Action": [
        "ec2:DeleteSnapshot"
      ],
      "Resource": "arn:aws:ec2:*::snapshot/*"
```

```
    }
  ]
}
```

Amazon EBS 암호화

Amazon EC2 인스턴스와 연결된 Amazon EBS 리소스의 간단한 암호화 솔루션으로 Amazon EBS 암호화를 사용합니다. Amazon EBS 암호화를 사용하면 자체 키 관리 인프라를 구축, 유지 관리 및 보호할 필요가 없습니다. Amazon EBS 암호화는 암호화된 볼륨 및 스냅샷을 생성할 때 AWS KMS keys 를 사용합니다.

암호화 작업은 EC2 인스턴스를 호스팅하는 서버에서 진행되며, 인스턴스와 인스턴스에 연결된 EBS 스토리지 간 유휴 데이터 및 전송 중 데이터의 보안을 모두 보장합니다.

인스턴스에는 암호화된 볼륨과 암호화되지 않은 볼륨을 동시에 연결할 수 있습니다. 모든 Amazon EC2 인스턴스 유형은 Amazon EBS 암호화를 지원합니다.

내용

- Amazon EBS의 암호화 방식
- 로컬 영역, Outpost 및 Wavelength 영역의 Amazon EBS 암호화
- Amazon EBS 암호화의 요구 사항
- 기본적으로 Amazon EBS 암호화 활성화
- EBS 리소스 암호화
- Amazon EBS 암호화에 사용되는 AWS KMS 키 교체
- Amazon EBS 암호화 예시

Amazon EBS의 암호화 방식

EC2 인스턴스의 부팅 및 데이터 볼륨을 모두 암호화할 수 있습니다.

암호화된 EBS 볼륨을 만들고 지원되는 인스턴스 유형에 이 볼륨을 연결하면 다음 유형의 데이터가 암호화됩니다.

- 볼륨 내부에 있는 데이터
- 볼륨과 인스턴스 사이에서 이동하는 모든 데이터
- 볼륨에서 생성된 모든 스냅샷
- 그런 스냅샷에서 생성된 모든 볼륨

Amazon EBS는 산업 표준 AES-256 데이터 암호화를 사용하여 데이터 키로 볼륨을 암호화합니다. 데이터 키는에서 생성된 AWS KMS 다음 볼륨 정보와 함께 저장되기 전에 AWS KMS 키 AWS KMS 로에서 암호화됩니다. Amazon EBS는 Amazon EBS 리소스를 생성하는 각 리전 AWS 관리형 키 에서 고유한를 자동으로 생성합니다. KMS 키의 별칭은 aws/ebs입니다. 기본적으로 Amazon EBS은(는) 암호화에 이 KMS 키를 사용합니다. 또는 생성한 대칭 고객 관리형 암호화 키를 사용할 수 있습니다. 자체 KMS 키를 사용하여 KMS 키 생성, 교체 및 비활성화 기능을 비롯한 다양한 작업을 수행할 수 있습니다.

Amazon EC2는와 협력하여 암호화된 볼륨 AWS KMS 을 생성하는 스냅샷의 암호화 여부에 따라 약간 다른 방식으로 EBS 볼륨을 암호화하고 복호화합니다.

스냅샷이 암호화된 경우에 EBS 암호화가 작동하는 방식

소유한 암호화된 스냅샷에서 암호화된 볼륨을 생성하면 Amazon EC2는와 함께 다음과 같이 EBS 볼륨 AWS KMS 을 암호화하고 해독합니다.

1. Amazon EC2는 볼륨 암호화에 대해 선택한 KMS 키를 AWS KMS지정하여 GenerateDataKeyWithoutPlaintext 요청을에 보냅니다.
2. 볼륨이 스냅샷과 동일한 KMS 키를 사용하여 암호화된 경우는 스냅샷과 동일한 데이터 키를 AWS KMS 사용하고 동일한 KMS 키로 암호화합니다. 볼륨이 다른 KMS 키를 사용하여 암호화된 경우는 새 데이터 키를 AWS KMS 생성하고 지정한 KMS 키로 암호화합니다. 암호화된 데이터 키는 Amazon EBS로 전송되어 볼륨 메타데이터와 함께 저장됩니다.
3. 암호화된 볼륨을 인스턴스에 연결하면 Amazon EC2에서는 데이터 키를 해독할 수 있도록 CreateGrant 요청을 AWS KMS 로 보냅니다.
4. AWS KMS 는 암호화된 데이터 키를 복호화하고 복호화된 데이터 키를 Amazon EC2로 전송합니다.
5. Amazon EC2는 Nitro 하드웨어의 일반 텍스트 데이터 키를 사용하여 디스크 I/O를 볼륨으로 암호화합니다. 볼륨이 인스턴스에 연결되어 있는 동안에는 일반 텍스트 형태의 데이터 키가 메모리에 유지됩니다.

스냅샷이 암호화되지 않은 경우에 EBS 암호화가 작동하는 방식

암호화되지 않은 스냅샷에서 암호화된 볼륨을 생성하는 경우에는 Amazon EC2가 AWS KMS 와 함께 다음과 같이 EBS 볼륨을 암호화하고 복호화합니다.

1. Amazon EC2는 스냅샷에서 생성된 볼륨을 암호화할 수 AWS KMS있도록 CreateGrant 요청을에 보냅니다.

2. Amazon EC2는 볼륨 암호화에 대해 선택한 KMS 키를 AWS KMS지정하여 GenerateDataKeyWithoutPlaintext 요청을에 보냅니다.
3. AWS KMS 는 새 데이터 키를 생성하고 볼륨 암호화를 위해 선택한 KMS 키로 암호화한 다음 암호화된 데이터 키를 Amazon EBS에 전송하여 볼륨 메타데이터와 함께 저장합니다.
4. Amazon EC2는에 Decrypt 요청을 전송 AWS KMS 하여 암호화된 데이터 키를 복호화한 다음 볼륨 데이터를 암호화하는 데 사용합니다.
5. 암호화된 볼륨을 인스턴스에 연결하면 Amazon EC2에서는 데이터 키를 해독할 수 있도록 CreateGrant 요청을 AWS KMS로 보냅니다.
6. 암호화된 볼륨을 인스턴스에 연결하면 Amazon EC2는 암호화된 데이터 키를 AWS KMS지정하여에 Decrypt 요청을 보냅니다.
7. AWS KMS 는 암호화된 데이터 키를 복호화하고 복호화된 데이터 키를 Amazon EC2로 전송합니다.
8. Amazon EC2는 Nitro 하드웨어의 일반 텍스트 데이터 키를 사용하여 디스크 I/O를 볼륨으로 암호화합니다. 볼륨이 인스턴스에 연결되어 있는 동안에는 일반 텍스트 형태의 데이터 키가 메모리에 유지됩니다.

자세한 내용은 AWS KMS개발자 안내서의 Amazon Elastic Block Store(Amazon EBS)가AWS Key Management Service 를 사용하는 방식 및 Amazon EC2 예제 2를 참조하세요.

사용할 수 없는 KMS 키가 데이터 키에 미치는 영향

KMS 키를 사용할 수 없게 되면 효과는 거의 즉각적으로 나타납니다(최종 일관성에 따라 다름). KMS 키의 키 상태는 새로운 조건을 반영하도록 변경되며 암호화 작업에서 KMS 키를 사용하려는 모든 요청은 실패합니다.

KMS 키를 사용할 수 없게 하려는 작업을 수행해도 EC2 인스턴스나 연결된 EBS 볼륨에 즉각적인 영향은 없습니다. Amazon EC2는 KMS 키가 아닌 데이터 키를 사용하여 볼륨이 인스턴스에 연결되어 있는 동안 모든 디스크 I/O를 암호화합니다.

단, 암호화된 EBS 볼륨이 EC2 인스턴스에서 삭제되면 Amazon EBS는 데이터 키를 Nitro 하드웨어에서 제거합니다. 다음에 암호화된 EBS 볼륨을 EC2 인스턴스에 연결할 때 Amazon EBS가 KMS 키를 사용하여 볼륨의 암호화된 데이터 키를 해독하지 못하므로 연결에 실패합니다. EBS 볼륨을 다시 사용하려면 KMS 키를 다시 사용할 수 있게 만들어야 합니다.

> **Tip**
>
> 사용할 수 없게 하려는 KMS 키에서 생성된 데이터 키로 암호화된 EBS 볼륨에 저장된 데이터에 더 이상 액세스하지 않으려면 KMS 키를 사용할 수 없게 하기 전에 EC2 인스턴스에서 EBS 볼륨을 분리하는 것이 좋습니다.

자세한 내용은 AWS Key Management Service 개발자 안내서에서 사용할 수 없는 KMS 키가 데이터 키에 영향을 미치는 방법을 참조하세요.

로컬 영역, Outpost 및 Wavelength 영역의 Amazon EBS 암호화

Amazon EBS 암호화 동작은 리소스를 생성하는 위치 유형에 따라 달라집니다.

Local Zones

다음 로컬 영역에서는 모든 Amazon EBS 리소스가 기본적으로 암호화됩니다. 암호화되지 않은 볼륨이나 스냅샷은 생성할 수 없습니다. KMS 키를 지정하지 않으면 Amazon EBS는 계정의 기본 암호화 키로 리소스를 암호화합니다. 기본적으로 ()입니다 AWS 관리형 키 `aws/ebs`. 계정의 기본 암호화 키를 변경하려면 섹션을 참조하세요기본 암호화 키.

- 북미: `us-east-1-atl-1a`, `us-east-1-bos-1a`, `us-east-1-chi-1a`, `us-east-1-dfw-1a`, `us-east-1-iah-1a`, `us-east-1-mci-1a`, `us-east-1-mia-1a`, `us-east-1-msp-1a`, `us-east-1-nyc-1a`, `us-east-1-phl-1a`, `us-east-1-qro-1a`, `us-east-2-mci-1b`, `us-east-2-mci-1aus-west-2-den-1a`, `us-west-2-hnl-1a`, `us-west-2-las-1a`, `us-west-2-las-1b`, , `us-west-2-phx-1a`, `us-west-2-pdx-1a`, `us-west-2-sea-1a`
- 아시아 태평양: `ap-northeast-1-tpe-1a`, `ap-south-1-ccu-1a`, `ap-south-1-ccu-2a`, `ap-south-1-del-1a`, `ap-south-1-del-2a`, `ap-southeast-1-bkk-1a`, `ap-southeast-1-han-1a`, `ap-southeast-1-mnl-1a`, `ap-southeast-2-akl-1a`, `ap-southeast-2-per-1a`
- 유럽: `eu-central-1-ath-1a`, `eu-central-1-ham-1a`, `eu-central-1-ist-1a`, `eu-central-1-waw-1a`, `eu-north-1-cph-1a`, `eu-north-1-hel-1a`
- 남아메리카: `us-east-1-bue-1a`, `us-east-1-lim-1a`, `us-east-1-scl-1a`
- 중동: `me-south-1-mct-1a`
- 아프리카: `af-south-1-los-1a`

다른 모든 로컬 영역에서 Amazon EBS 암호화는 상위 리전과 동일한 동작을 따릅니다.

Outpost 및 Wavelength Zone

Wavelength Zone의 AWS Outposts 모든 Amazon EBS 리소스는 기본적으로 암호화됩니다. 암호화되지 않은 볼륨이나 스냅샷은 생성할 수 없습니다. KMS 키를 지정하지 않으면 Amazon EBS는 계정의 기본 암호화 키로 리소스를 암호화합니다. 기본적으로 이는 AWS 관리형 키 ()입니다aws/ebs. 계정의 기본 암호화 키를 변경하려면 섹션을 참조하세요기본 암호화 키.

이러한 영역의 로컬 스냅샷에서 볼륨을 생성하면 볼륨은 소스 스냅샷과 동일한 KMS 키를 사용하여 암호화됩니다. 그러나 로컬 스냅샷을 복사할 때 암호화 키를 변경할 수 있습니다.

Amazon EBS 암호화의 요구 사항

시작하기 전에 다음 요구 사항을 충족하는지 확인하세요.

요구 사항

- 지원되는 볼륨 유형
- 지원되는 인스턴스 유형
- 사용자의 권한
- 인스턴스에 대한 권한

지원되는 볼륨 유형

암호화는 모든 EBS 볼륨 유형에서 지원됩니다. 암호화된 볼륨에서는 지연 시간에 대한 영향을 최소화한 채 암호화되지 않은 볼륨과 동일한 IOPS 성능을 기대할 수 있습니다. 암호화되지 않은 볼륨에 액세스하는 것과 동일한 방법으로 암호화된 볼륨에 액세스할 수 있습니다. 암호화 및 암호 해독은 중단 없이 처리되므로 사용자나 사용자의 애플리케이션에서 별도로 조치할 부분은 없습니다.

지원되는 인스턴스 유형

Amazon EBS 암호화는 모든 현재 세대 및 이전 세대 인스턴스 유형에서 사용할 수 있습니다.

사용자의 권한

EBS 암호화에 KMS 키를 사용하는 경우 KMS 키 정책은 필요한 AWS KMS 작업에 액세스할 수 있는 모든 사용자가이 KMS 키를 사용하여 EBS 리소스를 암호화하거나 복호화할 수 있도록 허용합니다. EBS 암호화를 사용하기 위해 다음 작업을 호출할 수 있는 권한을 사용자에게 부여해야 합니다.

- kms:CreateGrant
- kms:Decrypt
- kms:DescribeKey
- kms:GenerateDataKeyWithoutPlaintext
- kms:ReEncrypt

> **Tip**
>
> 최소 권한의 원칙을 따르려면 kms:CreateGrant에 대한 전체 액세스 권한을 허용하지 마세요. 대신 다음 예제와 AWS 같이 kms:GrantIsForAWSResource 조건 키를 사용하여 서비스가 사용자를 대신하여 권한 부여를 생성하는 경우에만 사용자가 KMS 키에 대한 권한 부여를 생성하도록 허용합니다.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "kms:CreateGrant",
            "Resource": [
                "arn:aws:kms:us-east-2:123456789012:key/abcd1234-a123-456d-a12b-
a123b4cd56ef"
            ],
            "Condition": {
                "Bool": {
                    "kms:GrantIsForAWSResource": true
                }
            }
        }
    ]
}
```

자세한 내용은 AWS Key Management Service 개발자 안내서의 기본 키 정책 섹션에서 AWS 계정에 대한 액세스 허용 및 IAM 정책 활성화를 참조하세요.

인스턴스에 대한 권한

인스턴스가 암호화된 AMI, 볼륨 또는 스냅샷과 상호 작용을 시도할 경우 인스턴스의 자격 증명 전용 역할에 KMS 키 부여가 발급됩니다. 자격 증명 전용 역할은 인스턴스가 사용자를 대신하여 암호화된 AMI, 볼륨 또는 스냅샷과 상호 작용하기 위해 사용하는 IAM 역할입니다.

자격 증명 전용 역할은 수동으로 만들거나 삭제할 필요가 없으며 관련 정책도 없습니다. 또한 자격 증명 전용 역할 보안 인증에는 액세스할 수 없습니다.

> **Note**
>
> 인스턴스의 애플리케이션에서는 자격 증명 전용 역할을 사용하여 Amazon S3 객체 또는 Dynamo DB 테이블과 같은 다른 AWS KMS 암호화된 리소스에 액세스하지 않습니다. 이러한 작업은 Amazon EC2 인스턴스 역할의 자격 증명 또는 인스턴스에 구성한 기타 자격 AWS 증명을 사용하여 수행됩니다.

자격 증명 전용 역할에는 서비스 제어 정책(SCP) 및 KMS 키 정책이 적용됩니다. SCP 또는 KMS 키가 KMS 키에 대한 자격 증명 전용 역할 액세스를 거부하는 경우 암호화된 볼륨이 있거나 암호화된 AMI 또는 스냅샷을 사용하여 EC2 인스턴스를 시작하지 못할 수 있습니다.

`aws:SourceIp`, , 또는 `aws:SourceVpce` AWS 전역 조건 키를 사용하여 네트워크 위치를 기반으로 액세스를 거부하는 SCP `aws:VpcSourceIp` `aws:SourceVpc`또는 키 정책을 생성하는 경우 이러한 정책 설명이 인스턴스 전용 역할에 적용되지 않도록 해야 합니다. 예제 정책은 데이터 경계 정책 예제를 참조하세요.

자격 증명 전용 역할 ARN은 다음 형식을 사용합니다.

```
arn:aws-partition:iam::account_id:role/aws:ec2-infrastructure/instance_id
```

인스턴스에 키 부여가 발행되면 해당 인스턴스에 고유한 위임된 역할 세션에 키 부여가 발행됩니다. 피부여자 위탁자 ARN은 다음 형식을 사용합니다.

```
arn:aws-partition:sts::account_id:assumed-role/aws:ec2-infrastructure/instance_id
```

기본적으로 Amazon EBS 암호화 활성화

생성한 새 EBS 볼륨 및 스냅샷 복사본의 암호화를 적용하도록 AWS 계정을 구성할 수 있습니다. 예를 들어 Amazon EBS는 인스턴스 시작 시 생성된 EBS 볼륨과 암호화되지 않은 스냅샷에서 복사하는 스

냅샷을 암호화합니다. 암호화되지 않은 EBS 리소스에서 암호화된 리소스로의 이전 예제는 암호화되지 않은 리소스 암호화 섹션을 참조하세요.

기본적으로 암호화는 기존 EBS 볼륨이나 스냅샷에 영향을 미치지 않습니다.

고려 사항

- 암호화 기본 제공은 리전별 설정입니다. 특정 기능에 대해 이 기능을 활성화하면 해당 리전의 개별 볼륨 또는 스냅샷에 대해 비활성화할 수 없습니다.
- Amazon EBS 암호화는 기본적으로 모든 현재 세대 및 이전 세대 인스턴스 유형에서 지원됩니다.
- 스냅샷을 복사하고 새 KMS 키로 암호화하면 전체(비증분) 복사본이 생성됩니다. 이로 인해 추가 스토리지 비용이 발생합니다.

Console

리전에서 암호화를 기본적으로 활성화하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 모음에서 해당 리전을 선택합니다.
3. 탐색 창에서 설정을 선택한 다음 데이터 보호 및 보안 탭을 선택합니다.
4. EBS 암호화 섹션에서 관리를 선택합니다.
5. 활성화를 선택합니다. 사용자를 대신하여 aws/ebs 생성된 별칭 AWS 관리형 키 을 기본 암호화 키로 사용하여를 유지하거나 대칭 고객 관리형 암호화 키를 선택합니다.
6. EBS 암호화 업데이트(Update EBS encryption)를 선택합니다.

AWS CLI

기본 설정에 따른 암호화 보기

get-ebs-encryption-by-default 명령을 사용합니다.

- 특정 리전

```
aws ec2 get-ebs-encryption-by-default --region region
```

- 계정 내 모든 리전

```
echo -e "Region      \t Encrypt \t Key"; \
```

```
echo -e "----------- \t ------- \t -------" ; \
for region in $(aws ec2 describe-regions --region us-east-1 --query "Regions[*].
[RegionName]" --output text);
do
    default=$(aws ec2 get-ebs-encryption-by-default --region $region --query
 "{Encryption_By_Default:EbsEncryptionByDefault}" --output text);
    kms_key=$(aws ec2 get-ebs-default-kms-key-id --region $region | jq
 '.KmsKeyId');
    echo -e "$region \t $default \t\t $kms_key";
done
```

기본적으로 암호화 활성화

enable-ebs-encryption-by-default 명령을 사용합니다.

- 특정 리전

```
aws ec2 enable-ebs-encryption-by-default --region region
```

- 계정 내 모든 리전

```
echo -e "Region      \t Encrypt \t Key"; \
echo -e "----------- \t ------- \t -------" ; \
for region in $(aws ec2 describe-regions --region us-east-1 --query "Regions[*].
[RegionName]" --output text);
do
    default=$(aws ec2 enable-ebs-encryption-by-default --region $region --query
 "{Encryption_By_Default:EbsEncryptionByDefault}" --output text);
    kms_key=$(aws ec2 get-ebs-default-kms-key-id --region $region | jq
 '.KmsKeyId');
    echo -e "$region \t $default \t\t $kms_key";
done
```

기본적으로 암호화 비활성화

disable-ebs-encryption-by-default 명령을 사용합니다.

- 특정 리전

```
aws ec2 disable-ebs-encryption-by-default --region region
```

- 계정 내 모든 리전

```
echo -e "Region      \t Encrypt \t Key"; \
echo -e "----------- \t ------- \t -------" ; \
for region in $(aws ec2 describe-regions --region us-east-1 --query "Regions[*].
[RegionName]" --output text);
do
    default=$(aws ec2 disable-ebs-encryption-by-default --region $region --query
 "{Encryption_By_Default:EbsEncryptionByDefault}" --output text);
    kms_key=$(aws ec2 get-ebs-default-kms-key-id --region $region | jq
 '.KmsKeyId');
    echo -e "$region \t $default \t\t $kms_key";
done
```

PowerShell

기본 설정에 따른 암호화 보기

Get-EC2EbsEncryptionByDefault cmdlet을 사용합니다.

- 특정 리전

```
Get-EC2EbsEncryptionByDefault -Region region
```

- 계정 내 모든 리전

```
(Get-EC2Region).RegionName |
    ForEach-Object {
    [PSCustomObject]@{
        Region                    = $_
        EC2EbsEncryptionByDefault = Get-EC2EbsEncryptionByDefault -Region $_
        EC2EbsDefaultKmsKeyId     = Get-EC2EbsDefaultKmsKeyId -Region $_
    } } |
    Format-Table -AutoSize
```

기본적으로 암호화 활성화

Enable-EC2EbsEncryptionByDefault cmdlet을 사용합니다.

- 특정 리전

```
Enable-EC2EbsEncryptionByDefault -Region region
```

- 계정 내 모든 리전

```
(Get-EC2Region).RegionName |
    ForEach-Object {
    [PSCustomObject]@{
        Region                      = $_
        EC2EbsEncryptionByDefault = Enable-EC2EbsEncryptionByDefault -Region $_
        EC2EbsDefaultKmsKeyId     = Get-EC2EbsDefaultKmsKeyId -Region $_
    } } |
    Format-Table -AutoSize
```

기본적으로 암호화 비활성화

Disable-EC2EbsEncryptionByDefault cmdlet을 사용합니다.

- 특정 리전

```
Disable-EC2EbsEncryptionByDefault -Region region
```

- 계정 내 모든 리전

```
(Get-EC2Region).RegionName |
    ForEach-Object {
    [PSCustomObject]@{
        Region                      = $_
        EC2EbsEncryptionByDefault = Disable-EC2EbsEncryptionByDefault -Region $_
        EC2EbsDefaultKmsKeyId     = Get-EC2EbsDefaultKmsKeyId -Region $_
    } } |
    Format-Table -AutoSize
```

기존 스냅샷이나 암호화된 볼륨과 연동되어 있는 KMS 키는 변경할 수 없습니다. 하지만 스냅샷 복사 작업 중에 다른 KMS 키와(과) 연동시킬 수는 있습니다. 복사된 스냅샷은 새로운 KMS 키(으)로 암호화됩니다.

기본 암호화 키

Amazon EBS는 Amazon EBS 리소스를 생성하는 각 리전 AWS 관리형 키 에서 고유한를 자동으로 생성합니다. 이 KMS 키의 별칭은 입니다aws/ebs. 기본 암호화 키입니다. 기본 암호화 키를 고객 관리형 키로 변경할 수 있습니다.

리전의 기본 암호화 키를 변경하려면 다음 방법 중 하나를 사용합니다.

Console

리전의 기본 암호화 키를 변경하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 모음에서 해당 리전을 선택합니다.
3. 탐색 창에서 설정을 선택한 다음 데이터 보호 및 보안 탭을 선택합니다.
4. EBS 암호화 섹션에서 관리를 선택합니다.
5. 기본 암호화 키에서 사용할 KMS 키 ID를 선택합니다.
6. EBS 암호화 업데이트(Update EBS encryption)를 선택합니다.

AWS CLI

modify-ebs-default-kms-key-id 명령을 사용합니다.

```
aws ec2 modify-ebs-default-kms-key-id --kms-key-id key-id --region region
```

PowerShell

Edit-EC2EbsDefaultKmsKeyId cmdlet을 사용합니다.

```
Edit-EC2EbsDefaultKmsKeyId -KmsKeyId key-id -Region region
```

EBS 리소스 암호화

암호화를 기본으로 사용하여 암호화를 활성화하거나 암호화하려는 볼륨을 생성할 때 암호화를 활성화하여 EBS 볼륨을 암호화합니다.

볼륨 암호화 시 볼륨을 암호화하는 데 사용할 대칭 암호화 KMS 키를 지정할 수 있습니다. KMS 키를 지정하지 않은 경우 암호화에 사용되는 KMS 키는 소스 스냅샷 및 해당 소유권의 암호화 상태에 따라 달라집니다. 자세한 내용은 암호화 결과표를 참조하십시오.

> **Note**
>
> API 또는를 사용하여 KMS 키를 AWS CLI 지정하는 경우가 KMS 키를 비동기적으로 AWS 인증한다는 점에 유의하세요. 유효하지 않은 KMS 키 ID, 별칭 또는 ARN을 지정하면 작업이 완료된 것처럼 보이지만 결국 실패합니다.

기존 스냅샷이나 볼륨과 연동되어 있는 KMS 키는 변경할 수 없습니다. 하지만 스냅샷 복사 작업 중에 다른 KMS 키와(과) 연동시킬 수는 있습니다. 복사된 스냅샷은 새로운 KMS 키(으)로 암호화됩니다.

빈 볼륨 생성 시 암호화

비어 있는 새 EBS 볼륨을 생성하는 경우 특정 볼륨 생성 작업에 대한 암호화를 활성화하여 암호화할 수 있습니다. EBS 암호화를 기본으로 활성화한 경우 볼륨은 EBS 암호화에 대한 기본 KMS 키를 사용하여 자동으로 암호화됩니다. 또는 특정 볼륨 생성 작업에 대해 다른 대칭 암호화 KMS 키를 지정할 수 있습니다. 볼륨은 최초로 사용 가능한 시점까지 암호화되므로 데이터가 항상 안전한 상태를 유지합니다. 자세한 절차는 Amazon EBS 볼륨 생성 섹션을 참조하세요.

기본적으로 볼륨을 생성할 때 선택한 KMS 키이(가) 볼륨에서 생성한 스냅샷과 암호화된 스냅샷에서 복원한 볼륨을 암호화합니다. 암호화된 볼륨 또는 스냅샷으로부터 암호화를 제거할 수 없습니다. 즉, 암호화된 스냅샷 또는 암호화된 스냅샷의 사본에서 복원된 볼륨은 항상 암호화됩니다.

암호화된 볼륨의 퍼블릭 스냅샷은 지원하지 않지만 암호화된 스냅샷을 특정 계정과 공유할 수는 있습니다. 자세한 지침은 Amazon EBS 스냅샷을 다른 AWS 계정과 공유 섹션을 참조하세요.

암호화되지 않은 리소스 암호화

암호화되지 않은 기존의 볼륨이나 스냅샷은 직접 암호화할 수 없습니다.

암호화되지 않은 볼륨을 암호화하려면 해당 볼륨의 스냅샷을 생성한 다음 스냅샷을 사용하여 새 암호화된 볼륨을 생성합니다. 자세한 내용은 스냅샷 생성 및 볼륨 생성 섹션을 참조하세요.

암호화되지 않은 스냅샷을 암호화하려면 해당 스냅샷의 암호화된 복사본을 생성합니다. 자세한 내용은 스냅샷 복사 섹션을 참조하세요.

계정의 기본 암호화를 활성화하면 암호화되지 않은 스냅샷에서 생성된 볼륨 및 스냅샷 복사본이 항상 암호화됩니다. 활성화되지 않은 경우에는 요청에 암호화 파라미터를 지정해야 합니다. 자세한 내용은 기본적으로 암호화 사용 섹션을 참조하세요.

Amazon EBS 암호화에 사용되는 AWS KMS 키 교체

암호화 모범 사례에 따르면 암호화 키를 광범위하게 사용하지 않는 것이 좋습니다.

Amazon EBS 암호화에 사용할 새 암호화 자료를 생성하려면 새 고객 관리형 키를 생성한 다음 새로 생성한 KMS 키를 사용하도록 애플리케이션을 변경하면 됩니다. 또는 기존 고객 관리형 키에 대해 자동 키 교체를 활성화할 수 있습니다.

고객 관리형 키에 대해 자동 키 교체를 활성화하면는 매년 KMS 키에 대한 새 암호화 구성 요소를 AWS KMS 생성합니다.는 암호화 구성 요소의 모든 이전 버전을 AWS KMS 저장하므로 해당 KMS 키 구성 요소로 이전에 암호화된 볼륨 및 스냅샷을 계속 해독하고 사용할 수 있습니다.는 KMS 키를 삭제할 때까지 교체된 키 구성 요소를 삭제하지 AWS KMS 않습니다.

교체된 고객 관리형 키를 사용하여 새 볼륨 또는 스냅샷을 암호화하는 경우는 현재(새) 키 구성 요소를 AWS KMS 사용합니다. 교체된 고객 관리형 키를 사용하여 볼륨 또는 스냅샷을 해독하는 경우 AWS KMS 는 이를 암호화하는 데 사용된 버전의 암호화 구성 요소를 사용합니다. 볼륨 또는 스냅샷이 이전 버전의 암호화 자료로 암호화된 경우 AWS KMS , 해당 이전 버전을 사용하여 복호화합니다.는 키 교체 후 새 암호화 자료를 사용하기 위해 이전에 암호화된 볼륨 또는 스냅샷을 다시 암호화하지 AWS KMS 않습니다. 원래 사용된 암호화 구성 요소로 암호화된 상태로 유지됩니다. 코드 변경 없이 애플리케이션 및 AWS 서비스에서 교체된 고객 관리형 키를 안전하게 사용할 수 있습니다.

> **Note**
>
> - 자동 키 교체는가 AWS KMS 생성하는 키 구성 요소가 있는 대칭 고객 관리형 키에만 지원됩니다.
> - AWS KMS 는 AWS 관리형 키 매년 자동으로 교체됩니다. AWS 관리형 키에 대한 키 교체를 사용 설정 또는 사용 중지할 수 없습니다.

자세한 내용을 알아보려면 AWS Key Management Service 개발자 안내서의 KMS 키 교체를 참조하세요.

Amazon EBS 암호화 예시

암호화된 EBS 리소스를 생성하면 볼륨 생성 파라미터에서 다른 고객 관리형 키를 지정하거나 AMI 또는 인스턴스에 대한 블록 디바이스 매핑을 지정하지 않는 한, 사용자 계정의 EBS 암호화에 대한 기본 KMS 키에 의해 암호화됩니다.

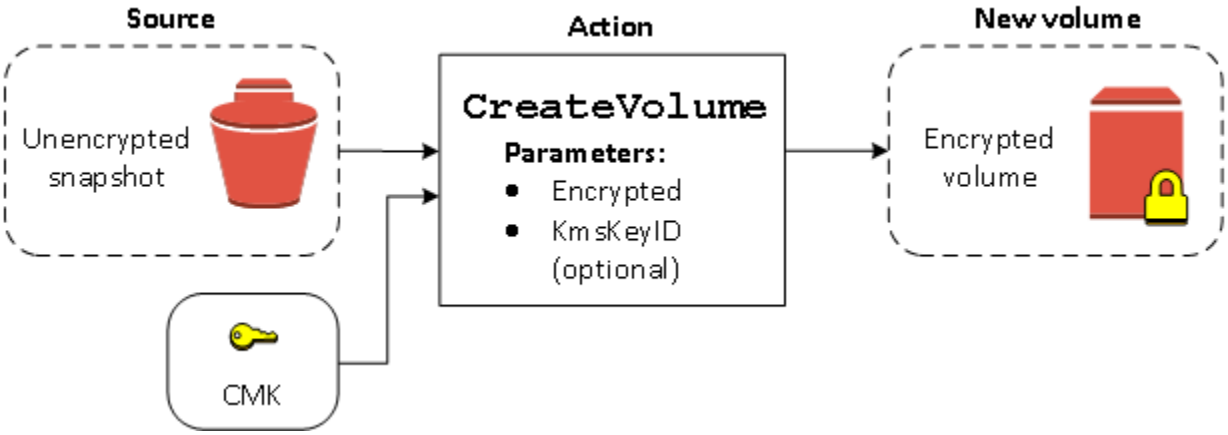
다음 예제에서는 볼륨 및 스냅샷의 암호화 상태를 관리할 수 있는 방법을 보여줍니다. 암호화 사례의 전체 목록은 암호화 결과표를 참조하십시오.

예제:

- 암호화되지 않은 볼륨(활성화되지 않은 암호화 기본 제공) 복원
- 암호화되지 않은 볼륨(활성화된 암호화 기본 제공) 복원
- 암호화되지 않은 스냅샷(활성화되지 않은 암호화 기본 제공) 복사
- 암호화되지 않은 스냅샷(활성화된 암호화 기본 제공) 복사
- 암호화된 볼륨의 재암호화
- 암호화된 스냅샷의 재암호화
- 암호화된 볼륨과 암호화되지 않은 볼륨 간 데이터 마이그레이션
- 암호화 결과

암호화되지 않은 볼륨(활성화되지 않은 암호화 기본 제공) 복원

암호화 기본 제공을 활성화하지 않은 상태에서는 암호화되지 않은 스냅샷에서 복원한 볼륨이 기본적으로 암호화되지 않습니다. 그러나 `Encrypted` 파라미터와, 선택적으로, `KmsKeyId` 파라미터를 설정하여, 결과로 얻은 볼륨을 암호화할 수 있습니다. 다음 다이어그램에서 프로세스를 보여 줍니다.

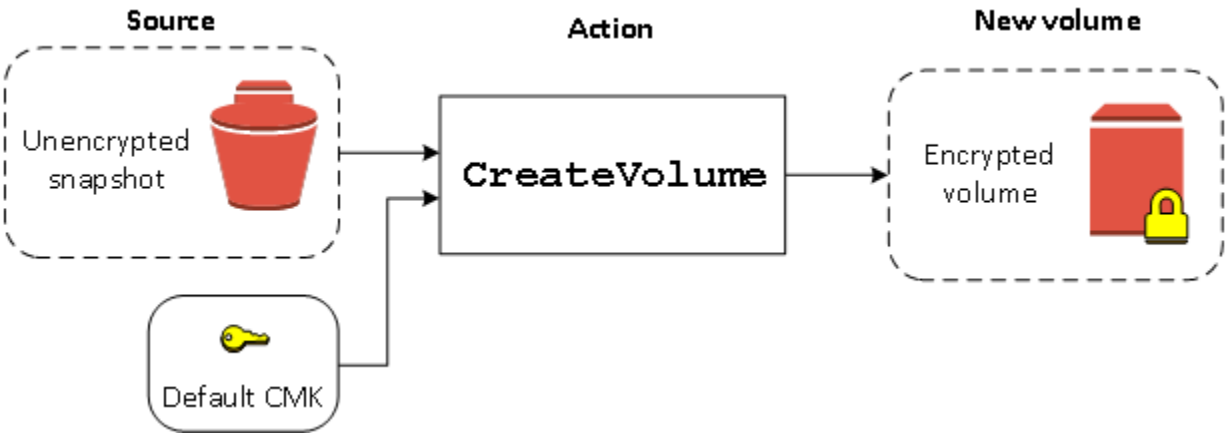


KmsKeyId 파라미터를 그대로 놓아두면 결과로 얻은 볼륨이 EBS 암호화에 대한 기본 KMS 키를 사용하여 암호화됩니다. KMS 키 ID를 지정하여 해당 볼륨을 다른 KMS 키(으)로 암호화해야 합니다.

자세한 내용은 Amazon EBS 볼륨 생성 섹션을 참조하세요.

암호화되지 않은 볼륨(활성화된 암호화 기본 제공) 복원

암호화를 기본적으로 활성화한 경우 암호화되지 않은 스냅샷에서 복원된 볼륨에 대한 암호화가 필수이며 사용자의 기본 KMS 키를 사용하는 데 암호화 파라미터는 필요하지 않습니다. 다음 다이어그램은 이러한 간단한 기본 사례를 보여 줍니다.

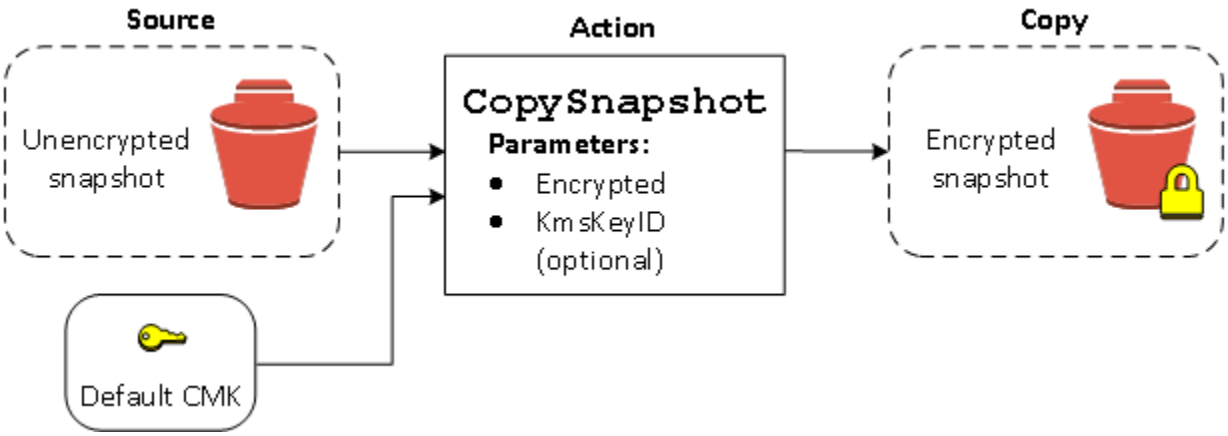


복원된 볼륨을 대칭 고객 관리형 암호화 키로 암호화하려면 Encrypted에서처럼 KmsKeyId 및 암호화되지 않은 볼륨(활성화되지 않은 암호화 기본 제공) 복원 파라미터를 모두 입력해야 합니다.

암호화되지 않은 스냅샷(활성화되지 않은 암호화 기본 제공) 복사

암호화 기본 제공을 활성화하지 않은 상태에서는 암호화되지 않은 스냅샷의 사본이 기본적으로 암호화되지 않습니다. 그러나 Encrypted 파라미터와, 선택적으로, KmsKeyId 파라미터를 설정하여, 결과로 얻은 볼륨을 암호화할 수 있습니다. KmsKeyId를 생략하면 결과로 얻은 스냅샷이 사용자의 기본 KMS 키로 암호화됩니다. KMS 키 ID를 지정하여 해당 볼륨을 다른 대칭 암호화 KMS 키로 암호화해야 합니다.

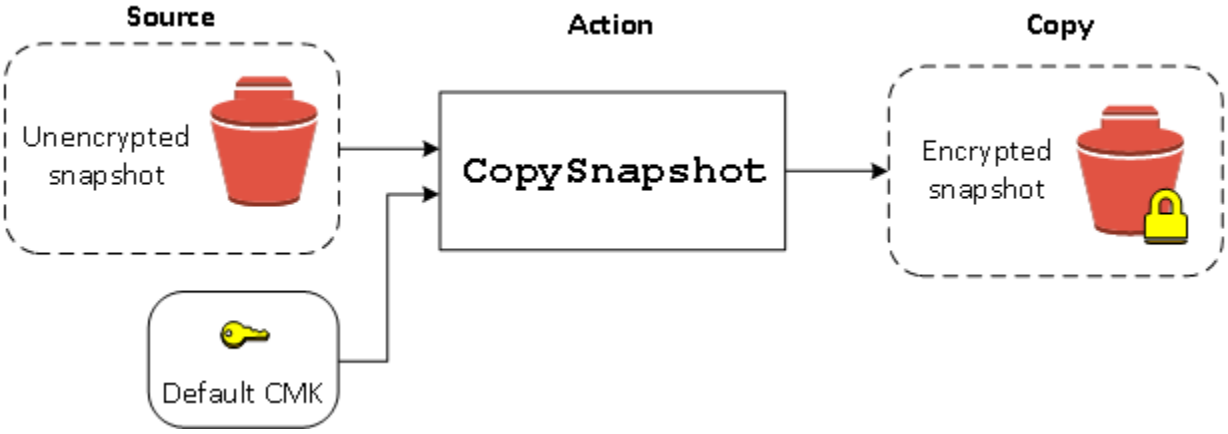
다음 다이어그램에서 프로세스를 보여 줍니다.



암호화되지 않은 스냅샷을 암호화된 스냅샷에 복사한 다음 암호화된 스냅샷에서 볼륨을 생성하여 EBS 볼륨을 암호화할 수 있습니다. 자세한 내용은 Amazon EBS 스냅샷 복사 섹션을 참조하세요.

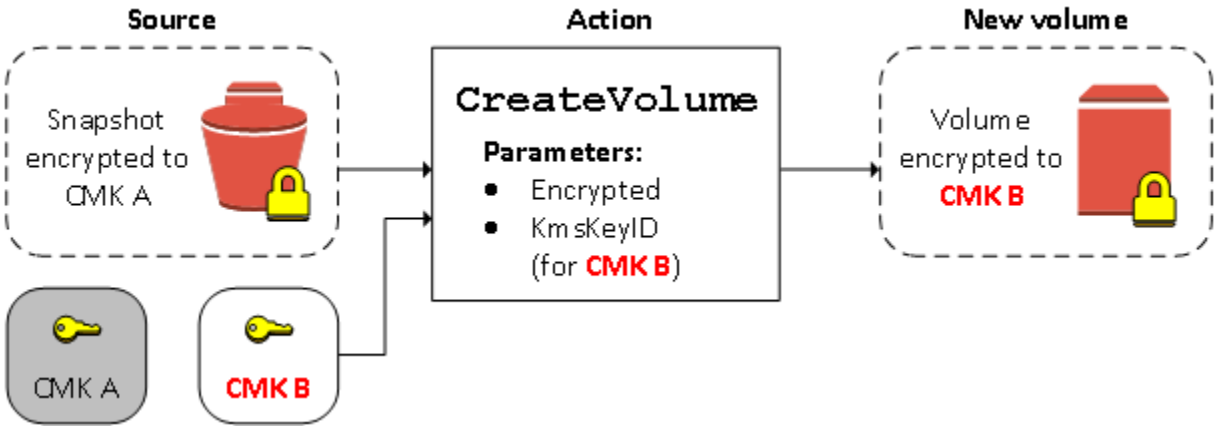
암호화되지 않은 스냅샷(활성화된 암호화 기본 제공) 복사

암호화를 기본적으로 활성화한 경우 암호화되지 않은 스냅샷의 복사가 필수이며 사용자의 기본 KMS 키이(가) 사용된다면 암호화 파라미터는 필요하지 않습니다. 다음 다이어그램에서 기본 사례를 보여 줍니다.



암호화된 볼륨의 재암호화

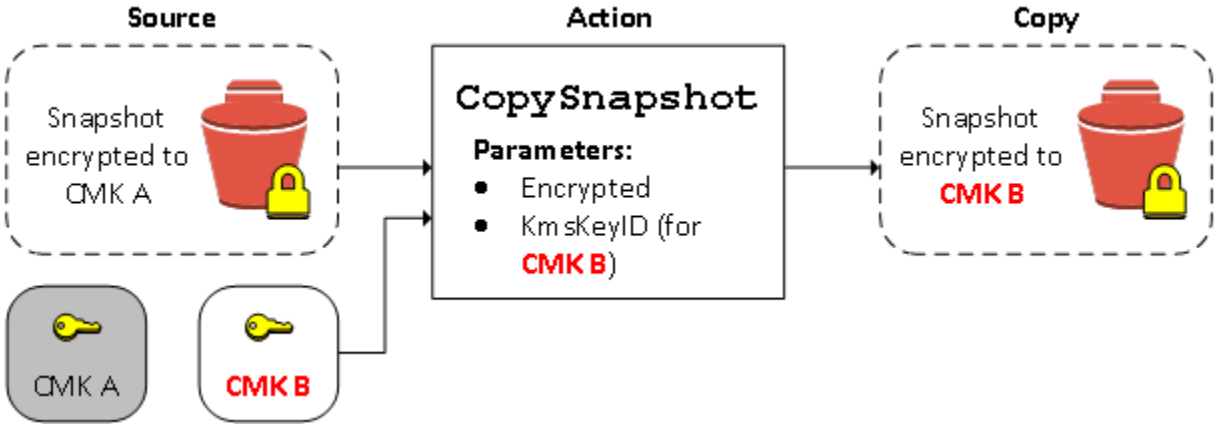
암호화된 스냅샷에서 `CreateVolume` 작업을 수행할 때 다른 KMS 키를 통해 재암호화하는 옵션이 있습니다. 다음 다이어그램에서 프로세스를 보여 줍니다. 이 예에서는 KMS 키 A와 KMS 키 B, 2개의 KMS 키이(가) 있습니다. 원본 스냅샷은 KMS 키 A로 암호화됩니다. 볼륨을 생성하는 동안 파라미터로 지정된 KMS 키 B의 KMS 키 ID를 통해 원본 데이터가 자동으로 암호 해독된 다음 KMS 키 B로 재암호화됩니다.



자세한 내용은 Amazon EBS 볼륨 생성 섹션을 참조하세요.

암호화된 스냅샷의 재암호화

복사 중에 스냅샷을 암호화하는 기능을 사용하여 자신이 소유하고 있는 이미 암호화된 스냅샷에 새 대칭 암호화 KMS 키를 적용할 수 있습니다. 새 KMS 키를 사용하여 결과 복사본에서 복원된 볼륨에만 액세스할 수 있습니다. 다음 다이어그램에서 프로세스를 보여 줍니다. 이 예에서는 KMS 키 A와 KMS 키 B, 2개의 KMS 키이(가) 있습니다. 원본 스냅샷은 KMS 키 A로 암호화됩니다. 복사하는 동안 파라미터로 지정된 KMS 키 B의 KMS 키 ID를 통해 원본 데이터가 자동으로 KMS 키 B로 재암호화됩니다.



관련된 시나리오에서 자신과 공유된 스냅샷의 복사본에 새 암호화 파라미터를 적용하도록 선택할 수도 있습니다. 기본적으로 이 복사본은 스냅샷의 소유자가 공유하는 KMS 키(으)로 암호화됩니다. 하지만 복사 프로세스 중에 자신이 관리하는 다른 KMS 키를 사용하여 공유 스냅샷의 복사본을 만드는 게 좋습니다. 이를 통해 원래 KMS 키이(가) 손상되거나 소유자가 어떤 이유로든 KMS 키를 취소하는 경우 볼륨에 대한 액세스 권한을 보호할 수 있습니다. 자세한 내용은 암호화 및 스냅샷 복사 섹션을 참조하세요.

암호화된 볼륨과 암호화되지 않은 볼륨 간 데이터 마이그레이션

암호화된 볼륨과 암호화되지 않은 볼륨에 모두 액세스할 수 있는 경우, 둘 사이에서 자유롭게 데이터를 전송할 수 있습니다. EC2는 암호화 및 복호화 작업을 투명하게 수행합니다.

Linux 인스턴스

예를 들어 rsync 명령을 사용하여 데이터를 복사합니다. 다음 명령에서 소스 데이터는 `/mnt/source`에 있고 대상 볼륨은 `/mnt/destination`에 마운트되어 있습니다.

```
[ec2-user ~]$ sudo rsync -avh --progress /mnt/source/ /mnt/destination/
```

Windows 인스턴스

예를 들어 robocopy 명령을 사용하여 데이터를 복사합니다. 다음 명령에서 소스 데이터는 `D:\`에 있고 대상 볼륨은 `E:\`에 마운트되어 있습니다.

```
PS C:\> robocopy D:\sourcefolder E:\destinationfolder /e /copyall /eta
```

숨겨진 폴더로 인한 잠재적 문제를 방지하기 위해 전체 볼륨을 복사하는 대신 폴더를 사용하는 것이 좋습니다.

암호화 결과

다음 표에서는 가능한 각 설정 조합에 대한 암호화 결과를 설명합니다.

암호화를 활성화합니까?	암호화를 기본 설정합니까?	볼륨의 소스	기본값(고객 관리형 키가 지정되지 않음)	사용자 지정(고객 관리형 키가 지정됨)
아니요	아니요	새(빈) 볼륨	암호화되지 않음	해당 사항 없음
아니요	아니요	암호화되지 않은 소유 스냅샷	암호화되지 않음	
아니요	아니요	암호화된 소유 스냅샷	동일한 키로 암호화됨	

암호화를 활성화합니까?	암호화를 기본 설정합니까?	볼륨의 소스	기본값(고객 관리형 키가 지정되지 않음)	사용자 지정(고객 관리형 키가 지정됨)
아니요	아니요	암호화되지 않은 공유 스냅샷	암호화되지 않음	
아니요	아니요	암호화된 공유 스냅샷	기본 고객 관리형 키로 암호화됨*	
예	아니요	새 볼륨	기본 고객 관리형 키로 암호화됨	지정된 고객 관리형 키로 암호화됨**
예	아니요	암호화되지 않은 소유 스냅샷	기본 고객 관리형 키로 암호화됨	
예	아니요	암호화된 소유 스냅샷	동일한 키로 암호화됨	
예	아니요	암호화되지 않은 공유 스냅샷	기본 고객 관리형 키로 암호화됨	
예	아니요	암호화된 공유 스냅샷	기본 고객 관리형 키로 암호화됨	
아니요	예	새(빈) 볼륨	기본 고객 관리형 키로 암호화됨	해당 사항 없음
아니요	예	암호화되지 않은 소유 스냅샷	기본 고객 관리형 키로 암호화됨	
아니요	예	암호화된 소유 스냅샷	동일한 키로 암호화됨	
아니요	예	암호화되지 않은 공유 스냅샷	기본 고객 관리형 키로 암호화됨	
아니요	예	암호화된 공유 스냅샷	기본 고객 관리형 키로 암호화됨	

암호화를 활성화합니까?	암호화를 기본 설정합니까?	볼륨의 소스	기본값(고객 관리형 키가 지정되지 않음)	사용자 지정(고객 관리형 키가 지정됨)
예	예	새 볼륨	기본 고객 관리형 키로 암호화됨	지정된 고객 관리형 키로 암호화됨
예	예	암호화되지 않은 소유 스냅샷	기본 고객 관리형 키로 암호화됨	
예	예	암호화된 소유 스냅샷	동일한 키로 암호화됨	
예	예	암호화되지 않은 공유 스냅샷	기본 고객 관리형 키로 암호화됨	
예	예	암호화된 공유 스냅샷	기본 고객 관리형 키로 암호화됨	

* 계정 AWS 및 리전의 EBS 암호화에 사용되는 기본 고객 관리형 키입니다. 기본적으로 EBS에 AWS 관리형 키 고유하거나 고객 관리형 키를 지정할 수 있습니다.

** 이는 시작할 때 볼륨에 지정된 고객 관리형 키입니다. 이 고객 관리형 키는 AWS 계정 및 리전의 기본 고객 관리형 키 대신 사용됩니다.

Amazon EBS 볼륨 성능

I/O 특성, 인스턴스와 볼륨의 구성 등 여러 가지 요인이 Amazon EBS 볼륨에 영향을 끼칠 수 있습니다. Amazon EBS 및 Amazon EC2 제품 세부 정보 페이지의 지침을 따른다면 일반적으로 우수한 성능을 달성할 수 있습니다. 다만 피크 성능을 달성하려면 조금 조정해야 하는 경우도 있습니다. 벤치마킹 외에도 실제 워크로드의 정보에 따라 성능을 튜닝하여 최적의 구성을 결정하는 것이 좋습니다. EBS 볼륨 작업의 기초를 배운 후에는 필요한 I/O 성능과 그러한 요구 사항에 맞게 Amazon EBS 성능을 향상하기 위한 옵션을 살펴보는 것이 좋습니다.

AWS EBS 볼륨 유형의 성능에 대한 업데이트는 기존 볼륨에 즉시 적용되지 않을 수 있습니다. 기존 볼륨의 전체 성능을 확인하려면 먼저 볼륨에 대해 `ModifyVolume` 작업을 수행해야 할 수 있습니다. 자세한 내용은 탄력적 볼륨 작업을 사용하여 Amazon EBS 볼륨 수정 단원을 참조하십시오.

내용

- Amazon EBS 성능 팁
- Amazon EBS 최적화
- Amazon EBS 볼륨 초기화
- 구성 가능한 인스턴스 대역폭 가중치
- Amazon EBS I/O 기능 및 모니터링
- Amazon EBS 및 RAID 구성
- Amazon EBS 볼륨 벤치마킹

Amazon EBS 성능 팁

이러한 팁은 다양한 사용자 시나리오에서 최적의 EBS 볼륨 성능을 달성하는 방법에 대한 모범 사례를 보여줍니다.

EBS 최적화 인스턴스 사용

EBS 최적화 처리량을 지원하지 않는 인스턴스에서는 네트워크 트래픽이 사용자의 인스턴스와 EBS 볼륨 간 트래픽과 경합할 수 있습니다. EBS 최적화 인스턴스에서는 이 두 유형의 트래픽이 분리되어 있습니다. 일부 EBS 최적화 인스턴스 구성은 추가 요금을 요구하지만(예: C3, R3, M3), 일부는 추가 요금 없이 항상 EBS에 최적화됩니다(예: M4, C4, C5, D2). 자세한 내용은 Amazon EBS 최적화 단원을 참조하십시오.

인스턴스 대역폭 구성

지원되는 인스턴스 유형의 경우, ebs-1 대역폭 가중치를 사용하여 Amazon EBS 대역폭을 25% 늘리도록 인스턴스 대역폭 가중치를 구성할 수 있습니다. 이 기능을 사용하면 EBS 및 VPC 네트워킹 간에 인스턴스의 네트워크 리소스 할당을 최적화하여 잠재적으로 I/O 집약적 워크로드의 EBS 성능을 개선할 수 있습니다. 자세한 내용은 구성 가능한 인스턴스 대역폭 가중치 단원을 참조하십시오.

성능 계산 방법 이해

EBS 볼륨의 성능을 측정할 때는 관련된 측정 단위와 성능 계산 방법을 이해해야 합니다. 자세한 내용은 Amazon EBS I/O 기능 및 모니터링 섹션을 참조하세요.

워크로드 이해

EBS 볼륨의 최대 성능, I/O 작업의 크기와 횟수, 각 작업을 완료하는 데 걸리는 시간은 서로 관련이 있습니다. 이러한 각 요소(성능, I/O 및 지연 시간)는 서로에게 영향을 미치며 애플리케이션마다 다른 요소에 더 민감합니다. 자세한 내용은 Amazon EBS 볼륨 벤치마킹 섹션을 참조하세요.

스냅샷에서 볼륨을 초기화하는 경우 성능 저하에 유의

스냅샷에서 생성된 새 EBS 볼륨의 각 데이터 블록에 처음 액세스할 때 지연 시간이 상당히 증가합니다. 다음 옵션 중 하나를 사용하여 이 성능 저하를 방지할 수 있습니다.

- 볼륨을 프로덕션에 투입하기 전에 각 블록에 액세스합니다. 이 프로세스를 초기화(이전에는 사전 워밍이라고 함)라고 합니다. 자세한 내용은 볼륨 생성 후 수동 초기화 단원을 참조하십시오.
- 스냅샷에서 빠른 스냅샷 복원을 활성화하여 스냅샷에서 생성된 EBS 볼륨이 생성 시 완전히 초기화되고 프로비저닝된 모든 성능을 즉시 제공할 수 있도록 합니다. 자세한 내용은 Amazon EBS 빠른 스냅샷 복원 단원을 참조하십시오.

HDD 성능을 저하시킬 수 있는 요인

처리량 최적화 HDD(st1) 또는 콜드 HDD(sc1) 볼륨의 스냅샷을 생성하는 경우, 스냅샷이 진행되는 동안 성능이 볼륨의 기준 값까지 떨어질 수 있습니다. 이 동작은 이러한 볼륨 유형에만 해당합니다. 성능을 제한할 수 있는 다른 요소로는 인스턴스가 지원할 수 있는 수준 이상의 처리량을 구동하는 경우, 스냅샷에서 생성된 볼륨을 초기화하는 동안의 성능 저하, 볼륨에 소량의 랜덤 I/O가 과도하게 많은 경우 등을 들 수 있습니다. HDD 볼륨의 처리량 계산에 관한 자세한 내용은 Amazon EBS 볼륨 유형 섹션을 참조하세요.

애플리케이션이 충분한 I/O 요청을 보내지 않는 경우에도 성능이 영향을 받을 수 있습니다. 볼륨의 대기열 길이와 I/O 크기를 보면 확인할 수 있습니다. 대기열 길이는 애플리케이션에서 볼륨으로 보내는 I/O 요청 중 대기 중인 요청의 수입니다. 일관성을 극대화하기 위해 HDD 지원 볼륨은 1MiB 순차 I/O를 수행하는 동안 4 이상의 대기열 길이(반올림)를 유지해야 합니다. 볼륨의 일관된 성능을 유지하는 방법에 대한 자세한 내용은 Amazon EBS I/O 기능 및 모니터링 섹션을 참조하세요.

st1 및 sc1에서 처리량이 많은 읽기 중심 워크로드의 미리 읽기 향상(Linux 인스턴스에만 해당)

일부 워크로드는 읽기 중심이며 운영 체제 페이지 캐시를 통해(예: 파일 시스템에서) 블록 디바이스에 액세스합니다. 이 경우 최대 처리량을 획득하려면 미리 읽기 설정을 1MiB로 구성하는 것이 좋습니다. 이것은 HDD 볼륨에만 적용되는 블록 디바이스별 설정입니다.

블록 디바이스의 현재 미리읽기 값을 검사하려면 다음 명령을 사용합니다.

```
$ sudo blockdev --report /dev/<device>
```

블록 디바이스 정보는 다음 형식으로 반환됩니다.

```
RO    RA   SSZ   BSZ   StartSec            Size   Device
rw   256   512  4096       4096      8587820544   /dev/<device>
```

보기의 디바이스는 256바이트(기본값)의 미리 읽기 값을 보고합니다. 이 값에 섹터 크기(512바이트)를 곱하면 미리읽기 버퍼의 크기를 구할 수 있습니다(이 경우에는 128KiB). 버퍼 값을 1MiB로 설정하려면 다음 명령을 사용합니다.

```
$ sudo blockdev --setra 2048 /dev/<device>
```

첫 번째 명령을 다시 실행해서 현재 미리읽기 설정에 2,048이 표시되는지 확인합니다.

워크로드가 대용량 순차 I/O로 구성된 경우에만 이 설정을 사용합니다. 대부분이 소량 랜덤 I/O로 구성된 경우 이 설정은 실제로 성능을 저하시킵니다. 일반적으로 워크로드의 대부분이 소량 또는 임의 I/O로 구성된 경우 st1 또는 sc1 볼륨보다 범용 SSD(gp2 및 gp3) 볼륨을 사용하는 것이 좋습니다.

최신 Linux 커널 사용(Linux 인스턴스에만 해당)

간접 서술자를 지원하는 최신 Linux 커널을 사용합니다. 현재 세대 EC2 인스턴스뿐만 아니라 Linux 커널 3.8 이상 버전도 모두 이 지원을 제공합니다. 평균 I/O 크기가 44KiB 정도인 경우에는 간접 서술자가

지원되지 않는 인스턴스 또는 커널을 사용 중일 수 있습니다. Amazon CloudWatch 측정치에서 평균 I/O 크기를 도출하는 방법에 관한 내용은 Amazon EBS I/O 기능 및 모니터링 섹션을 참조하세요.

st1 또는 sc1 볼륨의 처리량을 극대화하려면 xen_blkfront.max 파라미터(Linux 커널 버전 4.6 미만) 또는 xen_blkfront.max_indirect_segments 파라미터(Linux 커널 버전 4.6 이상)에 값 256을 적용하는 것이 좋습니다. 해당 파라미터는 OS 부팅 명령줄에서 설정할 수 있습니다.

예를 들어 이전 커널을 사용하는 Amazon Linux AMI에서 /boot/grub/menu.lst의 GRUB 구성에 있는 커널 라인의 끝에 이것을 추가할 수 있습니다.

```
kernel /boot/vmlinuz-4.4.5-15.26.amzn1.x86_64 root=LABEL=/ console=ttyS0
 xen_blkfront.max=256
```

이후 커널의 경우 명령은 다음과 같을 것입니다.

```
kernel /boot/vmlinuz-4.9.20-11.31.amzn1.x86_64 root=LABEL=/ console=tty1 console=ttyS0
 xen_blkfront.max_indirect_segments=256
```

이 설정을 적용하려면 인스턴스를 재부팅합니다.

자세한 내용은 반가상화 AMIs에 대해 GRUB 구성을 참조하세요. 다른 Linux 배포판, 특히 GRUB 부트로더를 사용하지 않는 경우에는 다른 방식으로 커널 파라미터를 조정해야 할 수 있습니다.

EBS I/O 특성에 관한 자세한 내용은 이 주제를 다룬 Amazon EBS: Designing for Performance re:Invent 발표를 참조하세요.

RAID 0을 사용하여 인스턴스 리소스 활용도 극대화

일부 인스턴스 유형은 단일 EBS 볼륨에 대해 프로비저닝할 수 있는 것보다 많은 I/O 처리량을 구동할 수 있습니다. 여러 볼륨을 RAID 0 구성으로 함께 조인하여 이 인스턴스에 사용 가능한 대역폭을 제공할 수 있습니다. 자세한 내용은 Amazon EBS 및 RAID 구성 단원을 참조하십시오.

Amazon EBS 볼륨 성능 모니터링

Amazon CloudWatch, 상태 확인 및 EBS 세부 성능 통계를 사용하여 Amazon EBS 볼륨의 성능을 모니터링하고 분석할 수 있습니다. 자세한 내용은 Amazon EBS에 대한 Amazon CloudWatch 지표 및 Amazon EBS 세부 성능 통계 섹션을 참조하세요.

Amazon EBS 최적화

Amazon EBS 최적화 인스턴스는 최적화된 구성 스택을 사용하며 Amazon EBS I/O를 위한 전용 용량을 추가로 제공합니다. 이 최적화는 Amazon EBS I/O와 인스턴스의 다른 트래픽 간의 경합을 최소화하여 EBS 볼륨에 최상의 성능을 제공합니다.

EBS 최적화 인스턴스는 Amazon EBS에 전용 대역폭을 제공합니다. EBS 최적화 인스턴스에 연결된 경우:

- `io2` Block Express 볼륨은 16KiB I/O 작업에 500마이크로초 미만의 평균 지연 시간을 제공하도록 설계되었습니다. `io2` 또한 Block Express 볼륨은 범용 볼륨에 비해 더 나은 이상 지연 시간을 제공하여 800마이크로초를 초과하는 I/O의 빈도를 10배 이상 줄입니다. `io1` 및 `io2` 볼륨은 지정된 해의 99.9% 시간 동안 프로비저닝된 IOPS 성능의 90% 이상을 제공하도록 설계되었습니다.
- `gp2` 및 `gp3` 볼륨은 지정된 해의 99% 시간 동안 프로비저닝된 IOPS 성능의 90% 이상을 제공하도록 설계되었습니다.
- `st1` 및 `sc1` 볼륨은 지정된 해의 99% 시간 동안 예상 처리량 성능의 90% 이상을 제공합니다.

매 시간 총 처리량 목표 99%를 달성하기 위해, 준수하지 않는 기간은 대략적으로 균등하게 분산됩니다. 자세한 내용은 Amazon EBS 볼륨 유형 단원을 참조하십시오.

I/O 크기가 클수록 지연 시간이 증가하므로 워크로드의 I/O 크기는 관찰된 평균 지연 시간에 영향을 미칩니다. 예를 들어, `io2` Block Express 볼륨은 16KiB I/O 작업에 500마이크로초 미만의 평균 지연 시간을 제공하도록 설계되었습니다.

자세한 내용을 알아보려면 Amazon EC2 사용 설명서의 Amazon EBS 최적화 인스턴스를 참조하세요.

Amazon EBS 볼륨 초기화

EBS 스냅샷 또는 다른 EBS 볼륨(볼륨 복사본)에서 Amazon EBS 볼륨을 생성할 때는 먼저 데이터 블록을 볼륨에 작성해야 액세스할 수 있습니다. 스냅샷에서 생성된 볼륨의 경우 데이터 블록을 Amazon S3에서 새 볼륨으로 다운로드해야 합니다. 볼륨 복사본의 경우 데이터 블록을 소스 볼륨에서 볼륨 복사본으로 복사해야 합니다. 이 프로세스를 볼륨 초기화라고 합니다. 이 시간 동안, 초기화되는 볼륨에 I/O 지연 시간이 증가하고 성능이 저하될 수 있습니다. 모든 스토리지 블록을 다운로드하고 볼륨에 기록한 후에만 최대 볼륨 성능이 구현됩니다.

> **Note**
>
> 빈 볼륨은 생성 후 즉시 최대 성능을 제공하며 초기화할 필요가 없습니다.

기본 볼륨 초기화 속도는 초기화 프로세스 내내 변동하므로 완료 시간을 예측할 수 없습니다. 볼륨 초기화와 관련된 성능 영향을 최소화하기 위해 다음과 같은 옵션을 사용할 수 있습니다.

주제

- 볼륨 초기화에 Amazon EBS 프로비저닝 속도 사용
- 빠른 스냅샷 복원이 활성화된 스냅샷을 사용합니다.
- 볼륨 생성 후 수동 초기화
- Amazon EBS 볼륨 초기화 상태 모니터링

> **Note**
>
> 볼륨 복사본에는 볼륨 초기화 및 빠른 스냅샷 복원을 위한 Amazon EBS 프로비저닝된 속도가 지원되지 않습니다. 자세한 내용은 볼륨 복사본 초기화를 참조하세요.

볼륨 초기화에 Amazon EBS 프로비저닝 속도 사용

스냅샷에서 Amazon EBS 볼륨을 생성할 때, 선택적으로 100~300MiB/s 범위의 볼륨 초기화를 위한 Amazon EBS 프로비저닝된 속도(볼륨 초기화 속도)를 지정할 수 있습니다. 볼륨 초기화 속도를 지정하는 경우, 스냅샷 블록이 Amazon S3에서 다운로드되고 생성 후 지정된 속도로 볼륨에 기록됩니다. 이는 예측 가능한 시간 내에 완전히 초기화되고 성능이 완전히 발휘되는 볼륨을 생성할 수 있도록 합니다.

볼륨 초기화 속도를 사용하면 여러 볼륨을 동시에 생성하고 예측 가능한 시간 내에 모든 볼륨을 초기화해야 할 때 특히 유용합니다.

> **Note**
>
> 볼륨 초기화를 위한 Amazon EBS 프로비저닝 속도는 모든 Amazon EBS 볼륨 유형과 Amazon EC2 Mac 인스턴스를 포함하는 모든 Amazon EC2 인스턴스 유형에서 지원됩니다.

다음에 대해 볼륨 초기화 속도를 지정할 수 있습니다.

- 개별 볼륨 생성 요청
- 인스턴스 시작 요청의 EBS 볼륨 블록 디바이스 매핑
- 시작 템플릿의 EBS 볼륨 블록 디바이스 매핑
- 루트 볼륨 교체 작업으로 생성된 EBS 볼륨
- Amazon EKS 클러스터(EBS CSI 드라이버에서 생성) 및 Amazon ECS 클러스터의 EBS 볼륨

주제

- 작동 방식
- 고려 사항
- 할당량
- 결제
- 볼륨 초기화 속도 지정

작동 방식

볼륨 초기화 속도로 볼륨을 생성하면 지정한 속도로 스냅샷 블록이 Amazon S3에서 볼륨으로 다운로드됩니다.

볼륨을 초기화하는 데 소요되는 시간은 다음에 따라 달라집니다.

- 생성 중인 볼륨의 크기가 아닌 스냅샷 데이터의 크기.

> **Tip**
> 스냅샷의 데이터 크기를 찾으려면 describe-snapshots 명령 출력의 `FullSnapshotSizeInBytes` 필드 또는 콘솔의 전체 스냅샷 크기 필드를 확인합니다.

- 지정한 볼륨 초기화 속도

예를 들어 10GiB의 데이터가 있는 스냅샷을 사용하여 20GiB 볼륨을 생성하고 볼륨 초기화 속도를 300MiB/s로 지정하는 경우, 볼륨이 약 34.1초(10GiB ÷ 300MiB/s = 34.1초) 내에 완전히 초기화됩니다. 마찬가지로 동일한 스냅샷과 볼륨 초기화 속도로 10개의 볼륨을 동시에 생성하면 10개 볼륨 모두 34.1초 내에 완전히 초기화됩니다.

고려 사항

- 볼륨 초기화 속도는 100~300MiB/s로 지정할 수 있습니다.
- 볼륨 초기화 속도를 지정하면 요금 및 완료 시간은 볼륨의 크기가 아닌 스냅샷 데이터의 크기와 지정한 속도를 기반으로 합니다. 자세한 내용은 결제 단원을 참조하십시오.
- Amazon EBS는 99%의 시간 동안 지정한 볼륨 초기화 속도의 10% 이내인 평균 속도를 제공합니다.
- 볼륨 초기화 속도를 지정하고 빠른 스냅샷 복원이 활성화된 스냅샷을 사용하면 Amazon EBS가 빠른 스냅샷 복원 대신 지정된 속도를 사용합니다. 대신 빠른 스냅샷 복원을 사용하려면 볼륨 초기화 속도를 지정하지 마세요.
- 용량 제약 또는 할당량 초과로 인해 Amazon EBS가 지정된 볼륨 초기화 속도로 볼륨을 초기화할 수 없는 경우 요청이 실패합니다.
- 로컬 영역 AWS Outposts또는 Wavelength 영역에서 생성된 볼륨에 대해서는 볼륨 초기화 속도를 지정할 수 없습니다.

할당량

동시 볼륨 생성 요청에서 요청할 수 있는 누적 볼륨 초기화 속도는 5,000MiB/s로 제한됩니다. 예를 들어, 100MiB/s 속도의 동시 볼륨 생성 요청 50개(동시 요청 50개 x 100MiB/s 속도) 또는 200MiB/s 속도의 동시 요청 25개(동시 요청 25개 x 200MiB/s 속도)가 가능합니다. 이 한도는 리전별로 적용됩니다. 이 한도를 초과하면 요청이 실패합니다. 진행 중인 요청 중 일부가 완료될 때까지 기다리거나 할당량 증가를 요청하세요. 자세한 내용은 Amazon EBS 할당량 단원을 참조하십시오.

결제

볼륨 초기화 속도로 볼륨을 생성하면 선택한 초기화 티어에 따라 GiB당 요금이 부과됩니다. 속도는 리전에 따라 달라집니다. 선택한 초기화 티어의 요금을 확인하려면 Amazon EBS 요금을 열고 리전을 선택한 다음 볼륨 초기화에 대한 프로비저닝된 요금 섹션을 찾습니다.

볼륨 크기가 아닌 전체 스냅샷 데이터 크기를 기준으로 요금이 청구됩니다. 예를 들어 50GiB의 쓰기 데이터가 포함된 100GiB 볼륨의 스냅샷을 생성하는 경우 전체 스냅샷 데이터 크기는 50GiB입니다. 해당 스냅샷을 사용하여 볼륨을 생성하고 볼륨 초기화 속도를 지정하는 경우 요금은 50GiB를 기준으로 합니다.

> **Tip**
>
> 전체 스냅샷 데이터 크기를 찾으려면 describe-snapshots 명령 출력의 `FullSnapshotSizeInBytes` 필드 또는 콘솔의 전체 스냅샷 크기 필드를 확인합니다.

공식은 다음과 같습니다.

```
Rate for the selected initialization tier in the Region × full snapshot data size
```

볼륨이 `active` 상태가 되는 즉시 전체 금액이 청구됩니다. 실패한 요청에는 요금이 청구되지 않습니다.

초기화가 완료되기 전에 볼륨을 삭제해도 요청된 볼륨 초기화 요금이 계속 청구됩니다.

볼륨 초기화 속도 지정

다음 방법 중 하나를 사용하여 볼륨 초기화 속도로 스냅샷에서 볼륨을 생성합니다.

Console

볼륨을 생성하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 볼륨을 선택한 다음 볼륨 생성을 선택합니다.
3. 볼륨 유형(Volume type)에서 생성할 볼륨 유형을 선택합니다.
4. 크기에 소스 스냅샷의 볼륨 크기보다 크거나 같은 크기를 입력합니다.
5. 가용 영역에서 볼륨을 생성할 가용 영역을 선택합니다.
6. 스냅샷 ID에서 사용할 스냅샷을 선택합니다.
7. 볼륨 초기화 속도에 `100-300`MiB/s의 속도를 입력합니다. 요금을 지정하면 추가 요금이 발생합니다. 기본 초기화 속도 또는 빠른 스냅샷 복원을 사용하려면 선택한 스냅샷에 대해 활성화된 경우이 필드를 비워 둡니다.

   > **Note**
   >
   > 속도 및 전체 스냅샷 데이터 크기가 초기화 시간에 미치는 영향에 대한 자세한 내용은 섹션을 참조하세요작동 방식. 요금에 대한 자세한 내용은 결제 섹션을 참조하세요.

8. 기타 필수 볼륨 설정을 구성하고 필요에 따라 태그를 추가합니다.
9. 볼륨 생성을 선택합니다.
10. 볼륨 그리드에서 볼륨의 초기화 상태 및 진행 상황을 확인합니다.

AWS CLI

볼륨을 생성하려면

create-volume 명령을 사용합니다. 다음 예시에서는 gp3 볼륨 초기화 속도가 300 MiB/s인 지정된 가용 영역의 스냅샷에서 볼륨을 생성합니다. 예제 리전, 가용 영역 및 스냅샷 ID를 리소스 값으로 바꿉니다.

```
aws ec2 create-volume \
    --region us-east-1 \
    --availability-zone us-east-1a \
    --volume-type gp3 \
    --snapshot-id snap-0abcdef1234567890 \
    --volume-initialization-rate 300
```

--size를 생략하면 볼륨은 소스 스냅샷의 볼륨 크기를 사용합니다. --size를 지정하는 경우 값은 소스 스냅샷의 볼륨 크기보다 크거나 같아야 합니다.

PowerShell

볼륨을 생성하려면

New-EC2Volume cmdlet을 사용합니다. 다음 예시에서는 gp3 볼륨 초기화 속도가 300 MiB/s인 지정된 가용 영역의 스냅샷에서 볼륨을 생성합니다. 명령은 반환된 볼륨 객체를에 저장합니다$volume. 예제 리전, 가용 영역 및 스냅샷 ID를 리소스 값으로 바꿉니다.

```
$volume = New-EC2Volume `
    -Region us-east-1 `
    -AvailabilityZone us-east-1a `
    -VolumeType gp3 `
    -SnapshotId snap-0abcdef1234567890 `
    -VolumeInitializationRate 300
```

초기화 모니터링

볼륨 초기화 정보를 업데이트하는 데 최대 5분이 걸릴 수 있습니다. 초기화 모니터링에 대한 지침은 섹션을 참조하세요Amazon EBS 볼륨 초기화 상태 모니터링.

빠른 스냅샷 복원이 활성화된 스냅샷을 사용합니다.

빠른 스냅샷 복원이 활성화된 스냅샷에서 볼륨을 생성하는 경우, 볼륨이 생성 시 완전히 초기화되고 즉시 전체 성능을 제공합니다. 빠른 스냅샷 복원의 사용에 대한 자세한 내용은 Amazon EBS 빠른 스냅샷 복원 섹션을 참조하세요.

볼륨 생성 후 수동 초기화

Amazon EBS 볼륨을 생성한 후 수동으로 초기화하여 볼륨 초기화의 성능 영향을 최소화할 수 있습니다.

다음 절차를 사용하여 Amazon EBS 볼륨을 생성 후 수동으로 초기화할 수 있습니다.

> **Important**
>
> 스냅샷에서 생성된 Provisioned IOPS SSD 볼륨을 초기화할 경우 볼륨의 성능이 예상 수준보다 50퍼센트 이하로 떨어질 수 있으며, 이로 인해 볼륨의 I/O 성능 상태 확인에 `warning` 상태가 표시될 수 있습니다. 이는 예상된 동작이므로 초기화 중에는 Provisioned IOPS SSD 볼륨에 대한 `warning` 상태를 무시해도 됩니다. 자세한 내용은 Amazon EBS 볼륨 상태 확인 단원을 참조하십시오.

Linux 인스턴스

Linux의 스냅샷에서 생성된 볼륨을 초기화하려면

1. 새로 복원된 볼륨을 Linux 인스턴스에 연결합니다.
2. lsblk 명령을 사용하여 인스턴스의 블록 디바이스를 나열합니다.

   ```
   $ lsblk
   NAME  MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
   xvdf  202:80   0  30G  0 disk
   xvda1 202:1    0   8G  0 disk /
   ```

   여기서 새로운 볼륨인 `/dev/xvdf`가 연결되었지만 마운트되지는 않았음을 확인할 수 있습니다. `MOUNTPOINT` 열 아래에 나열된 경로가 없기 때문입니다.
3. dd 또는 fio 유틸리티를 사용하여 디바이스의 모든 블록을 읽습니다. dd 명령은 Linux 시스템에 기본으로 설치되지만, fio는 다중 스레드 읽기를 허용하기 때문에 훨씬 더 빠릅니다.

> **Note**
>
> 이 단계는 EC2 인스턴스 대역폭, 볼륨에 대해 프로비저닝된 IOPS 및 볼륨 크기에 따라 몇 분에서 몇 시간까지 걸릴 수 있습니다.

[dd] `if`(입력 파일) 파라미터는 초기화할 드라이브로 설정해야 합니다. `of`(파일 출력) 파라미터를 Linux null 가상 디바이스인 `/dev/null`로 설정해야 합니다. `bs` 파라미터는 읽기 작업의 블록 크기를 설정합니다. 최적의 성능을 얻으려면 이 값을 1MB로 설정해야 합니다.

> **Important**
>
> dd를 잘못 사용하면 볼륨 데이터가 쉽게 삭제될 수 있습니다. 아래 예제 명령을 정확하게 따라야 합니다. 읽고 있는 디바이스의 이름에 따라 `if=/dev/`*`xvdf`* 파라미터만 다를 수 있습니다.

```
$ sudo dd if=/dev/xvdf of=/dev/null bs=1M
```

[fio] 시스템에 fio가 설치되어 있는 경우, 다음 명령을 사용하여 볼륨을 초기화할 수 있습니다. `--filename`(입력 파일) 파라미터는 초기화할 드라이브로 설정해야 합니다.

```
$ sudo fio --filename=/dev/xvdf --rw=read --bs=1M --iodepth=32 --ioengine=libaio --direct=1 --name=volume-initialize
```

fio을 Amazon Linux에 설치하려면 다음 명령을 사용하십시오.

```
sudo yum install -y fio
```

Ubuntu에 fio를 설치하려면 다음 명령을 사용합니다.

```
sudo apt-get install -y fio
```

작업이 끝나면 읽기 작업에 대한 보고서가 나타납니다. 이제 볼륨을 사용할 준비가 되었습니다. 자세한 내용은 Amazon EBS 볼륨을 사용할 수 있도록 만들기 단원을 참조하십시오.

Windows 인스턴스

어느 도구든 사용하기 전에 다음과 같이 시스템의 디스크에 관한 정보를 수집하세요.

시스템 디스크에 대한 정보를 수집하려면

1. wmic 명령을 사용하여 시스템에서 사용 가능한 디스크를 나열합니다.

   ```
   wmic diskdrive get size,deviceid
   ```

   다음은 예제 출력입니다.

   ```
   DeviceID            Size
   \\.\PHYSICALDRIVE2  80517265920
   \\.\PHYSICALDRIVE1  80517265920
   \\.\PHYSICALDRIVE0  128849011200
   \\.\PHYSICALDRIVE3  107372805120
   ```

2. dd 또는 fio를 사용하여 초기화할 디스크를 식별합니다. `C:` 드라이브는 `\\.\PHYSICALDRIVE0`에 있습니다. 어떤 드라이브 번호를 사용해야 하는지 확실하지 않은 경우 `diskmgmt.msc` 유틸리티를 사용하여 드라이브 문자를 디스크 드라이브 번호와 비교합니다.

Use the dd utility

다음 절차를 완료하여 dd를 설치하고 사용하여 볼륨을 초기화합니다.

중요 고려 사항

- 볼륨 초기화는 EC2 인스턴스 대역폭, 볼륨에 프로비저닝된 IOPS 및 볼륨 크기에 따라 몇 분에서 몇 시간까지 걸릴 수 있습니다.
- dd를 잘못 사용하면 볼륨 데이터가 쉽게 삭제될 수 있습니다. 다음 절차를 정확하게 수행하세요.

Windows용 dd를 설치하려면

Windows용 dd 프로그램은 Linux 및 Unix 시스템에 일반적으로 사용할 수 있는 dd 프로그램과 비슷한 환경을 제공하며, 이 프로그램을 사용하여 스냅샷에서 생성된 Amazon EBS 볼륨을 초기화할 수 있습니다. 최신 베타 버전은 `/dev/null` 가상 디바이스를 지원합니다. 이전 버전을 설치하는 경우 `nul` 가상 디바이스를 대신 사용할 수 있습니다. 전체 설명서는 http://www.chrysocome.net/dd에서 제공됩니다.

1. 최신 바이너리 버전의 Windows용 dd를 http://www.chrysocome.net/dd에서 다운로드합니다.
2. (선택 사항) `C:\bin`과 같이 찾기 쉽고 기억하기 쉬운 명령줄 유틸리티용 폴더를 만듭니다. 명령줄 유틸리티용으로 지정된 폴더가 이미 있는 경우 다음 단계에서 해당 폴더를 대신 사용할 수 있습니다.
3. 바이너리 패키지의 압축을 풀고 `dd.exe` 파일을 명령줄 유틸리티 폴더(예: `C:\bin`)에 복사합니다.
4. 명령줄 유틸리티 폴더를 경로 환경 변수에 추가합니다 그러면 해당 폴더에 있는 프로그램을 어디서나 실행할 수 있습니다.
   a. 시작(Start)을 선택하고 컴퓨터(Computer)에서 컨텍스트(오른쪽 클릭) 메뉴를 연 후, 속성(Properties)을 선택합니다.
   b. 고급 시스템 설정(Advanced system settings), 환경 변수(Environment Variables)를 선택합니다.
   c. 시스템 변수(System Variables)에서 경로(Path) 변수를 선택하고 편집(Edit)을 선택합니다.
   d. 변수 값(Variable value)에서 세미콜론과 명령줄 유틸리티 폴더의 위치(**`;C:\bin\`**)를 기존 값 끝에 추가합니다.
   e. 확인(OK)을 선택하여 시스템 변수 편집(Edit System Variable) 창을 닫습니다.
5. 새 명령 프롬프트 창을 엽니다. 이전 단계에서는 현재 명령 프롬프트 창의 환경 변수가 업데이트되지 않습니다. 이전 단계를 완료한 후 지금 여는 명령 프롬프트 창이 업데이트됩니다.

Windows용 dd를 사용하여 볼륨을 초기화하려면

다음 명령을 실행하여 지정된 디바이스에 있는 모든 블록을 읽고 출력을 `/dev/null` 가상 디바이스에 전송합니다. 이 명령은 기존 데이터를 안전하게 초기화합니다.

```
dd if=\\.\PHYSICALDRIVEn of=/dev/null bs=1M --progress --size
```

dd가 볼륨의 끝을 지나 읽기를 시도할 경우 오류가 발생할 수 있습니다. 이 오류는 무시해도 됩니다.

이전 버전의 dd 명령을 사용한 경우 `/dev/null` 디바이스가 지원되지 않습니다. 대신 다음과 같이 `nul` 디바이스를 사용할 수 있습니다.

```
dd if=\\.\PHYSICALDRIVEn of=nul bs=1M --progress --size
```

Use the fio utility

다음 절차를 완료하여 fio를 설치하고 사용하여 볼륨을 초기화합니다.

Windows용 fio를 설치하려면

Windows용 fio 프로그램은 Linux 및 Unix 시스템에 일반적으로 사용할 수 있는 fio 프로그램과 비슷한 환경을 제공하며, 이 프로그램을 사용하여 스냅샷에서 생성된 Amazon EBS 볼륨을 초기화할 수 있습니다. 자세한 내용은 https://github.com/axboe/fio를 참조하세요.

1. 최신 릴리스의 자산을 확장하고 MSI 설치 관리자를 선택하여 fio MSI 설치 관리자를 다운로드합니다.
2. fio를 설치합니다.

Windows용 fio를 사용하여 볼륨을 초기화하려면

1. 다음과 비슷한 명령을 실행하여 볼륨을 초기화합니다.

```
fio --filename=\\.\PHYSICALDRIVEn  --rw=read --bs=1M --iodepth=32 --direct=1 --
name=volume-initialize
```

2. 작업이 완료되면 새 볼륨을 사용할 준비가 된 것입니다. 자세한 내용은 Amazon EBS 볼륨을 사용할 수 있도록 만들기 단원을 참조하십시오.

Amazon EBS 볼륨 초기화 상태 모니터링

스냅샷 또는 다른 볼륨(볼륨 복사본)에서 볼륨을 생성할 때 볼륨 초기화의 상태를 모니터링하여 초기화 프로세스가 완료되었는지 확인할 수 있습니다. 다음과 같은 옵션을 사용하여 볼륨 초기화를 모니터링할 수 있습니다.

주제

- AWS CLI, AWS 용 도구 PowerShell및 Amazon EC2 콘솔
- Amazon EventBridge

AWS CLI, AWS 용 도구 PowerShell및 Amazon EC2 콘솔

AWS CLI, AWS 용 도구 PowerShell및 Amazon EC2 콘솔을 사용하여 볼륨이 생성된 후 언제든지 볼륨 초기화 상태를 확인할 수 있습니다. 다음 정보가 제공됩니다.

- 초기화 유형(AWS CLI 및 AWS 도구PowerShell만 해당) - 사용되는 볼륨 초기화 유형을 나타냅니다. `default` 빠른 스냅샷 복원 및 기본 볼륨 초기화, 볼륨 초기화`provisioned-rate`를 위한 Amazon EBS 프로비저닝된 속도 및 볼륨 복사 초기화`volume-copy`를 위한 입니다.
- 예상 완료 시간(AWS CLI 및 AWS 도구PowerShell만 해당) - 볼륨 초기화를 위한 Amazon EBS 프로비저닝된 속도를 사용하여 생성된 볼륨에만 해당됩니다. 볼륨 초기화 완료까지 남은 시간의 예상치(초)입니다.
- 진행 상황 - 볼륨 초기화 프로세스의 진행 상황을 백분율(0~100)로 표시합니다. 빠른 스냅샷 복원으로 초기화된 볼륨의 경우 생성 즉시 진행률이 100%가 됩니다.
- 초기화 상태 - 볼륨 초기화의 전체 상태(`initializing` 또는 `completed`). 빠른 스냅샷 복원으로 초기화된 볼륨의 경우 생성 즉시 상태가 `completed`가 됩니다.

> **Note**
>
> 볼륨 초기화 정보가 업데이트되는 데는 최대 5분이 걸릴 수 있습니다.

다음 탭을 사용하여 볼륨의 볼륨 초기화 상태를 확인합니다.

Console

볼륨 초기화 상태를 모니터링하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 볼륨을 선택합니다.
3. 볼륨 초기화 상태를 확인할 볼륨을 선택합니다.
4. 그리드와 세부 정보 탭의 초기화 상태 필드는 초기화 상태(진행률 %) 형식으로 진행률 정보를 제공합니다. 예를 들어 초기화 중(75%)와 같습니다.

   가능한 초기화 상태는 초기화와 완료를 포함합니다.

AWS CLI

볼륨 초기화 상태를 모니터링하려면

describe-volume-status AWS CLI 명령을 사용하여 초기화 상태를 확인합니다. 출력에는 볼륨 초기화를 위한 Amazon EBS 프로비저닝된 속도로 생성된 볼륨`EstimatedTimeToCompleteInSeconds`에 대해서만 포함됩니다.

예를 들어 다음 명령은 볼륨 초기화를 위한 Amazon EBS 프로비저닝된 속도로 생성된 볼륨 vol-1111111111111111의 초기화 상태를 확인합니다.

```
aws ec2 describe-volume-status --volume-ids vol-1111111111111111
```

다음은 예제 출력입니다.

```
{
    "VolumeStatuses": [
        {
            "Actions": [],
            "AvailabilityZone": "us-east-1a",
            "Events": [],
            "VolumeId": "vol-1111111111111111",
            "VolumeStatus": {
                "Details": [
                    {
                        "Name": "io-enabled",
                        "Status": "passed"
                    },
                    {
                        "Name": "io-performance",
                        "Status": "not-applicable"
                    },
                    {
                        "Name": "initialization-state",
                        "Status": "completed"
                    }
                ],
                "Status": "ok"
            },
            "InitializationStatusDetails": {
                "InitializationType": "provisioned-rate",
                "Progress": 75,
                "EstimatedTimeToCompleteInSeconds": 850
            }
        }
    ]
}
```

PowerShell

볼륨 초기화 상태를 모니터링하려면

Get-EC2VolumeStatus cmdlet을 사용하여 초기화 상태를 확인합니다. 출력에는 볼륨 초기화를 위한 Amazon EBS 프로비저닝된 속도로 생성된 볼륨`EstimatedTimeToCompleteInSeconds`에 대해서만 포함됩니다.

예를 들어 다음 명령은 볼륨 초기화를 위한 Amazon EBS 프로비저닝된 속도로 생성된 볼륨 `vol-1111111111111111`의 초기화 상태를 확인합니다.

```
(Get-EC2VolumeStatus `
    -Region us-east-1 `
    -VolumeId vol-1111111111111111).InitializationStatusDetails |
    Select-Object `
        InitializationType,
        Progress,
        EstimatedTimeToCompleteInSeconds
```

Amazon EventBridge

Amazon EventBridge 이벤트는 볼륨 초기화가 완료된 후 5분 이내에 계정으로 전송됩니다. 이러한 이벤트에 대한 응답으로 프로그래밍 동작을 트리거하는 규칙을 생성할 수 있습니다.

> **Note**
>
> - 이벤트는 최선의 작업을 기반으로 발생됩니다.
> - 초기화가 완료되기 전이나 초기화가 완료된 후 5분 이내에 볼륨을 삭제하는 경우, 이벤트가 수신되지 않을 수 있습니다.

이벤트에 대한 자세한 내용은 EBS 볼륨 초기화 이벤트 섹션을 참조하세요.

EventBridge를 사용하여 볼륨 초기화의 상태를 모니터링하려면

1. Amazon EventBridge 콘솔(https://console.aws.amazon.com/events/)을 엽니다.
2. 규칙, 규칙 생성을 선택합니다.
3. 1단계의 경우 다음을 수행합니다.
   a. 규칙에 대한 이름 및 설명을 지정합니다.
   b. 이벤트 버스에서 이벤트를 수신할 버스를 선택합니다. 사용자 지정 이벤트 버스를 생성하지 않은 경우 기본값을 유지하거나 이벤트 버스 생성을 참조하세요.

c. 규칙 유형에서 이벤트 패턴이 있는 규칙을 유지합니다.

d. 다음을 선택합니다.

4. 2단계의 경우 다음을 수행합니다.

a. Event source에서 AWS 이벤트 또는 EventBridge 파트너 이벤트를 유지합니다.

b. 생성 방법에서 사용자 지정 패턴(JSON 편집기)을 선택합니다.

c. 이벤트 패턴에서 다음을 추가합니다.

```
{
    "detail-type": ["EBS Volume Notification"],
    "source": ["aws.ec2"],
    "detail": {
        "event": ["initializeVolume"],
        "result": ["succeeded"]
    }
}
```

예제 이벤트는 EBS 볼륨 초기화 이벤트을 참조하세요.

d. 다음을 선택합니다.

5. 3단계의 경우 다음을 수행합니다.

a. 대상 유형에서 AWS 서비스를 선택합니다.

b. 대상 선택에서 SNS 주제를 선택하고 주제에서 필요한 주제를 선택합니다. 주제를 생성하지 않은 경우 주제 생성을 참조하세요.

c. 권한에서 실행 역할 사용(권장)을 선택된 상태로 유지합니다.

d. 실행 역할에서 이 특정 리소스에 대해 새 역할 생성을 선택된 상태로 유지하고 기본 역할 이름을 유지합니다.

e. 다음을 선택합니다.

6. 4단계에서 필요한 경우 규칙에 대한 태그를 지정하고 다음을 선택합니다.

7. 5단계에서 규칙을 검토한 다음 규칙 생성을 선택합니다.

구성 가능한 인스턴스 대역폭 가중치

인스턴스 대역폭 구성(IBC)은 Amazon EC2 인스턴스에 대한 Amazon EBS와 VPC 네트워킹 간의 네트워크 대역폭 할당을 조정할 수 있게 해 주는 기능입니다. 이 기능은 특정 대역폭 요구 사항이 있는 워크

로드의 성능을 최적화하는 데 도움이 될 수 있습니다. 인스턴스 대역폭 구성은 일부 인스턴스에서만 지원됩니다. 자세한 내용은 인스턴스 대역폭 가중치 구성을 참조하세요.

EBS 성능의 경우 ebs-1 대역폭 가중치를 사용하면 기본 EBS 대역폭이 25% 증가하고 VPC 네트워킹 대역폭은 동일한 절대량만큼 감소합니다. 이는 더 높은 EBS 처리량이 필요한 I/O 집약적 워크로드에 유용할 수 있습니다.

워크로드를 계획할 때는 I/O 크기와 패턴을 신중하게 고려하세요. 일반적으로 I/O 크기가 작을수록 대역폭 제한의 영향을 덜 받을 수 있는 반면, I/O 크기나 순차적 워크로드가 클수록 대역폭 변경으로 인해 더 많은 영향을 받을 수 있습니다. 선택한 대역폭 가중치로 최적의 성능을 보장하려면 특정 워크로드를 철저히 테스트하는 것이 중요합니다.

고려 사항

- 구성 가능한 인스턴스 대역폭은 일부 인스턴스 유형에서 지원됩니다. 자세한 내용은 지원되는 인스턴스 유형을 참조하세요.
- ebs-1 대역폭 가중치를 사용하면 EBS 대역폭이 최대 25% 증가하여 I/O 집약적 애플리케이션의 성능을 개선할 수 있습니다. 그러나 VPC 네트워킹 대역폭이 동일한 절대량만큼 감소한다는 점에 유의하세요(즉, EBS와 네트워킹 간의 합산된 대역폭 사양은 변경되지 않음).
- 대역폭 가중치 변경은 I/O 성능에 상당한 영향을 미칠 수 있습니다. vpc-1 대역폭 가중치를 사용하면 네트워크 대역폭이 증가하지만 EBS 볼륨의 IOPS가 예상보다 낮아질 수 있습니다. 이는 특히 I/O 크기가 더 큰 경우 IOPS 한도 전에 EBS 대역폭 한도에 도달할 수 있기 때문입니다. 예를 들어, 일반적으로 16KiB I/O 크기에 240,000 IOPS를 지원하는 인스턴스 유형은 vpc-1 대역폭 가중치를 사용할 때 EBS 대역폭 감소로 인해 IOPS가 줄어들 수 있습니다.
- 선택한 대역폭 가중치가 성능 요구 사항을 충족하는지 확인하려면 항상 특정 워크로드를 테스트하세요.
- 인스턴스 시작 중에 대역폭 가중치를 구성하거나 중지된 인스턴스에 대해 가중치를 수정할 수 있습니다. 자세한 내용은 인스턴스 대역폭 가중치 구성을 참조하세요.
- 추가 비용 없이 인스턴스 대역폭 가중치를 구성할 수 있습니다.

Amazon EBS I/O 기능 및 모니터링

지정된 볼륨 구성에서 특정 I/O 특성은 EBS 볼륨의 성능 동작을 구동합니다.

- 범용 SSD(gp2 및 gp3) 및 프로비저닝된 IOPS SSD(io1 및 io2)와 같은 SSD 지원 볼륨은 I/O 작업이 임의든 순차든 관계없이 일관된 성능을 제공합니다.

- 처리량 최적화 HDD(st1) 및 콜드 HDD(sc1)와 같은 HDD 지원 볼륨은 I/O 작업이 대용량이고 순차인 경우에만 최적의 성능을 제공합니다.

SSD 및 HDD 볼륨이 애플리케이션에서 어떻게 작동하는지 이해하려면 볼륨 수요와 가용 IOPS 수량, I/O 작업을 완료하는 데 소요되는 시간, 볼륨의 처리량 제한 사이의 관계를 아는 것이 중요합니다.

주제

- IOPS
- 볼륨 대기열 길이 및 지연 시간
- I/O 크기 및 볼륨 처리량 제한이 없음
- CloudWatch를 사용하여 I/O 특성 모니터링
- 실시간 I/O 성능 통계 모니터링
- 관련 리소스

IOPS

IOPS는 초당 입력/출력 작업을 나타내는 측정 단위입니다. 작업은 KiB로 측정되며 볼륨 유형에서 단일 I/O로 계산되는 데이터의 최대 양은 기반 드라이브 기술에 따라 결정됩니다. I/O 크기는 SSD 볼륨의 경우 256KiB로, HDD 볼륨의 경우 1,024KiB로 제한되는데 SSD 볼륨은 소량 또는 임의 I/O를 HDD 볼륨보다 훨씬 더 효율적으로 처리하기 때문입니다.

소량의 I/O 작업이 물리적으로 연속되는 경우 Amazon EBS는 최대 I/O 크기까지 단일 I/O 작업으로 병합을 시도합니다. 마찬가지로, I/O 작업이 최대 I/O 크기보다 큰 경우 Amazon EBS는 I/O 작업을 더 작은 I/O 작업으로 분할을 시도합니다. 다음 표에 몇 가지 예가 나와 있습니다.

볼륨 유형	최대 I/O 크기	애플리케이션의 I/O 작업	IOPS 수	참고
SSD	256KiB	1 x 1024KiB I/O 작업	4(1,024÷256=4)	Amazon EBS는 1,024KiB I/O 작업을 4개의 작은 256KiB 작업으로 분할합니다.
		8 x 순차적 32KiB I/O 작업	1(8x32=256)	Amazon EBS는 8개의 순차적 32KiB

볼륨 유형	최대 I/O 크기	애플리케이션의 I/O 작업	IOPS 수	참고
				I/O 작업을 하나의 256KiB 작업으로 병합합니다.
		8개의 임의 32KiB I/O 작업	8	Amazon EBS는 임의 I/O 작업을 별도로 계산합니다.
HDD	1,024KiB	1 x 1024KiB I/O 작업	1	I/O 작업이 이미 최대 I/O 크기와 같습니다. 병합되거나 분할되지 않습니다.
		8 x 순차적 128KiB I/O 작업	1(8x128=1,024)	Amazon EBS는 8개의 순차적 128KiB I/O 작업을 하나의 1,024KiB I/O 작업으로 병합합니다.
		8개의 임의 32KiB I/O 작업	8	Amazon EBS는 임의 I/O 작업을 별도로 계산합니다.

그러므로 (3,000 IOPS를 제공하는 `io1` 또는 `io2` 볼륨을 프로비저닝하거나 `gp2` 볼륨의 크기를 1,000GiB로 지정하거나 `gp3` 볼륨을 사용하는 방법으로) 3,000 IOPS를 지원하는 SSD 지원 볼륨을 생성하고 충분한 대역폭을 제공할 수 있는 EBS 최적화 인스턴스에 연결할 경우, 초당 최대 3,000 I/O 데이터 전송이 가능하며 처리량은 I/O 크기에 따라 결정됩니다.

볼륨 대기열 길이 및 지연 시간

볼륨 대기열 길이는 디바이스에 대해 보류 중인 I/O 요청 수입니다. 지연 시간은 I/O 작업의 실제 종단 간 클라이언트 시간입니다. 다시 말해 EBS로 I/O를 전송한 후 EBS로부터 I/O 읽기 또는 쓰기가 완료되었다는 승인을 받기까지 소요된 시간입니다. 대기열 길이를 I/O 크기 및 지연 시간에 따라 정확히 보정

하여, 게스트 운영 체제나 EBS로 연결되는 네트워크 링크에 병목 현상이 발생하지 않도록 해야 합니다.

최적의 대기열 길이는 워크로드마다 다른데, IOPS 및 지연 시간에 대한 특정 애플리케이션의 민감도에 따라 결정됩니다. 워크로드가 EBS 볼륨에 대해 사용 가능한 성능을 전부 사용할 만큼 충분한 I/O 요청을 제공하지 않는 경우, 프로비저닝된 처리량이나 IOPS를 볼륨이 제공하지 못할 수 있습니다.

트랜잭션 집약적 애플리케이션은 I/O 지연 시간 증가에 민감하며, SSD 기반 볼륨에 적합합니다. 대기열 길이를 줄이고 볼륨에서 사용할 수 있는 IOPS 개수를 늘리면 높은 IOPS를 유지하는 동시에 지연 시간을 단축할 수 있습니다. 볼륨이 수용할 수 있는 수준보다 높은 IOPS를 계속 구동하면 I/O 지연 시간이 길어질 수 있습니다. 일관성을 극대화하려면 볼륨이 1분 동안 프로비저닝된 1,000 IOPS마다 1개의 평균 대기열 깊이(가장 가까운 정수로 반올림)를 유지해야 합니다. 예를 들어 3,000 IOPS로 프로비저닝된 볼륨의 경우 대기열 깊이 평균은 3이어야 합니다.

처리량 집약적인 애플리케이션은 I/O 지연 시간 증가에 덜 민감하며, HDD 기반 볼륨에 적합합니다. 대용량 순차 I/O를 수행할 때 대기열 길이를 길게 유지하면 HDD 지원 볼륨에서 높은 처리량을 유지할 수 있습니다.

I/O 크기 및 볼륨 처리량 제한이 없음

SSD 지원 볼륨의 경우, I/O 크기가 매우 크면 볼륨 처리량 제한에 도달하기 때문에 프로비저닝한 것보다 IOPS가 적을 수 있습니다. 예를 들어 버스트 크레딧을 사용할 수 있는 1,000GiB 미만의 gp2 볼륨에는 3,000 IOPS 제한과 250MiB/s의 볼륨 처리량 제한이 있습니다. 256KiB I/O 크기를 사용하는 경우, 볼륨은 1000 IOPS에서 처리량 제한에 도달합니다(1000 x 256KiB = 250MiB). I/O 크기가 작다면(예: 16KiB), 처리량이 250MiB/s에 훨씬 못 미치기 때문에 동일한 볼륨이 3,000 IOPS를 유지할 수 있습니다. (이 예제는 볼륨의 I/O가 인스턴스의 처리량 제한에 도달하지 않는다고 가정합니다.) 각 EBS 볼륨 유형의 처리량 제한에 대한 자세한 내용은 Amazon EBS 볼륨 유형 섹션을 참조하세요.

소용량 I/O 작업의 경우, 인스턴스 내에서 측정했을 때 프로비저닝된 IOPS 값보다 큰 값을 관찰할 수 있습니다. 인스턴스 운영 체제가 소용량 I/O 작업을 Amazon EBS로 전달하기 전 대용량 작업에 병합할 때 이런 결과가 발생합니다.

워크로드가 HDD 지원 `st1` 및 `sc1` 볼륨의 순차 I/O를 사용한다면 인스턴스 내에서 측정했을 때 예상보다 높은 IOPS를 관찰할 수 있습니다. 인스턴스 운영 체제가 순차 I/O를 병합하고 1,024KiB 크기 단위로 계산되는 경우에 이런 결과가 발생합니다. 워크로드가 소용량 또는 랜덤 I/O를 사용하는 경우 예상보다 적은 처리량을 관찰할 수 있습니다. 이는 비순차적인 각 무작위 I/O가 총 IOPS 수에 포함되므로 볼륨의 IOPS 한도에 예상보다 빨리 도달할 수 있기 때문입니다.

EBS 볼륨 유형이 무엇이든 구성에서 예상되는 IOPS 또는 처리량이 발생하지 않는 경우 EC2 인스턴스 대역폭이 제한 요소가 아닌지 확인합니다. 최적의 성능을 위해 항상 현재 세대 EBS 최적화 인스턴

스(또는 10Gb/s 네트워크 연결을 포함한 인스턴스)를 사용해야 합니다. EBS 볼륨에 충분한 I/O를 구동하고 있지 않은 경우에도 IOPS가 예상과 다를 수 있습니다.

CloudWatch를 사용하여 I/O 특성 모니터링

각 볼륨의 CloudWatch 볼륨 지표로 이러한 I/O 특성을 모니터링할 수 있습니다.

멈춘 I/O의 모니터링

`VolumeStalledIOCheck`는 EBS 볼륨의 상태를 모니터링하여 볼륨이 손상된 시점을 확인합니다. 지표는 EBS 볼륨이 I/O 작업을 완료할 수 있는지 여부에 따라 `0`(통과) 또는 `1`(실패) 상태를 반환하는 바이너리 값입니다.

지표가 실패하면 `VolumeStalledIOCheck` AWS 가 문제를 해결할 때까지 기다리거나 영향을 받는 볼륨을 교체하거나 볼륨이 연결된 인스턴스를 중지했다가 다시 시작하는 등의 작업을 수행할 수 있습니다. 대부분의 경우 이 지표가 실패하면 EBS는 몇 분 내에 볼륨을 자동으로 진단하고 복구합니다. 에서 I/O 일시 중지 작업을 사용하여 제어된 실험 AWS Fault Injection Service 을 실행하여이 지표를 기반으로 아키텍처 및 모니터링을 테스트하여 스토리지 장애에 대한 복원력을 개선할 수 있습니다.

볼륨의 I/O 지연 시간 모니터링

Amazon EBS 볼륨의 읽기 및 쓰기 작업에 대한 평균 지연 시간을 각각 `VolumeAvgReadLatency` 및 `VolumeAvgWriteLatency` 지표를 사용하여 모니터링할 수 있습니다. 의 지연 시간 주입 작업을 사용하여 제어된 실험 AWS Fault Injection Service 을 실행하여이 지표를 기반으로 아키텍처 및 모니터링을 테스트하여 스토리지 성능 저하에 대한 복원력을 개선할 수 있습니다.

I/O 지연 시간이 필요한 수준보다 긴 경우 애플리케이션이 볼륨에 프로비저닝한 것보다 높은 IOPS 또는 처리량을 시도하고 있지 않은지 확인하세요. `VolumeAvgIOPS` 및 `VolumeAvgThroughput` 지표를 사용하여 1분 내에 볼륨에서 실행되는 평균 IOPS 및 처리량을 모니터링한 다음 볼륨의 프로비저닝된 IOPS 및 처리량과 비교할 수 있습니다. 볼륨이 1분 동안 작업을 실행하지 않는 경우 지표는 `0`의 값을 보고합니다. 1분 미만의 짧은 시간 동안 높은 IOPS 또는 처리량의 버스트가 발생한 경우 볼륨에 미세 버스트가 발생하지만 평균 IOPS 및 처리량 지표는 볼륨의 프로비저닝된 IOPS 또는 처리량 한도보다 성능이 낮다고 보고할 수 있습니다. 주어진 1분 동안 볼륨이 성능 버스트를 경험하는지 여부를 식별하려면 `VolumeIOPSExceededCheck` 및 `VolumeThroughputExceededCheck` 지표를 사용할 수 있습니다. 이러한 지표를 모니터링하여 주어진 1분 동안 워크로드가 볼륨의 프로비저닝된 성능보다 더 큰 IOPS 또는 처리량을 지속적으로 실행하려고 시도했는지 확인할 수 있습니다. 1분 내 언제든지 IOPS가 볼륨의 프로비저닝된 IOPS 성능을 지속적으로 초과하는 경우 `VolumeIOPSExceededCheck` 지표는 `1`을 반환합니다. 1분 내 언제든지 처리량이 볼륨의 프로비저닝된 처리량 성능을 지속적으로 초과하는 경우 `VolumeThroughputExceededCheck` 지표는 `1`을 반환합니다. IOPS 및 처리량이 볼륨의 프로비저닝된 성능 이내인 경우 지표는 `0`을 반환합니다.

IOPS가 볼륨에서 제공할 수 있는 수보다 많이 애플리케이션에 필요한 경우 다음 중 하나를 사용하는 것을 고려해야 합니다.

- 필수 지연 시간 달성에 충분한 IOPS가 프로비저닝되는 gp3, io2 또는 io1 볼륨
- 충분한 기준 IOPS 성능을 제공하는 더 큰 gp2 볼륨

HDD 지원 st1 및 sc1 볼륨은 1,024KiB 최대 I/O 크기를 활용하는 워크로드에서 가장 잘 작동하도록 설계되었습니다. 볼륨의 평균 I/O 크기를 구하려면 VolumeWriteBytes를 VolumeWriteOps 값으로 나눕니다. 읽기 작업에도 같은 계산 방법이 적용됩니다. 평균 I/O 크기는 64KiB 미만이며, st1 또는 sc1 볼륨으로 보내는 I/O 작업의 크기가 큰 경우 성능을 개선해야 합니다.

gp2, **st1** 및 **sc1** 볼륨에 대한 버스트 버킷 밸런스 모니터링

BurstBalance는 gp2, st1, sc1 볼륨에 대한 버스트 버킷 잔고를 남은 잔고에 대한 비유로 표시합니다. 버스트 버킷이 모두 사용되면 볼륨 I/O(gp2 볼륨용) 또는 볼륨 처리량(st1 및 sc1 볼륨용)이 기준 수준으로 스로틀링됩니다. BurstBalance 값을 확인하여 이런 이유로 볼륨이 조절되는지 판단합니다. 사용 가능한 Amazon EBS 지표의 전체 목록은 Amazon EBS에 대한 Amazon CloudWatch 지표 및 Nitro 기반 인스턴스용 Amazon EBS 지표를 참조하세요.

실시간 I/O 성능 통계 모니터링

Nitro 기반 Amazon EC2 인스턴스에 연결된 Amazon EBS 볼륨에 대한 실시간 세부 성능 통계에 액세스할 수 있습니다.

이러한 통계를 결합하여 평균 지연 시간과 IOPS를 도출하거나 I/O 작업이 완료되고 있는지 여부를 확인할 수 있습니다. 또한 애플리케이션이 EBS 볼륨 또는 연결된 인스턴스의 프로비저닝된 IOPS 또는 처리량 한도를 초과한 총 시간을 볼 수 있습니다. 시간 경과에 따른 이러한 통계에서 증가를 추적하면 애플리케이션의 성능을 최적화하기 위해 프로비저닝된 IOPS 또는 처리량 한도를 늘려야 하는지 여부를 식별할 수 있습니다. 또한 상세 성능 통계에는 지연 시간 대역 내에서 완료된 총 I/O 작업 수를 추적하여 I/O 지연 시간의 분포를 제공하는 읽기 및 쓰기 I/O 작업에 대한 히스토그램이 포함됩니다.

자세한 내용은 Amazon EBS 세부 성능 통계 단원을 참조하십시오.

관련 리소스

Amazon EBS I/O 특성에 관한 자세한 내용은 Amazon EBS: Designing for Performance re:Invent 발표를 참조하세요.

Amazon EBS 및 RAID 구성

Amazon EBS를 사용하면 기존 운영 체제 미설치 서버에서 사용 가능한 스탠다드 RAID 구성을 사용할 수 있습니다. 단, 해당 RAID 구성이 인스턴스에 대한 운영 체제에서 지원되어야 합니다. 이는 모든 RAID가 소프트웨어 수준에서 실행되기 때문입니다.

Amazon EBS 볼륨 데이터는 단일 구성 요소의 고장으로 인한 데이터 손실을 방지하기 위해 가용 영역의 여러 서버에 복제됩니다. 이 복제 기능으로 인해 Amazon EBS 볼륨이 일반 상용 디스크 드라이브보다 10배 더 안정적입니다. 자세한 내용은 Amazon EBS 기능을 참조하세요.

내용

- RAID 구성 옵션
- RAID 0 어레이 생성
- RAID 어레이에 볼륨 스냅샷 생성

RAID 구성 옵션

RAID 0 어레이를 생성하면 단일 Amazon EBS 볼륨에서 프로비저닝할 때보다 파일 시스템의 성능이 더 향상됩니다. I/O 성능이 무엇보다 중요할 경우 RAID 0를 사용하세요. RAID 0를 사용할 경우 I/O가 스트라이프의 볼륨에 분산됩니다. 볼륨을 추가하면 처리량 및 IOPS도 그에 따라 바로 추가됩니다. 그러나 스트라이프의 성능은 세트에서 성능이 가장 낮은 볼륨의 성능으로 제한되며 세트에서 단일 볼륨이 손실되면 어레이의 데이터가 완전히 손실됩니다.

RAID 0 어레이의 결과 크기는 어레이 내 볼륨의 크기 합계이고, 대역폭은 어레이 내 볼륨의 가용 대역폭 합계입니다. 예를 들어, 4,000의 프로비저닝된 IOPS가 있는 두 500GiB `io1` 볼륨은 각각 가용 대역폭이 8,000 IOPS이고 처리량이 1,000MB/s인 1,000GiB RAID 0 어레이를 생성합니다.

> **Important**
>
> RAID 5 및 RAID 6는 이 RAID 모드의 패리티 쓰기 작업에서 볼륨에 사용 가능한 IOPS의 일부를 사용하기 때문에 Amazon EBS에 권장되지 않습니다. RAID 어레이의 구성에 따라 이러한 RAID 모드에서는 RAID 0 구성보다 20-30% 더 적은 가용 IOPS를 제공합니다. 이러한 RAID 모드는 비용 증가의 한 요인이기도 합니다. 볼륨 크기와 속도가 동일할 경우 2 볼륨 RAID 0 어레이가 두 배 더 비싼 4 볼륨 RAID 6 어레이보다 더 우수한 성능을 제공합니다.
>
> 또한 RAID 1은 Amazon EBS와 함께 사용하지 않는 것이 좋습니다. RAID 1의 경우 데이터를 동시에 여러 볼륨에 쓰기 때문에 비RAID 구성에 비해 Amazon EC2와 Amazon EBS 사이에 더 큰 대역폭이 필요합니다. 또한 RAID 1은 쓰기 성능 향상 효과를 제공하지 않습니다.

RAID 0 어레이 생성

다음 절차에 따라 RAID 0 어레이를 생성합니다.

고려 사항

- 이 절차를 수행하기 전에 RAID 0 어레이의 크기와 프로비저닝할 IOPS 수를 결정해야 합니다.
- 어레이에 대해 크기 및 IOPS 성능 값이 동일한 볼륨을 생성합니다. EC2 인스턴스의 가용 대역폭을 초과하는 어레이를 생성하지 마세요.
- RAID 볼륨에서는 부팅하지 않아야 합니다. 디바이스 중 하나에 장애가 발생하면 운영 체제를 부팅하지 못할 수 있습니다.

Linux 인스턴스

Linux에서 RAID 0 어레이를 생성하려면

1. 어레이에 대한 Amazon EBS 볼륨을 생성합니다. 자세한 내용은 Amazon EBS 볼륨 생성 단원을 참조하십시오.
2. 어레이를 호스팅할 인스턴스에 Amazon EBS 볼륨을 연결합니다. 자세한 내용은 Amazon EC2 인스턴스에 Amazon EBS 볼륨 연결 섹션을 참조하세요.
3. mdadm 명령을 사용하여 새로 연결된 Amazon EBS 볼륨에서 로직 RAID 디바이스를 생성합니다. *number_of_volumes*에 대한 어레이의 볼륨 수와 *device_name*에 대한 어레이에 있는 각 볼륨의 디바이스 이름(예: `/dev/xvdf`)을 대체합니다. 어레이의 고유 이름으로 *MY_RAID*를 대체할 수도 있습니다.

   > **Note**
   >
   > lsblk 명령으로 인스턴스에 디바이스를 나열하여 디바이스 이름을 찾을 수 있습니다.

   RAID 0 어레이를 생성하려면 다음 명령을 실행합니다(어레이를 스트라이프하려면 `--level=0` 옵션 사용).

   ```
   [ec2-user ~]$ sudo mdadm --create --verbose /dev/md0 --level=0 --name=MY_RAID --raid-devices=number_of_volumes device_name1 device_name2
   ```

> **Tip**
>
> `mdadm: command not found` 오류가 발생하면 `sudo yum install mdadm` 명령을 사용하여 mdadm을 설치합니다.

4. RAID 어레이가 초기화되고 동기화될 때까지 기다립니다. 다음 명령을 사용하여 이 작업의 진행을 추적할 수 있습니다.

```
[ec2-user ~]$ sudo cat /proc/mdstat
```

다음은 예제 출력입니다.

```
Personalities : [raid0]
md0 : active raid0 xvdc[1] xvdb[0]
      41910272 blocks super 1.2 512k chunks

unused devices: <none>
```

일반적으로 다음 명령을 사용하여 RAID 어레이에 대한 자세한 정보를 표시할 수 있습니다.

```
[ec2-user ~]$ sudo mdadm --detail /dev/md0
```

다음은 예제 출력입니다.

```
/dev/md0:
           Version : 1.2
     Creation Time : Wed May 19 11:12:56 2021
        Raid Level : raid0
        Array Size : 41910272 (39.97 GiB 42.92 GB)
      Raid Devices : 2
     Total Devices : 2
       Persistence : Superblock is persistent

       Update Time : Wed May 19 11:12:56 2021
             State : clean
    Active Devices : 2
   Working Devices : 2
    Failed Devices : 0
     Spare Devices : 0
```

```
        Chunk Size : 512K

Consistency Policy : none

              Name : MY_RAID
              UUID : 646aa723:db31bbc7:13c43daf:d5c51e0c
            Events : 0

    Number   Major   Minor   RaidDevice State
       0     202       16        0      active sync   /dev/sdb
       1     202       32        1      active sync   /dev/sdc
```

5. RAID 어레이에서 파일 시스템을 생성하고 이후 해당 파일 시스템에 마운트할 때 사용할 레이블을 지정합니다. 예를 들어, 레이블 *MY_RAID*로 ext4 파일 시스템을 생성하려면 다음 명령을 사용합니다.

```
[ec2-user ~]$ sudo mkfs.ext4 -L MY_RAID /dev/md0
```

애플리케이션의 요구 사항 또는 운영 체제의 제한에 따라 다른 파일 시스템 유형(예: ext3 또는 XFS)을 사용할 수 있습니다. 해당 파일 시스템 생성 명령은 파일 시스템 설명서를 참조하세요.

6. 부팅 시 RAID 어레이가 자동으로 재조립되도록 하려면 RAID 정보를 포함하는 구성 파일을 생성합니다.

```
[ec2-user ~]$ sudo mdadm --detail --scan | sudo tee -a /etc/mdadm.conf
```

> **Note**
>
> Amazon Linux 이외의 Linux 배포판을 사용하는 경우 이 명령을 수정해야 할 수도 있습니다. 예를 들어 파일을 다른 위치에 배치하거나 `--examine` 파라미터를 추가해야 할 수 있습니다. 자세한 내용을 보려면 Linux 인스턴스에서 man mdadm.conf를 실행하세요.

7. 새 RAID 구성을 위해 블록 디바이스 모듈을 올바르게 미리 로드하려면 새 ramdisk 이미지를 만듭니다.

```
[ec2-user ~]$ sudo dracut -H -f /boot/initramfs-$(uname -r).img $(uname -r)
```

8. RAID 어레이에 대한 마운트 지점을 생성합니다.

```
[ec2-user ~]$ sudo mkdir -p /mnt/raid
```

9. 마지막으로 생성한 탑재 지점에 RAID 디바이스를 탑재합니다.

```
[ec2-user ~]$ sudo mount LABEL=MY_RAID /mnt/raid
```

이제 RAID 디바이스를 사용할 준비가 되었습니다.

10. (선택 사항) 시스템을 재부팅할 때마다 이 Amazon EBS 볼륨을 탑재하려면 디바이스에 대한 항목을 /etc/fstab 파일에 추가합니다.

a. 수정 도중 실수로 이 파일이 손상되거나 삭제되는 경우에 대비하여 /etc/fstab 파일의 백업을 생성합니다.

```
[ec2-user ~]$ sudo cp /etc/fstab /etc/fstab.orig
```

b. 자주 사용하는 텍스트 편집기를 사용하여 /etc/fstab 파일(예: nano 또는 vim)을 엽니다.

c. "UUID="로 시작하는 줄을 주석으로 처리하고, 파일 끝에 다음 형식으로 RAID 볼륨 파일의 새 줄을 추가합니다.

```
device_label mount_point file_system_type fs_mntops fs_freq fs_passno
```

이 줄의 마지막 세 필드는 파일 시스템 마운트 옵션, 파일 시스템의 덤프 빈도 및 부팅 시 파일 시스템 확인 순서입니다. 이러한 값이 무엇인지 모르는 경우 아래 예제의 값을 사용합니다 (defaults,nofail 0 2)). /etc/fstab 항목에 대한 자세한 내용은 fstab 매뉴얼 페이지를 참조하세요(명령줄에서 man fstab 입력). 예를 들어, MY_RAID 레이블이 있는 디바이스에 /mnt/raid 탑재 지점에서 ext4 파일 시스템을 탑재하려면 /etc/fstab에 다음 항목을 추가합니다.

> **Note**
>
> 이 볼륨을 연결하지 않고 인스턴스를 부팅하려면(예: 이 볼륨이 서로 다른 인스턴스 사이를 이동할 수 있도록) 볼륨 마운트 시 오류가 있어도 인스턴스가 부팅되도록 하는 nofail 마운트 옵션을 추가해야 합니다. Ubuntu와 같은 Debian 계열 시스템에서는 nobootwait 마운트 옵션도 추가해야 합니다.

```
LABEL=MY_RAID       /mnt/raid   ext4    defaults,nofail        0       2
```

d. /etc/fstab에 새 항목을 추가한 다음에는 해당 항목이 작동하는지 확인해야 합니다. 그런 다음 sudo mount -a 명령을 실행하여 /etc/fstab에 있는 모든 파일 시스템을 탑재합니다.

```
[ec2-user ~]$ sudo mount -a
```

이전 명령에서 오류가 발생하지 않으면 /etc/fstab 파일이 정상이고 다음 부팅 시 파일 시스템이 자동으로 탑재됩니다. 명령에서 오류가 발생하면 오류를 검토한 다음 /etc/fstab를 수정합니다.

> **Warning**
>
> /etc/fstab 파일에서 오류가 발생하면 시스템이 부팅되지 않을 수 있습니다. /etc/fstab 파일에서 오류가 발생한 시스템을 종료하지 마십시오.

e. (선택 사항) /etc/fstab 오류 수정 방법을 모르는 경우 다음 명령으로 항상 백업 /etc/fstab 파일을 복원할 수 있습니다.

```
[ec2-user ~]$ sudo mv /etc/fstab.orig /etc/fstab
```

Windows 인스턴스

Windows에서 RAID 0 어레이를 생성하려면

1. 어레이에 대한 Amazon EBS 볼륨을 생성합니다. 자세한 내용은 Amazon EBS 볼륨 생성 단원을 참조하십시오.
2. 어레이를 호스팅할 인스턴스에 Amazon EBS 볼륨을 연결합니다. 자세한 내용은 Amazon EC2 인스턴스에 Amazon EBS 볼륨 연결 섹션을 참조하세요.
3. Windows 인스턴스에 연결합니다. 자세한 내용은 Windows 인스턴스에 연결을 참조하세요.
4. 명령 프롬프트를 열고 diskpart 명령을 입력합니다.

```
diskpart

Microsoft DiskPart version 6.1.7601
Copyright (C) 1999-2008 Microsoft Corporation.
```

```
On computer: WIN-BM6QPPL51CO
```

5. DISKPART 프롬프트에서 다음 명령을 사용하여 사용 가능한 디스크를 나열합니다.

```
DISKPART> list disk

  Disk ###  Status         Size     Free     Dyn  Gpt
  --------  -------------  -------  -------  ---  ---
  Disk 0    Online           30 GB      0 B
  Disk 1    Online            8 GB      0 B
  Disk 2    Online            8 GB      0 B
```

어레이에서 사용할 디스크를 식별하고 해당 디스크 번호를 기록해 둡니다.

6. 어레이에서 사용할 각 디스크는 기존 볼륨을 포함하지 않는 온라인 동적 디스크여야 합니다. 다음 단계에 따라 기본 디스크를 동적 디스크로 변환하고 기존 볼륨을 삭제합니다.

   a. 다음 명령을 사용하여 어레이에서 사용할 디스크를 선택합니다. 여기에서 *n*을 디스크 번호로 대체합니다.

   ```
   DISKPART> select disk n

   Disk n is now the selected disk.
   ```

   b. 선택한 디스크가 Offline으로 나열되는 경우 online disk 명령을 실행하여 온라인으로 전환합니다.

   c. 선택한 디스크에서 이전 Dyn 명령 출력의 list disk 열에 별표가 없는 경우 디스크를 동적 디스크로 전환해야 합니다.

   ```
   DISKPART> convert dynamic
   ```

   > **Note**
   >
   > 디스크가 쓰기 금지되었다는 오류가 표시되는 경우 ATTRIBUTE DISK CLEAR READONLY 명령을 사용하여 읽기 전용 플래그를 지운 다음 동적 디스크 전환을 다시 시도할 수 있습니다.

   d. detail disk 명령을 사용하여 선택한 디스크의 기존 볼륨을 확인합니다.

   ```
   DISKPART> detail disk
   ```

```
XENSRC PVDISK SCSI Disk Device
Disk ID: 2D8BF659
Type   : SCSI
Status : Online
Path   : 0
Target : 1
LUN ID : 0
Location Path : PCIROOT(0)#PCI(0300)#SCSI(P00T01L00)
Current Read-only State : No
Read-only  : No
Boot Disk  : No
Pagefile Disk  : No
Hibernation File Disk  : No
Crashdump Disk  : No
Clustered Disk  : No

  Volume ###  Ltr  Label        Fs     Type        Size     Status     Info
  ----------  ---  -----------  -----  ----------  -------  ---------  --------
  Volume 2     D   NEW VOLUME   FAT32  Simple      8189 MB  Healthy
```

디스크의 볼륨 번호를 기록해 둡니다. 이 예제에서 볼륨 번호는 2입니다. 볼륨이 없는 경우 다음 단계를 건너뛸 수 있습니다.

e. (이전 단계에서 볼륨이 식별된 경우에만 필요함) 이전 단계에서 식별된 디스크의 기존 볼륨을 선택하여 삭제합니다.

> **Warning**
>
> 그러면 볼륨의 기존 데이터가 소멸됩니다.

i. 볼륨을 선택합니다. 여기에서 *n*을 볼륨 번호로 대체합니다.

```
DISKPART> select volume n
Volume n is the selected volume.
```

ii. 볼륨을 삭제합니다.

```
DISKPART> delete volume
```

```
DiskPart successfully deleted the volume.
```

iii. 선택한 디스크에서 삭제할 각 볼륨에 대해 이 하위 단계를 반복합니다.

f. 어레이에서 사용할 각 디스크에 대해 Step 6을 반복합니다.

7. 사용할 디스크가 새 동적 디스크인지 확인합니다. 이 경우이 예제에서는 RAID 볼륨에 디스크 1과 2를 사용합니다.

```
DISKPART> list disk

  Disk ###  Status         Size     Free     Dyn  Gpt
  --------  -------------  -------  -------  ---  ---
  Disk 0    Online           30 GB      0 B
  Disk 1    Online            8 GB      0 B   *
  Disk 2    Online            8 GB      0 B   *
```

8. RAID 어레이를 생성합니다. Windows에서는 RAID 0 볼륨을 스트라이프 볼륨이라고 합니다.

   디스크 1과 2에서 스트라이프 볼륨 어레이를 생성하려면 다음 명령을 사용합니다(어레이를 스트라이프하려면 `stripe` 옵션 사용).

```
DISKPART> create volume stripe disk=1,2
DiskPart successfully created the volume.
```

9. 새 볼륨을 확인합니다.

```
DISKPART> list volume

  DISKPART> list volume

  Volume ###  Ltr  Label        Fs     Type        Size     Status     Info
  ----------  ---  -----------  -----  ----------  -------  ---------  --------
  Volume 0     C                NTFS   Partition     29 GB  Healthy    System
  Volume 1                      RAW    Stripe        15 GB  Healthy
```

   이제 Type 열에 볼륨 1이 `stripe` 볼륨으로 표시됩니다.

10. 볼륨을 사용할 수 있도록 볼륨을 선택하여 포맷합니다.

    a. 포맷할 볼륨을 선택합니다. 여기에서 *n*을 볼륨 번호로 대체합니다.

```
DISKPART> select volume n
```

```
Volume n is the selected volume.
```

b. 볼륨을 포맷합니다.

> **Note**
>
> 전체 포맷을 수행하려면 quick 옵션을 생략합니다.

```
DISKPART> format quick recommended label="My new volume"

  100 percent completed

DiskPart successfully formatted the volume.
```

c. 사용 가능한 드라이브 문자를 볼륨에 배정합니다.

```
DISKPART> assign letter f

DiskPart successfully assigned the drive letter or mount point.
```

이제 새 볼륨을 사용할 준비가 되었습니다.

RAID 어레이에 볼륨 스냅샷 생성

스냅샷을 사용하여 RAID 배열의 EBS 볼륨에 데이터를 백업하려면 스냅샷이 일치하는지 확인해야 합니다. 이러한 볼륨의 스냅샷은 독립적으로 생성되기 때문입니다. 동기화되지 않은 스냅샷을 사용하여 RAID 배열의 EBS 볼륨을 복원할 경우 배열의 무결성이 손상됩니다.

RAID 배열에 일관된 스냅샷 집합을 생성하려면 EBS 다중 볼륨 스냅샷을 사용합니다. 다중 볼륨 스냅샷을 통해 EC2 인스턴스에 연결된 여러 EBS 볼륨에서 특정 시점, 데이터 조정 및 충돌 일치 스냅샷을 생성할 수 있습니다. 스냅샷은 여러 EBS 볼륨에서 자동으로 생성되므로 일관성을 보장하기 위해 볼륨 간에 조정하기 위해 인스턴스를 중지할 필요가 없습니다. 자세한 내용은 Amazon EBS 스냅샷 생성 아래에서 다중 볼륨 스냅샷 생성 단계를 참조하세요.

Amazon EBS 볼륨 벤치마킹

I/O 워크로드를 시뮬레이션하여 Amazon EBS 볼륨의 성능을 테스트할 수 있습니다. 프로세스는 다음과 같습니다.

1. EBS에 최적화된 인스턴스 시작.
2. 새 EBS 볼륨을 생성합니다.
3. EBS에 최적화된 인스턴스에 볼륨 추가.
4. 블록 디바이스를 구성하고 마운트합니다.
5. I/O 성능 벤치마크를 위한 도구 설치.
6. 볼륨의 I/O 성능 벤치마크.
7. 요금이 계속 발생하지 않도록 볼륨 삭제 및 인스턴스 종료.

> **Important**
>
> 일부 절차를 수행할 경우 자신이 벤치마크하는 EBS 볼륨에 있는 기존 데이터가 소멸되는 결과를 낳게 됩니다. 벤치마킹 절차는 프로덕션 볼륨이 아니라 테스트 목적으로 특별히 생성된 볼륨에 적용하기 위한 것입니다.

인스턴스 설정

EBS 볼륨에서 최적의 성능을 얻으려면 EBS에 최적화된 인스턴스를 사용하는 것이 좋습니다. EBS에 최적화된 인스턴스는 Amazon EC2 ~ Amazon EBS 간에 전용 대역폭을 전송하며, 인스턴스 유형에 따라 지정할 수 있습니다.

EBS 최적화 인스턴스를 생성하려면 Amazon EC2 콘솔을 사용하여 인스턴스를 시작할 때 EBS 최적화 인스턴스로 시작을 선택하거나 명령줄을 사용할 때 --ebs-optimized를 지정합니다. 이 옵션이 지원되는 인스턴스 유형을 선택해야 합니다.

Provisioned IOPS SSD 또는 범용 SSD 볼륨 설정

Amazon EC2 콘솔을 사용하여 프로비저닝된 IOPS SSD(`io1` 및 `io2`) 또는 범용 SSD(gp2 및 gp3) 볼륨을 생성하려면 [볼륨 유형(Volume type)]에서 [프로비저닝된 IOPS SSD(io1)(Provisioned IOPS SSD (io1))], [프로비저닝된 IOPS SSD(io2)(Provisioned IOPS SSD (io2))], [범용 SSD(gp2)(General Purpose SSD (gp2))] 또는 [범용 SSD(gp3)(General Purpose SSD (gp3))]를 선택합니다. 명령줄에서

io1 파라미터에 대해 io2, gp2, gp3 또는 --volume-type을 지정합니다. io1, io2 및 gp3 볼륨의 경우 --iops 파라미터에 대한 IOPS(초당 I/O 작업) 수를 지정합니다. 자세한 내용은 Amazon EBS 볼륨 유형 및 Amazon EBS 볼륨 생성 섹션을 참조하세요.

(Linux 인스턴스에만 해당) 예시 테스트의 경우 6개의 볼륨이 있는 RAID 0 어레이를 생성하는 것이 좋습니다. 이 어레이는 높은 수준의 성능을 제공합니다. 볼륨 수가 아닌 프로비저닝된 기가바이트와 io1, io2 및 gp3 볼륨에 대해 프로비저닝된 IOPS 수를 기준으로 요금이 부과되므로, 여러 개의 작은 볼륨을 생성하고 볼륨을 사용하여 스트라이프 세트를 생성하는 데 드는 추가 비용은 없습니다. Oracle Orion을 사용하여 볼륨을 벤치마크하는 경우 Oracle ASM과 동일한 방법으로 스트라이프를 시뮬레이트할 수 있으므로 Orion을 사용하여 스트라이프를 수행하는 것이 좋습니다. 다른 벤치마크 도구를 사용하는 경우 볼륨을 직접 스트라이프해야 합니다.

RAID 0 어레이 생성 방법에 대한 자세한 내용은 RAID 0 어레이 생성 섹션을 참조하세요.

처리량 최적화 HDD(**st1**) 또는 콜드 HDD(**sc1**) 볼륨 설정

st1 볼륨을 생성하려면 Amazon EC2 콘솔을 사용하여 볼륨을 생성할 때 처리량 최적화 HDD를 선택하거나 명령줄을 사용할 때 --type **st1**을 지정합니다. sc1 볼륨을 생성하려면 Amazon EC2 콘솔을 사용하여 볼륨을 생성할 때 콜드 HDD를 선택하거나 명령줄을 사용할 때 --type **sc1**을 지정합니다. EBS 볼륨 생성에 대한 자세한 내용은 Amazon EBS 볼륨 생성 섹션을 참조하세요. 인스턴스에 이러한 볼륨 연결에 대한 자세한 내용은 Amazon EC2 인스턴스에 Amazon EBS 볼륨 연결 섹션을 참조하세요.

(Linux 인스턴스만 해당)는이 설정 절차를 간소화 CloudFormation 하는와 함께 사용할 JSON 템플릿을 AWS 제공합니다. 템플릿에 액세스하여 JSON 파일로 저장합니다.를 CloudFormation 사용하면 자체 SSH 키를 구성할 수 있으며 st1 볼륨을 평가하기 위한 성능 테스트 환경을 더 쉽게 설정할 수 있습니다. 템플릿은 현재 세대 인스턴스와 2TiB st1 볼륨을 생성하고, /dev/xvdf에서 볼륨을 인스턴스에 연결합니다.

(Linux 인스턴스에만 해당) 템플릿을 사용하여 HDD 볼륨을 생성하는 방법

1. https://console.aws.amazon.com/cloudformation CloudFormation 콘솔을 엽니다.
2. 스택 생성을 선택합니다.
3. Amazon S3에 템플릿 업로드를 선택하고 이전에 얻은 JSON 템플릿을 선택합니다.
4. 스택에 "ebs-perf-testing" 같은 이름을 붙이고 인스턴스 유형(기본은 r3.8xlarge)과 SSH 키를 선택합니다.
5. 다음을 두 번 선택한 다음, 스택 생성을 선택합니다.
6. 새로운 스택의 상태가 CREATE_IN_PROGRESS에서 COMPLETE로 전환된 후에 출력을 선택하여 2TiB st1 볼륨이 연결될 새 인스턴스의 퍼블릭 DNS 항목을 가져옵니다.

7. 이전 단계의 DNS 항목에서 얻은 호스트 이름을 통해 SSH를 사용하여 **ec2-user**라는 사용자로 새로운 스택에 연결합니다.
8. 벤치마크 도구 설치 항목으로 이동합니다.

벤치마크 도구 설치

EBS 볼륨 성능 벤치마크에 사용할 수 있는 도구 일부가 다음 표에 나열되어 있습니다.

Linux 인스턴스

도구	설명

fio	I/O 성능을 벤치마크합니다. (fio는 `libaio-devel` 에 대해 종속성이 있습니다.)
	
	fio을 Amazon Linux에 설치하려면 다음 명령을 실행하십시오.
	
	`$ sudo yum install -y fio`
	
	Ubuntu에 fio를 설치하려면 다음 명령을 실행합니다.
	
	`sudo apt-get install -y fio`

Oracle Orion 보정 도구	Oracle 데이터베이스와 함께 사용할 스토리지 시스템의 I/O 성능을 보정합니다.

Windows 인스턴스

도구	설명
DiskSpd	DiskSpd는 Microsoft에서 Windows, Windows Server 및 Cloud Server Infrastructure 엔지니어링 팀의 스토리지 성능 도구입니다. https://github.com/Microsoft/diskspd/releases에서 다운로드할 수 있습니다. `diskspd.exe` 실행 파일을 다운로드한 후 관리 권한으로 명령 프롬프트를 연 다음(“관리자 권한으로 실행” 선택) `diskspd.exe` 파일을 복사한 디렉터리로 이동합니다.

도구	설명
	원하는 `diskspd.exe` 실행 파일을 적절한 실행 폴더(`amd64fre`, `armfre` 또는 `x86fre`))에서 `C:\DiskSpd` 같이 짧고 간단한 경로로 복사합니다. 대부분의 경우 `amd64fre` 폴더에서 64비트 버전의 DiskSpd를 사용해야 합니다. DiskSpd의 소스 코드는 GitHub에서 호스팅됩니다(https://github.com/Microsoft/diskspd).
CrystalDiskMark	CrystalDiskMark는 간단한 디스크 벤치마크 소프트웨어입니다. https://crystalmark.info/en/software/crystaldiskmark/에서 이 소프트웨어를 다운로드할 수 있습니다.

이러한 벤치마크 도구는 다양한 테스트 파라미터를 지원합니다. 볼륨이 지원하는 작업에 근접하는 명령을 사용해야 합니다. 아래 제공된 명령은 사용자가 시작하는 데 도움이 되는 예시용입니다.

볼륨 대기열 길이 선택

워크로드와 볼륨 유형에 따라 최적의 볼륨 대기열 길이를 선택합니다

SSD 지원 볼륨에서 대기열 길이

SSD 기반 볼륨의 워크로드에 대한 최적의 대기열 길이를 확인하려면 사용 가능한 모든 1000 IOPS에 대해 대기열 길이를 1로 지정하는 것이 좋습니다(범용 SSD 볼륨의 경우 기준 및 Provisioned IOPS SSD 볼륨의 경우 프로비저닝된 양). 그러면 애플리케이션 성능을 모니터링하고 애플리케이션 요구 사항을 기준으로 해당 값을 조정할 수 있습니다.

대기열 길이를 길게 하면 프로비저닝된 IOPS, 처리량 또는 최적 시스템 대기열 길이 값(현재 32로 설정)을 얻을 때까지 유용합니다. 예를 들어 프로비저닝된 IOPS가 3,000인 볼륨은 대기열 길이 3을 목표로 해야 합니다. 이 값을 높이거나 낮추면서 튜닝을 시도하여 애플리케이션에 가장 적합한 설정을 찾아야 합니다.

HDD 지원 볼륨에서 대기열 길이

HDD 지원 볼륨에서 워크로드에 가장 적합한 대기열 길이를 알아내려면 1MiB 순차 I/O를 수행하는 동시에 최소 4 이상의 대기열 길이를 목표로 하는 것이 좋습니다. 그러면 애플리케이션 성능을 모니터링하고 애플리케이션 요구 사항을 기준으로 해당 값을 조정할 수 있습니다. 예를 들어 버스트 처리량은 500MiB/s, IOPS는 500인 2TiB `st1` 볼륨의 경우 1,024KiB, 512KiB 또는 256KiB 순차 I/O를 수행하는

동시에 각각 4, 8 또는 16 대기열 길이를 목표로 해야 합니다. 이 값을 높이거나 낮추면서 튜닝을 시도하여 애플리케이션에 가장 적합한 설정을 찾아야 합니다.

C 상태 비활성화

벤치마킹 실행 전에 프로세서 C 상태를 비활성화해야 합니다. 일시적으로 지원 CPU 내 유휴 코어가 C 상태가 되어 전력을 절감할 수 있습니다. 코어가 호출되어 처리를 재개할 때 코어가 다시 완전히 작동하기까지 특정 시간이 흐릅니다. 이 지연 시간이 프로세서 벤치마킹 루틴을 방해할 수 있습니다. C 상태 및 이를 지원하는 EC2 인스턴스 유형에 대한 자세한 내용은 EC2 인스턴스에 대한 프로세서 상태 제어를 참조하세요.

Linux 인스턴스

Amazon Linux, RHEL 및 CentOS에서 다음과 같이 C 상태를 비활성화할 수 있습니다.

1. C 상태 수를 가져옵니다.

```
$ cpupower idle-info | grep "Number of idle states:"
```

2. c1부터 cN까지 C 상태를 비활성화합니다. 이상적인 경우 코어는 c0 상태여야 합니다.

```
$ for i in `seq 1 $((N-1))`; do cpupower idle-set -d $i; done
```

Windows 인스턴스

다음과 같이 Windows에서 C 상태를 비활성화할 수 있습니다.

1. PowerShell에서 현재 활성 전력 체계를 가져옵니다.

```
$current_scheme = powercfg /getactivescheme
```

2. 전력 체계 GUID를 가져옵니다.

```
(Get-WmiObject -class Win32_PowerPlan -Namespace "root\cimv2\power" -Filter
 "ElementName='High performance'").InstanceID
```

3. 전력 설정 GUID를 가져옵니다.

```
(Get-WmiObject -class Win32_PowerSetting -Namespace "root\cimv2\power" -Filter
 "ElementName='Processor idle disable'").InstanceID
```

4. 전력 설정 하위 그룹 GUID를 가져옵니다.

```
(Get-WmiObject -class Win32_PowerSettingSubgroup -Namespace "root\cimv2\power" -
Filter "ElementName='Processor power management'").InstanceID
```

5. 인덱스의 값을 1로 설정하여 C 상태를 비활성화합니다. 값이 0인 경우 C 상태가 비활성화되었음을 나타냅니다.

```
powercfg /
setacvalueindex <power_scheme_guid> <power_setting_subgroup_guid> <power_setting_guid>
 1
```

6. 활성 체계를 설정하여 설정이 저장되었는지 확인합니다.

```
powercfg /setactive <power_scheme_guid>
```

벤치마킹 수행

다음 절차에서는 다양한 EBS 볼륨 유형에 대한 벤치마킹 명령을 설명합니다.

연결된 EBS 볼륨이 있는 EBS에 최적화된 인스턴스에서 다음 명령을 실행합니다. 스냅샷에서 EBS 볼륨을 생성한 경우, 반드시 벤치마킹 전에 초기화해야 합니다. 자세한 내용은 볼륨 생성 후 수동 초기화 단원을 참조하십시오.

> **Tip**
>
> EBS 세부 성능 통계에서 제공되는 I/O 지연 시간 히스토그램을 사용하여 벤치마킹 테스트의 I/O 성능 분포를 비교할 수 있습니다. 자세한 내용은 Amazon EBS 세부 성능 통계 단원을 참조하십시오.

볼륨 테스트를 마치면 정리 도움말은 Amazon EBS 볼륨 삭제 및 인스턴스 종료를 참조하세요.

Provisioned IOPS SSD 및 범용 SSD 볼륨 벤치마크

Linux 인스턴스

생성한 RAID 0 어레이에서 fio를 실행합니다.

다음 명령은 16KB 임의 쓰기 작업을 수행합니다.

```
$ sudo fio --directory=/mnt/p_iops_vol0 --ioengine=psync --name fio_test_file --
direct=1 --rw=randwrite --bs=16k --size=1G --numjobs=16 --time_based --runtime=180 --
group_reporting --norandommap
```

다음 명령은 16KB 임의 읽기 작업을 수행합니다.

```
$ sudo fio --directory=/mnt/p_iops_vol0 --name fio_test_file --direct=1 --rw=randread
 --bs=16k --size=1G --numjobs=16 --time_based --runtime=180 --group_reporting --
norandommap
```

결과를 해석하는 방법에 대한 자세한 내용은 Inspecting disk IO performance with fio 자습서를 참조하십시오.

Windows 인스턴스

생성한 볼륨에서 DiskSpd를 실행합니다.

다음 명령은 C: 드라이브에 있는 20GB 테스트 파일(25% 쓰기 및 75% 읽기 비율, 8K 블록 크기)을 사용하여 30초 임의 I/O 테스트를 실행합니다. 각각 4개의 미해결 I/O와 1GB의 쓰기 엔트로피 값 시드가 있는 8개의 작업자 스레드를 사용합니다. 테스트 결과는 DiskSpeedResults.txt라는 텍스트 파일에 저장됩니다. 이러한 파라미터는 SQL Server OLTP 워크로드를 시뮬레이션합니다.

```
diskspd -b8K -d30 -o4 -t8 -h -r -w25 -L -Z1G -c20G C:\iotest.dat > DiskSpeedResults.txt
```

결과를 해석하는 방법에 대한 자세한 내용은 Inspecting disk IO performance with DiskSPd 자습서를 참조하십시오.

st1 및 sc1 볼륨 벤치마크(Linux 인스턴스)

fio 또는 st1 볼륨에서 sc1를 실행합니다.

> **Note**
>
> 이러한 테스트를 실행하기 전, st1 및 sc1에서 처리량이 많은 읽기 중심 워크로드의 미리 읽기 향상(Linux 인스턴스에만 해당)에 설명된 대로 인스턴스에 버퍼 I/O를 설정합니다.

다음 명령은 연결된 st1 블록 디바이스(예: /dev/xvdf)에 대해 1MiB 순차 읽기 작업을 수행합니다.

```
$ sudo fio --filename=/dev/<device> --direct=1 --rw=read --randrepeat=0
 --ioengine=libaio --bs=1024k --iodepth=8 --time_based=1 --runtime=180 --
name=fio_direct_read_test
```

다음 명령은 연결된 st1 블록 디바이스에 대해 1MiB 순차 쓰기 작업을 수행합니다.

```
$ sudo fio --filename=/dev/<device> --direct=1 --rw=write --randrepeat=0
 --ioengine=libaio --bs=1024k --iodepth=8 --time_based=1 --runtime=180 --
name=fio_direct_write_test
```

일부 워크로드는 블록 디바이스의 다양한 부분에 순차 읽기와 순차 쓰기를 혼합하여 수행합니다. 이러한 워크로드를 벤치마크하려면 읽기와 쓰기에 별도의 fio 작업을 동시에 사용하고, 각 작업에 대해 서로 다른 블록 디바이스 위치를 목표로 하기 위해 fio offset_increment 옵션을 사용하는 것이 좋습니다.

이 워크로드 실행은 순차 쓰기나 순차 읽기 워크로드보다 다소 복잡합니다. 텍스트 편집기를 사용하여 다음을 포함한 fio 작업 파일(이 예에서는 fio_rw_mix.cfg)을 만듭니다.

```
[global]
clocksource=clock_gettime
randrepeat=0
runtime=180

[sequential-write]
bs=1M
ioengine=libaio
direct=1
iodepth=8
filename=/dev/<device>
do_verify=0
rw=write
rwmixread=0
rwmixwrite=100

[sequential-read]
bs=1M
ioengine=libaio
direct=1
iodepth=8
filename=/dev/<device>
do_verify=0
```

```
rw=read
rwmixread=100
rwmixwrite=0
offset=100g
```

그런 다음, 다음 명령을 실행합니다.

```
$ sudo fio fio_rw_mix.cfg
```

결과를 해석하는 방법에 대한 자세한 내용은 Inspecting disk I/O performance with fio 자습서를 참조하세요.

순차 읽기나 쓰기 작업을 사용하는 경우라 하더라도 직접 I/O에 대한 다수의 fio 작업은 st1 및 sc1 볼륨에 기대했던 처리량보다 낮은 수준을 나타낼 수 있습니다. 하나의 직접 I/O 작업을 사용하고 iodepth 파라미터를 사용해 동시 I/O 작업의 개수를 제어하는 것이 좋습니다.

Amazon Data Lifecycle Manager를 사용하여 백업 자동화

Amazon Data Lifecycle Manager를 사용하여 EBS 스냅샷 및 EBS-backed AMI의 생성, 보존 및 삭제를 자동화할 수 있습니다. 스냅샷 및 AMI 관리를 자동화하면 다음과 같은 이점이 있습니다.

- 정기적인 백업 일정을 실행하여 중요한 데이터를 보호합니다.
- 정기적으로 새로 고칠 수 있는 표준화된 AMI를 생성합니다.
- 감사 기관이나 내부 규정 준수 부서에서 요구하는 백업을 보관합니다.
- 오래된 백업을 삭제하여 스토리지 비용을 절감합니다.
- 격리된 리전 또는 계정에 데이터를 백업하는 재해 복구 백업 정책을 생성합니다.

Amazon EventBridge 및 AWS CloudTrail 모니터링과 결합된 Amazon Data Lifecycle Manager는 추가 비용 없이 Amazon EC2 인스턴스 및 EBS 볼륨을 위한 완벽한 백업 솔루션을 제공합니다.

> **Important**
>
> - Amazon Data Lifecycle Manager는 다른 방법으로 생성된 스냅샷 또는 AMI를 관리할 수 없습니다.
> - Amazon Data Lifecycle Manager는 인스턴스 스토어 지원 AMI의 생성, 보존 및 삭제를 자동화할 수 없습니다.

Amazon Data Lifecycle Manager는 Amazon Elastic Block Store(Amazon EBS)의 서비스 기능으로 평가됩니다. Amazon EBS를 나열하는 규정 준수 프로그램(FedRAMP, HIPAA BAA, SOC 등) 범위 내 모든AWS 서비스도 Amazon Data Lifecycle Manager에 적용됩니다.

할당량

AWS 계정에는 Amazon Data Lifecycle Manager와 관련된 다음과 같은 할당량이 있습니다.

설명	할당량
리전당 사용자 지정 수명 주기 정책	100

설명	할당량
리전별 EBS 스냅샷에 대한 기본 정책	1
리전별 EBS 지원 AMI에 대한 기본 정책	1
리소스당 태그	45

Amazon Data Lifecycle Manager 작동 방식

다음은 Amazon Data Lifecycle Manager의 핵심 요소입니다.

요소

- 정책
- 정책 일정 (사용자 지정 정책만 해당)
- 대상 리소스 태그(사용자 지정 정책만 해당)
- 스냅샷
- EBS-backed AMI
- Amazon Data Lifecycle Manager 태그

정책

Amazon Data Lifecycle Manager로 정책을 생성하여 백업 생성 및 보존 요구 사항을 정의합니다. 이러한 정책은 일반적으로 다음을 지정합니다.

- 정책 유형 - 정책에서 관리하는 백업 리소스의 유형(스냅샷 또는 EBS 지원 AMI)을 정의합니다.
- 대상 리소스 - 정책의 대상이 되는 리소스의 유형(인스턴스 또는 EBS 볼륨)을 정의합니다.
- 생성 빈도 - 정책이 실행되고 스냅샷 또는 AMI를 생성하는 빈도를 정의합니다.
- 보존 임곗값 - 정책이 생성 후 스냅샷 또는 AMI를 유지하는 기간을 정의합니다.
- 추가 작업 - 크로스 리전 복사, 보관 또는 리소스 태깅과 같이 정책에서 수행해야 하는 추가 작업을 정의합니다.

Amazon Data Lifecycle Manager는 기본 정책과 사용자 지정 정책을 제공합니다.

기본 정책

기본 정책은 최근 백업이 없는 리전의 모든 볼륨과 인스턴스를 백업합니다. 제외 파라미터를 지정하여 볼륨과 인스턴스를 선택적으로 제외할 수 있습니다.

Amazon Data Lifecycle Manager는 다음과 같은 기본 정책을 지원합니다.

- EBS 스냅샷의 기본 정책 - 볼륨을 대상으로 하고 스냅샷의 생성, 보존 및 삭제를 자동화합니다.
- EBS 지원 AMI의 기본 정책 - 인스턴스를 대상으로 하고 EBS 지원 AMI의 생성, 보존 및 등록 취소를 자동화합니다.

각 계정 및 AWS 리전의 리소스 유형당 하나의 기본 정책만 가질 수 있습니다.

사용자 지정 정책

사용자 지정 정책은 할당된 태그를 기반으로 특정 리소스를 대상으로 하며 빠른 스냅샷 복원, 스냅샷 아카이빙, 크로스 계정 복사, 사전 및 사후 스크립트와 같은 고급 기능을 지원합니다. 사용자 지정 정책에는 최대 4개의 일정이 포함될 수 있으며, 각 일정에는 고유한 생성 빈도, 보존 임곗값 및 고급 기능 구성이 있을 수 있습니다.

Amazon Data Lifecycle Manager는 다음과 같은 사용자 지정 정책을 지원합니다.

- EBS 스냅샷 정책 - 볼륨 또는 인스턴스를 대상으로 하고 EBS 스냅샷의 생성, 보존 및 삭제를 자동화합니다.
- EBS 지원 AMI 정책 - 인스턴스를 대상으로 하고 EBS 지원 AMI의 생성, 보존 및 등록 취소를 자동화합니다.
- 크로스 계정 복사 이벤트 정책 - 공유된 스냅샷의 크로스 리전 복사 작업을 자동화합니다.

자세한 내용은 Amazon Data Lifecycle Manager 기본 정책 대 사용자 지정 정책 단원을 참조하십시오.

정책 일정 (사용자 지정 정책만 해당)

정책 일정은 정책에 따라 스냅샷 또는 AMI가 생성되는 시기를 정의합니다. 정책은 최대 4개의 일정(하나의—필수 일정과 최대 3개의 선택적 일정)을 가질 수 있습니다.

단일 정책에 여러 일정을 추가하면 동일한 정책을 사용하여 서로 다른 빈도로 스냅샷 또는 AMI를 생성할 수 있습니다. 예를 들어, 일별, 주별, 월별 및 연도별 스냅샷을 생성하는 단일 정책을 생성할 수 있습니다. 이렇게 하면 여러 정책을 관리할 필요가 없습니다.

각 일정에 대해 빈도, 빠른 스냅샷 복원 설정(스냅샷 수명 주기 정책만 해당), 교차 리전 복사 규칙 및 태그를 정의할 수 있습니다. 일정에 할당된 태그는 일정이 시작될 때 생성된 스냅샷 또는 AMI에 자동으로 할당됩니다. 또한 Amazon Data Lifecycle Manager는 일정의 빈도에 따라 각 스냅샷 또는 AMI에 시스템 생성 태그를 자동으로 할당합니다.

각 일정은 빈도에 따라 개별적으로 시작됩니다. 여러 일정이 동시에 시작되는 경우 Amazon Data Lifecycle Manager는 하나의 스냅샷 또는 AMI만 생성하고 보존 기간이 가장 높은 일정의 스냅샷 보존 설정을 적용합니다. 시작된 모든 일정의 태그가 스냅샷 또는 AMI에 적용됩니다.

- (스냅샷 수명 주기 정책에만 해당) 빠른 스냅샷 복원에 대해 시작된 일정 중 두 개 이상이 활성화된 경우, 시작된 모든 일정에 지정된 모든 가용 영역에서 빠른 스냅샷 복원에 대해 스냅샷이 활성화됩니다. 시작된 일정의 가장 높은 보존 설정이 각 가용 영역에 사용됩니다.
- 교차 리전 복사에 대해 시작된 일정이 중 두 개 이상이 활성화된 경우, 시작된 모든 일정에 지정된 모든 리전에 스냅샷 또는 AMI가 복사됩니다. 시작된 일정의 가장 높은 보존 기간이 적용됩니다.

대상 리소스 태그(사용자 지정 정책만 해당)

Amazon Data Lifecycle Manager 사용자 지정 정책은 리소스 태그를 사용하여 백업할 리소스를 식별합니다. 스냅샷 또는 EBS 지원 AMI 정책을 생성할 때 여러 대상 리소스 태그를 지정할 수 있습니다. 지정된 대상 리소스 태그 중 하나 이상이 있는 지정된 유형(인스턴스 또는 볼륨)의 모든 리소스가 정책의 대상이 됩니다. 예를 들어 볼륨을 대상으로 하는 스냅샷 정책을 생성하고`purpose=prod`, `costcenter=prod` 및 `environment=live`를 대상 리소스 태그로 지정하면 해당 태그-키 값 페어가 있는 모든 볼륨이 정책의 대상이 됩니다.

리소스에서 여러 정책을 실행하려는 경우 대상 리소스에 여러 태그를 할당한 다음에 특정 리소스 태그가 각각 대상이 되는 별도의 정책을 생성할 수 있습니다.

\ 또는= 문자는 태그 키에 사용할 수 없습니다. 대상 리소스 태그는 대소문자를 구분합니다. 자세한 내용은 리소스에 태그 지정을 참조하세요.

스냅샷

스냅샷은 EBS 볼륨에서 데이터를 백업하는 기본 방법입니다. 스토리지 비용을 절약하기 위해 이전 스냅샷 이후로 변경된 볼륨 데이터만 연속 스냅샷에 증분식으로 포함시킵니다. 특정 볼륨의 스냅샷 시리즈에서 스냅샷 하나를 삭제하면 해당 스냅샷에 고유한 데이터만 제거됩니다. 캡처된 볼륨 기록의 나머지는 보존됩니다. 자세한 내용은 Amazon EBS 스냅샷 단원을 참조하십시오.

EBS-backed AMI

Amazon Machine Image(AMI)는 인스턴스를 시작하는 데 필요한 정보를 제공합니다. 동일한 구성의 인스턴스가 여러 개 필요할 때는 한 AMI에서 여러 인스턴스를 시작할 수 있습니다. Amazon Data Lifecycle Manager는 EBS 지원 AMI만 지원합니다. EBS-backed AMI에는 소스 인스턴스에 연결된 각 EBS 볼륨에 대한 스냅샷이 포함됩니다. 자세한 내용은 Amazon Machine Image(AMI)를 참조하세요.

Amazon Data Lifecycle Manager 태그

Amazon Data Lifecycle Manager는 정책에 따라 생성된 모든 스냅샷 및 AMI에 다음 시스템 태그를 적용하여 다른 방법으로 생성된 스냅샷 및 AMI와 구분합니다.

- `aws:dlm:lifecycle-policy-id`
- `aws:dlm:lifecycle-schedule-name`
- `aws:dlm:expirationTime` - 기간 기반 일정으로 생성된 스냅샷용입니다. 표준 계층에서 스냅샷을 삭제할 시기를 나타냅니다.
- `dlm:managed`
- `aws:dlm:archived` - 일정에 따라 아카이브된 스냅샷용입니다.
- `aws:dlm:pre-script` - 사전 스크립트로 생성된 스냅샷의 경우
- `aws:dlm:post-script` - 사후 스크립트로 생성된 스냅샷의 경우

스냅샷 및 AMI를 생성할 때 사용자 지정 태그가 적용되도록 지정할 수도 있습니다. \ 또는= 문자는 태그 키에 사용할 수 없습니다.

필요한 경우 Amazon Data Lifecycle Manager에서 볼륨을 스냅샷 정책에 연결할 때 사용되는 대상 태그를 정책에 의해 생성된 스냅샷에 적용할 수 있습니다. 마찬가지로 인스턴스를 AMI 정책에 연결하는 데 사용되는 대상 태그를 정책에 의해 생성된 AMI에 선택적으로 적용할 수 있습니다.

Amazon Data Lifecycle Manager 기본 정책 대 사용자 지정 정책

이 섹션에서는 기본 정책과 사용자 지정 정책을 비교하고 유사점과 차이점을 강조합니다.

주제

- EBS 스냅샷 정책 비교
- EBS 지원 AMI 정책 비교

EBS 스냅샷 정책 비교

다음 표는 EBS 스냅샷의 기본 정책과 사용자 지정 EBS 스냅샷 정책의 차이를 설명합니다.

기능	EBS 스냅샷의 기본 정책	사용자 지정 EBS 스냅샷 정책
관리형 백업 리소스	EBS 스냅샷	EBS 스냅샷
대상 리소스 유형	볼륨	볼륨 또는 인스턴스
리소스 대상	최근 스냅샷이 없는 리전 내 모든 볼륨을 대상으로 합니다. 제외 파라미터를 지정하여 특정 볼륨을 제외할 수 있습니다.	특정 태그가 있는 볼륨 또는 인스턴스만 대상으로 합니다.
제외 파라미터	예, 부팅 볼륨, 특정 볼륨 유형 및 특정 태그가 있는 볼륨을 제외할 수 있습니다.	예, 인스턴스를 대상으로 할 때 부팅 볼륨과 특정 태그가 있는 볼륨을 제외할 수 있습니다.
지원 AWS Outposts	아니요	예
여러 일정 지원	아니요	예, 정책당 최대 4개의 일정
지원되는 보존 유형	경과 시간 기준 보존만	경과 시간 기준 및 개수 기준 보존
스냅샷 생성 빈도	1~7일마다.	cron 표현식을 사용한 일별, 주별, 월별, 연별 또는 사용자 지정 빈도
스냅샷 보존	2~14일.	최대 1,000개의 스냅샷(개수 기준) 또는 최대 100년(경과 시간 기준).
애플리케이션에 일관되게 적용되는 스냅샷 지원	아니요	예, 사전 및 사후 스크립트 사용

기능	EBS 스냅샷의 기본 정책	사용자 지정 EBS 스냅샷 정책
스냅샷 아카이빙 지원	아니요	예
빠른 스냅샷 복원 지원	아니요	예
크로스 리전 복사 지원	예, 기본 설정 사용 시 [1]	예, 사용자 지정 설정 사용 시
크로스 계정 공유 지원	아니요	예
확장 삭제 지원 [2]	예	아니요

[1] 기본 정책의 경우

- 크로스 리전 사본에는 태그를 복사할 수 없습니다.
- 사본은 소스 스냅샷과 동일한 보존 기간을 사용합니다.
- 사본은 소스 스냅샷과 동일한 암호화 상태를 갖습니다. 대상 리전이 기본적으로 암호화에 대해 활성화된 경우, 소스 스냅샷이 암호화되지 않았더라도 사본은 항상 암호화됩니다. 사본은 항상 대상 리전에 대한 기본 KMS 키로 암호화됩니다.

[2] 기본 및 사용자 지정 정책의 경우

- 대상 인스턴스 또는 볼륨이 삭제되는 경우 Amazon Data Lifecycle Manager는 보존 기간을 기준으로 마지막 스냅샷까지 계속 삭제합니다. 단, 마지막 스냅샷은 삭제하지 않습니다. 기본 정책의 경우 마지막 스냅샷을 포함하도록 삭제를 확장할 수 있습니다.
- 정책이 삭제되거나 오류 또는 비활성 상태가 되면 Amazon Data Lifecycle Manager는 스냅샷 삭제를 중지합니다. 기본 정책의 경우 마지막 스냅샷을 포함하여 스냅샷을 계속 삭제하도록 삭제를 확장할 수 있습니다.

EBS 지원 AMI 정책 비교

다음 표는 EBS 지원 AMI의 기본 정책과 사용자 지정 EBS 지원 AMI 정책의 차이를 설명합니다.

기능	EBS 지원 AMI에 대한 기본 정책	사용자 지정 EBS 지원 AMI 정책
관리형 백업 리소스	EBS-backed AMI	EBS-backed AMI
대상 리소스 유형	인스턴스	인스턴스
리소스 대상	최근 AMI가 없는 리전의 모든 인스턴스를 대상으로 합니다. 제외 파라미터를 지정하여 특정 인스턴스를 제외할 수 있습니다.	특정 태그가 있는 인스턴스만 대상으로 합니다.
AMI 생성 전 인스턴스 재부팅	아니요	예
제외 파라미터	예, 특정 태그가 있는 인스턴스를 제외할 수 있습니다.	아니요
여러 일정 지원	아니요	예, 정책당 최대 4개의 일정.
AMI 생성 빈도	1~7일마다.	cron 표현식을 사용한 일별, 주별, 월별, 연별 또는 사용자 지정 빈도
지원되는 보존 유형	경과 시간 기준 보존만	경과 시간 기준 및 개수 기준 보존.
AMI 보존	2~14일.	최대 1,000개의 AMI(개수 기준) 또는 최대 100년(경과 시간 기준).
AMI 사용 중단 지원	아니요	예
크로스 리전 복사 지원	예, 기본 설정 사용 시 [1]	예, 사용자 지정 설정 사용 시
확장 삭제 지원 [2]	예	아니요

[1]기본 정책의 경우

- 크로스 리전 사본에는 태그를 복사할 수 없습니다.
- 사본은 소스 AMI와 동일한 보존 기간을 사용합니다.
- 사본은 소스 AMI와 동일한 암호화 상태를 갖습니다. 대상 리전이 기본적으로 암호화에 대해 활성화된 경우, 소스 AMI가 암호화되지 않았더라도 사본은 항상 암호화됩니다. 사본은 항상 대상 리전에 대한 기본 KMS 키로 암호화됩니다.

[2] 기본 및 사용자 지정 정책의 경우

- 대상 인스턴스가 종료되는 경우 Amazon Data Lifecycle Manager는 보존 기간을 기준으로 마지막 AMI까지 계속 등록 취소합니다. 단, 마지막 스냅샷은 등록 취소하지 않습니다. 기본 정책의 경우 마지막 AMI를 포함하도록 등록 취소를 확장할 수 있습니다.
- 정책이 삭제되거나 오류 또는 비활성 상태가 되면 Amazon Data Lifecycle Manager는 AMI 등록 취소를 중지합니다. 기본 정책의 경우 마지막 AMI를 포함하여 AMI를 계속 등록 취소하도록 등록 취소를 확장할 수 있습니다.

Amazon Data Lifecycle Manager 기본 정책 생성

인스턴스에서 주기적 EBS 지원 AMI를 생성하려면 EBS 지원 AMI에 기본 정책을 사용합니다. 연결 상태에 관계없이 모든 볼륨의 스냅샷을 생성하거나 특정 볼륨을 제외하려면 EBS 스냅샷에 기본 정책을 사용합니다.

이 섹션에서는 기본 정책을 생성하는 방법을 설명합니다.

주제

- 기본 정책 고려 사항
- Amazon EBS 스냅샷에 대한 기본 정책 생성
- EBS 지원 AMI에 대한 기본 정책 생성
- 여러 계정 및 리전에서 Data Lifecycle Manager 기본 정책 활성화

기본 정책 고려 사항

기본 정책 작업 시 다음 사항에 유의하세요.

- 기본 정책은 최근 백업(스냅샷 또는 AMI)이 있는 대상 리소스(인스턴스 또는 볼륨)를 백업하지 않습니다. 생성 빈도에 따라 백업할 리소스가 결정됩니다. 볼륨 또는 인스턴스는 마지막 스냅샷 또는 AMI

가 정책의 생성 빈도보다 오래된 경우에만 백업됩니다. 예를 들어, 생성 빈도를 3일로 지정하는 경우 EBS 스냅샷의 기본 정책은 마지막 스냅샷이 3일보다 오래된 경우에만 볼륨의 스냅샷을 생성합니다.

- 기본적으로 기본 정책은 제외 파라미터가 지정되지 않은 한 해당 리전의 모든 인스턴스 또는 볼륨을 대상으로 합니다.
- 기본 정책은 최소한의 고유한 스냅샷 세트를 생성합니다. 예를 들어, EBS 지원 AMI 정책과 EBS 스냅샷 정책을 활성화하는 경우 스냅샷 정책은 EBS 지원 AMI 정책에 의해 이미 백업된 볼륨의 스냅샷을 복제하지 않습니다.
- 기본 정책은 최소 24시간 이상 경과된 리소스만 대상으로 하기 시작합니다.
- 볼륨을 삭제하거나 기본 정책의 대상이 되는 인스턴스를 종료하면 Amazon Data Lifecycle Manager는 보존 기간에 따라 마지막 백업까지 이전에 생성된 백업(스냅샷 또는 AMI)을 계속 삭제합니다. 단, 마지막 백업은 삭제하지 않습니다. 이 백업이 필요하지 않은 경우 수동으로 삭제해야 합니다.

  Amazon Data Lifecycle Manager가 마지막 백업을 삭제하도록 하려면 삭제 확장을 활성화하면 됩니다.
- 기본 정책이 삭제되거나 오류 또는 비활성 상태가 되면 Amazon Data Lifecycle Manager는 이전에 생성된 백업(스냅샷 또는 AMI) 삭제를 중지합니다. Amazon Data Lifecycle Manager가 마지막 백업을 포함하여 백업을 계속 삭제하도록 하려면 정책을 삭제하기 전 또는 정책 상태가 비활성 또는 삭제됨으로 변경되기 전에 삭제 확장을 활성화해야 합니다.
- 기본 정책을 생성하고 활성화하면 Amazon Data Lifecycle Manager는 4시간 기간에 대상 리소스를 무작위로 할당합니다. 대상 리소스는 지정된 생성 빈도로 할당된 기간 동안 백업됩니다. 예를 들어, 정책의 생성 빈도가 3일이고 대상 리소스가 12:00~16:00 기간에 할당된 경우 해당 리소스는 3일마다 12:00~16:00에 백업됩니다.

Amazon EBS 스냅샷에 대한 기본 정책 생성

다음 절차는 EBS 스냅샷에 대한 기본 정책을 생성하는 방법을 보여줍니다.

Console

EBS 스냅샷에 대한 기본 정책 생성

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 패널에서 Lifecycle Manager를 선택한 다음 수명 주기 정책 생성을 선택합니다.
3. 정책 유형에서 기본 정책을 선택한 다음 EBS 스냅샷 정책을 선택합니다.
4. 설명(Description)에 정책에 대한 간략한 설명을 입력합니다.
5. IAM 역할에서 스냅샷을 관리할 수 있는 권한이 있는 IAM 역할을 선택합니다.

기본값을 선택하여 Amazon Data Lifecycle Manager가 제공하는 기본 IAM 역할을 사용하는 것이 좋습니다. 그러나 이전에 생성된 사용자 지정 IAM 역할을 사용할 수도 있습니다.

6. 생성 빈도에 정책을 실행하고 볼륨의 스냅샷을 생성할 빈도를 지정합니다.

   지정하는 빈도에 따라 백업할 볼륨도 결정됩니다. 정책은 지정된 빈도 내에서 다른 방법으로 백업되지 않은 볼륨만 백업합니다. 예를 들어, 생성 빈도를 3일로 지정하는 경우 정책은 최근 3일 이내에 백업되지 않은 볼륨의 스냅샷만 생성합니다.

7. 보존 기간에 정책에서 생성한 스냅샷을 정책에 유지할 기간을 지정합니다. 스냅샷이 보존 임곗값에 도달하면 자동으로 삭제됩니다. 보존 기간은 생성 빈도보다 크거나 같아야 합니다.

8. (선택 사항) 예약된 백업에서 특정 볼륨을 제외하도록 제외 파라미터를 구성합니다. 제외된 볼륨은 정책이 실행될 때 백업되지 않습니다.

   a. 부트 볼륨을 제외하려면 부트 볼륨 제외를 선택합니다. 부트 볼륨을 제외하면 정책에 의해 부트 볼륨이 아닌 데이터 볼륨만 백업됩니다. 즉, 인스턴스에 부트 볼륨으로 연결된 볼륨의 스냅샷은 생성되지 않습니다.

   b. 특정 볼륨 유형을 제외하려면 특정 볼륨 유형 제외를 선택한 다음 제외할 볼륨 유형을 선택합니다. 나머지 유형의 볼륨만 정책에 의해 백업됩니다.

   c. 특정 태그가 있는 볼륨을 제외하려면 태그 추가를 선택한 다음 태그 키와 값을 지정합니다. 이 정책은 지정된 태그가 있는 볼륨의 스냅샷을 생성하지 않습니다.

9. (선택 사항) 고급 설정에서 정책이 수행해야 하는 추가 작업을 지정합니다.

   a. 할당된 태그를 소스 볼륨에서 스냅샷으로 복사하려면 볼륨에서 태그 복사를 선택합니다.

   b. 삭제 확장이 비활성화된 상태에서

      - 소스 볼륨이 삭제되는 경우 Amazon Data Lifecycle Manager는 보존 기간을 기준으로 이전에 생성된 스냅샷을 마지막 스냅샷까지 계속 삭제합니다. 단, 마지막 스냅샷은 삭제하지 않습니다. Amazon Data Lifecycle Manager가 마지막 스냅샷을 포함하여 모든 스냅샷을 삭제하도록 하려면 삭제 확장을 선택합니다.
      - 정책이 삭제되거나 `error` 또는 `disabled` 상태가 되면 Amazon Data Lifecycle Manager는 스냅샷 삭제를 중지합니다. Amazon Data Lifecycle Manager가 마지막 스냅샷을 포함하여 스냅샷을 계속 삭제하도록 하려면 삭제 확장을 선택합니다.

> **Note**
>
> 삭제 확장을 활성화하면 위에서 설명한 두 동작이 모두 동시에 재정의됩니다.

c. 정책에 의해 생성된 스냅샷을 다른 리전에 복사하려면 크로스 리전 사본 생성을 선택한 다음 최대 3개의 대상 리전을 선택합니다.

- 소스 스냅샷이 암호화되거나 대상 리전에 암호화가 기본적으로 활성화되는 경우 복사된 스냅샷이 대상 리전의 EBS 암호화를 위한 기본 KMS 키를 사용하여 암호화됩니다.
- 소스 스냅샷이 암호화되지 않고 대상 리전에 암호화가 기본적으로 비활성화되는 경우 복사된 스냅샷이 암호화되지 않습니다.

10. (선택 사항) 정책에 태그를 추가하려면 태그 추가를 선택하고 태그 키와 값 페어를 지정합니다.
11. 기본 정책 생성을 선택합니다.

> **Note**
>
> `Role with name AWSDataLifecycleManagerDefaultRole already exists` 오류가 발생하는 경우 자세한 내용은 Amazon Data Lifecycle Manager 문제 해결 섹션을 참조하세요.

AWS CLI

EBS 스냅샷에 대한 기본 정책 생성

create-lifecycle-policy 명령을 사용합니다. 사용 사례 또는 기본 설정에 따라 두 가지 방법 중 하나로 요청 파라미터를 지정할 수 있습니다.

- 방법 1

```
$ aws dlm create-lifecycle-policy \
--state ENABLED | DISABLED \
--description "policy_description" \
--execution-role-arn role_arn \
--default-policy VOLUME \
--create-interval creation_frequency_in_days (1-7) \
--retain-interval retention_period_in_days (2-14) \
--copy-tags | --no-copy-tags \
--extend-deletion | --no-extend-deletion \
--cross-region-copy-targets TargetRegion=destination_region_code \
--exclusions ExcludeBootVolumes=true | false,
 ExcludeTags=[{Key=tag_key,Value=tag_value}], ExcludeVolumeTypes="standard | gp2 |
 gp3 | io1 | io2 | st1 | sc1"
```

예를 들어, 리전의 모든 볼륨을 대상으로 하고, 기본 IAM 역할을 사용하고, 매일 실행되고(기본값), 스냅샷을 7일간 유지(기본값)하는 EBS 스냅샷에 대한 기본 정책을 생성하려면 다음 파라미터를 지정해야 합니다.

```
$ aws dlm create-lifecycle-policy \
--state ENABLED \
--description "Daily default snapshot policy" \
--execution-role-arn arn:aws:iam::account_id:role/
AWSDataLifecycleManagerDefaultRole \
--default-policy VOLUME
```

- 방법 2

```
$ aws dlm create-lifecycle-policy \
--state ENABLED | DISABLED \
--description "policy_description" \
--execution-role-arn role_arn \
--default-policy VOLUME \
--policy-details file://policyDetails.json
```

여기에서 policyDetails.json은 다음을 포함합니다.

```
{
    "PolicyLanguage": "SIMPLIFIED",
    "PolicyType": "EBS_SNAPSHOT_MANAGEMENT",
    "ResourceType": "VOLUME",
    "CopyTags": true | false,
    "CreateInterval": creation_frequency_in_days (1-7),
    "RetainInterval": retention_period_in_days (2-14),
    "ExtendDeletion": true | false,
    "CrossRegionCopyTargets": [{"TargetRegion":"destination_region_code"}],
    "Exclusions": {
        "ExcludeBootVolume": true | false,
  "ExcludeVolumeTypes": ["standard | gp2 | gp3 | io1 | io2 | st1 | sc1"],
        "ExcludeTags": [{
            "Key": "exclusion_tag_key",
            "Value": "exclusion_tag_value"
        }]
    }
}
```

EBS 지원 AMI에 대한 기본 정책 생성

다음 절차는 EBS 지원 AMI에 대한 기본 정책을 생성하는 방법을 보여줍니다.

Console

EBS 지원 AMI에 대한 기본 정책 생성

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 패널에서 Lifecycle Manager를 선택한 다음 수명 주기 정책 생성을 선택합니다.
3. 정책 유형에서 기본 정책을 선택하고 EBS 지원 AMI 정책을 선택합니다.
4. 설명(Description)에 정책에 대한 간략한 설명을 입력합니다.
5. IAM 역할에서 AMI를 관리할 수 있는 권한이 있는 IAM 역할을 선택합니다.

   기본값을 선택하여 Amazon Data Lifecycle Manager가 제공하는 기본 IAM 역할을 사용하는 것이 좋습니다. 그러나 이전에 생성된 사용자 지정 IAM 역할을 사용할 수도 있습니다.
6. 생성 빈도에 정책을 실행하고 인스턴스에서 AMI를 생성할 빈도를 지정합니다.

   지정하는 빈도에 따라 백업할 인스턴스도 결정됩니다. 정책은 지정된 빈도 내에서 다른 방법으로 백업되지 않은 인스턴스만 백업합니다. 예를 들어, 생성 빈도를 3일로 지정하는 경우 정책은 최근 3일 이내에 백업되지 않은 인스턴스의 AMI만 생성합니다.
7. 보존 기간에 정책에서 생성한 AMI를 정책에 유지할 기간을 지정합니다. AMI가 보존 임곗값에 도달하면 자동으로 등록 취소되고 연결된 스냅샷이 삭제됩니다. 보존 기간은 생성 빈도보다 크거나 같아야 합니다.
8. (선택 사항) 예약된 백업에서 특정 인스턴스를 제외하도록 제외 파라미터를 구성합니다. 제외된 인스턴스는 정책이 실행될 때 백업되지 않습니다.

   - 특정 태그가 있는 인스턴스를 제외하려면 태그 추가를 선택한 다음 태그 키와 값을 지정합니다. 이 정책은 지정된 태그가 있는 인스턴스에서 AMI를 생성하지 않습니다.
9. (선택 사항) 고급 설정에서 정책이 수행해야 하는 추가 작업을 지정합니다.

   a. 소스 인스턴스에서 해당 AMI로 할당된 태그를 복사하려면 인스턴스에서 태그 복사를 선택합니다.

   b. 삭제 확장이 비활성화된 상태에서

      - 소스 인스턴스가 종료되는 경우 Amazon Data Lifecycle Manager는 보존 기간을 기준으로 이전에 생성된 AMI를 마지막 AMI까지 계속 등록 취소합니다. 단, 마지막 AMI는 등록

취소하지 않습니다. Amazon Data Lifecycle Manager가 마지막 AMI를 포함한 모든 AMI를 등록 취소하도록 하려면 삭제 확장을 선택합니다.

- 정책이 삭제되거나 `error` 또는 `disabled` 상태가 되면 Amazon Data Lifecycle Manager는 AMI 등록 취소를 중지합니다. Amazon Data Lifecycle Manager가 마지막 AMI를 포함하여 계속해서 AMI를 등록 취소하도록 하려면 삭제 확장을 선택합니다.

> **Note**
>
> 확장 삭제를 활성화하면 위에서 설명한 두 동작이 모두 동시에 재정의됩니다.

c. 정책에 의해 생성된 AMI를 다른 리전에 복사하려면 크로스 리전 사본 생성을 선택한 다음 최대 3개의 대상 리전을 선택합니다.

- 소스 AMI가 암호화되거나 대상 리전에 암호화가 기본적으로 활성화되는 경우 복사된 AMI가 대상 리전의 EBS 암호화를 위한 기본 KMS 키를 사용하여 암호화됩니다.
- 소스 AMI가 암호화되지 않고 대상 리전에 암호화가 기본적으로 비활성화되는 경우 복사된 AMI가 암호화되지 않습니다.

10. (선택 사항) 정책에 태그를 추가하려면 태그 추가를 선택하고 태그 키와 값 페어를 지정합니다.
11. 기본 정책 생성을 선택합니다.

> **Note**
>
> `Role with name AWSDataLifecycleManagerDefaultRoleForAMIManagement already exists` 오류가 발생하는 경우 자세한 내용은 Amazon Data Lifecycle Manager 문제 해결 섹션을 참조하세요.

AWS CLI

EBS 지원 AMI에 대한 기본 정책 생성

create-lifecycle-policy 명령을 사용합니다. 사용 사례 또는 기본 설정에 따라 두 가지 방법 중 하나로 요청 파라미터를 지정할 수 있습니다.

- 방법 1

```
$ aws dlm create-lifecycle-policy \
```

```
--state ENABLED | DISABLED \
--description "policy_description" \
--execution-role-arn role_arn \
--default-policy INSTANCE \
--create-interval creation_frequency_in_days (1-7) \
--retain-interval retention_period_in_days (2-14) \
--copy-tags | --no-copy-tags \
--extend-deletion | --no-extend-deletion \
--cross-region-copy-targets TargetRegion=destination_region_code \
--exclusions ExcludeTags=[{Key=tag_key,Value=tag_value}]
```

예를 들어, 리전의 모든 인스턴스를 대상으로 하고, 기본 IAM 역할을 사용하고, 매일 실행되고(기본값), AMI를 7일간 유지(기본값)하는 EBS 지원 AMI에 대한 기본 정책을 생성하려면 다음 파라미터를 지정해야 합니다.

```
$ aws dlm create-lifecycle-policy \
--state ENABLED \
--description "Daily default AMI policy" \
--execution-role-arn arn:aws:iam::account_id:role/
AWSDataLifecycleManagerDefaultRoleForAMIManagement \
--default-policy INSTANCE
```

- 방법 2

```
$ aws dlm create-lifecycle-policy \
--state ENABLED | DISABLED \
--description "policy_description" \
--execution-role-arn role_arn \
--default-policy INSTANCE \
--policy-details file://policyDetails.json
```

여기에서 `policyDetails.json`은 다음을 포함합니다.

```
{
    "PolicyLanguage": "SIMPLIFIED",
    "PolicyType": "IMAGE_MANAGEMENT",
    "ResourceType": "INSTANCE",
    "CopyTags": true | false,
    "CreateInterval": creation_frequency_in_days (1-7),
    "RetainInterval": retention_period_in_days (2-14),
    "ExtendDeletion": true | false,
 "CrossRegionCopyTargets": [{"TargetRegion":"destination_region_code"}],
```

```
    "Exclusions": {
        "ExcludeTags": [{
            "Key": "exclusion_tag_key",
            "Value": "exclusion_tag_value"
        }]
    }
}
```

여러 계정 및 리전에서 Data Lifecycle Manager 기본 정책 활성화

CloudFormation StackSets를 사용하면 단일 작업으로 여러 계정 및 AWS 리전에서 Amazon Data Lifecycle Manager 기본 정책을 활성화할 수 있습니다.

스택 세트를 사용하여 다음 방법 중 하나로 기본 정책을 활성화할 수 있습니다.

- AWS 조직 전체 - 조직의 전체 AWS 조직 또는 특정 조직 단위에서 기본 정책이 일관되게 활성화되고 구성되도록 합니다. 이는 서비스 관리형 권한을 사용하여 수행됩니다. CloudFormation StackSets는 사용자를 대신하여 필요한 IAM 역할을 생성합니다.
- 특정 AWS 계정 간 - 특정 대상 계정에서 기본 정책이 일관되게 활성화되고 구성되어 있는지 확인합니다. 여기에는 자체 관리형 권한이 필요합니다. 스택 세트 관리자 계정과 대상 계정 간의 신뢰 관계를 설정하는 데 필요한 IAM 역할을 생성합니다.

자세한 내용은 AWS CloudFormation 사용 설명서의 스택 세트에 대한 권한 모델을 참조하세요.

다음 절차를 사용하여 전체 AWS 조직, 특정 OUs 또는 특정 대상 계정에서 Amazon Data Lifecycle Manager 기본 정책을 활성화합니다.

사전 조건

기본 정책을 활성화하는 방법에 따라 다음 중 하나를 수행합니다.

- (AWS 조직 전체) 조직의 모든 기능을 활성화하고 를 사용하여 신뢰할 수 있는 액세스를 활성화 AWS Organizations해야 합니다. 또한 조직의 관리 계정 또는 위임된 관리자 계정을 사용해야 합니다.
- (특정 대상 계정) 스택 세트 관리자 계정과 대상 계정 간에 신뢰할 수 있는 관계를 설정하는 데 필요한 역할을 생성하여 자체 관리형 권한을 부여해야 합니다.

Console

AWS 조직 또는 특정 대상 계정에서 기본 정책을 활성화하려면

1. https://console.aws.amazon.com/cloudformation CloudFormation 콘솔을 엽니다.
2. 탐색 창에서 StackSets를 선택한 다음 StackSet 생성을 선택합니다.
3. 권한에서 기본 정책을 활성화하는 방법에 따라 다음 중 하나를 수행합니다.
   - (AWS 조직 전체) 서비스 관리형 권한을 선택합니다.
   - (특정 대상 계정) 셀프 서비스 권한을 선택합니다. 그런 다음 IAM 관리자 역할 ARN에서 관리자 계정에 대해 생성한 IAM 서비스 역할을 선택하고 IAM 실행 역할 이름에서 대상 계정에서 생성한 IAM 서비스 역할의 이름을 입력합니다.
4. 템플릿 준비에서 샘플 템플릿 사용을 선택합니다.
5. 샘플 템플릿에서 다음 중 하나를 수행합니다.
   - (EBS 스냅샷에 대한 기본 정책) EBS 스냅샷에 대한 Amazon Data Lifecycle Manager 기본 정책 생성을 선택합니다.
   - (EBS 지원 AMI에 대한 기본 정책) EBS 지원 AMI에 대한 Amazon Data Lifecycle Manager 기본 정책 생성을 선택합니다.
6. 다음을 선택합니다.
7. StackSet 이름 및 StackSet 설명에 서술식 이름과 간단한 설명을 입력합니다.
8. 파라미터 섹션에서 필요에 따라 기본 정책 설정을 구성합니다.

   > **Note**
   >
   > 중요한 워크로드의 경우 CreateInterval = 1일, RetainInterval = 7일 을 사용하는 것이 좋습니다.

9. 다음을 선택합니다.
10. (선택 사항) 태그에서 StackSet 및 스택 리소스를 식별하는 데 도움이 되는 태그를 지정합니다.
11. 관리형 실행에서 활성을 선택합니다.
12. 다음을 선택합니다.
13. Add stacks to stack set(스택 세트에 스택 추가)에서 Deploy new stacks(새 스택 배포)를 선택합니다.
14. 기본 정책을 활성화하는 방법에 따라 다음 중 하나를 수행합니다.

- (AWS 조직 전체) 배포 대상에서 다음 옵션 중 하나를 선택합니다.
  - 전체 AWS 조직에 배포하려면 조직에 배포를 선택합니다.
  - 특정 조직 단위(OU)에 배포하려면 조직 단위에 배포를 선택한 다음 OU ID에 OU ID를 입력합니다. OU를 추가하려면 다른 OU 추가를 선택합니다.
- (특정 대상 계정) 계정에서 다음 중 하나를 수행합니다.
  - 특정 대상 계정에 배포하려면 계정에 스택 배포를 선택한 다음 계정 번호에 대상 계정의 IDs를 입력합니다.
  - 특정 OU의 모든 계정에 배포하려면 조직 단위의 모든 계정에 스택 배포를 선택한 다음 조직 번호에 대상 OU의 ID를 입력합니다.

15. 자동 배포에서 활성화됨을 선택합니다.
16. 계정 제거 동작에서 스택 보관을 선택합니다.
17. 리전 지정에서 기본 정책을 활성화할 특정 리전을 선택하거나 모든 리전 추가를 선택하여 모든 리전에서 기본 정책을 활성화합니다.
18. 다음을 선택합니다.
19. 스택 세트 설정을 검토하고이 IAM 리소스를 생성할 CloudFormation 수 있음을 승인합니다를 선택한 다음 제출을 선택합니다.

AWS CLI

AWS 조직 전체에서 기본 정책을 활성화하려면

1. 스택 세트를 생성합니다. create-stack-set 명령을 사용합니다.

   `--permission-model`에서 `SERVICE_MANAGED`를 지정합니다.

   `--template-url`에 다음 템플릿 URL 중 하나를 지정합니다.

   - (EBS 지원 AMI에 대한 기본 정책) `https://s3.amazonaws.com/cloudformation-stackset-sample-templates-us-east-1/DataLifecycleManagerAMIDefaultPolicy.yaml`
   - (EBS 스냅샷에 대한 기본 정책) `https://s3.amazonaws.com/cloudformation-stackset-sample-templates-us-east-1/DataLifecycleManagerEBSSnapshotDefaultPolicy.yaml`

`--parameters`에 기본 정책의 설정을 지정합니다. 지원되는 파라미터, 파라미터 설명 및 유효한 값의 경우 URL을 사용하여 템플릿을 다운로드한 다음 텍스트 편집기를 사용하여 템플릿을 봅니다.

`--auto-deployment`에서 `Enabled=true`, `RetainStacksOnAccountRemoval=true`를 지정합니다.

```
$ aws cloudformation create-stack-set \
--stack-set-name stackset_name \
--permission-model SERVICE_MANAGED \
--template-url template_url \
--parameters "ParameterKey=param_name_1,ParameterValue=param_value_1"
 "ParameterKey=param_name_2,ParameterValue=param_value_2" \
--auto-deployment "Enabled=true, RetainStacksOnAccountRemoval=true"
```

2. 스택 세트를 배포합니다. create-stack-instances 명령을 사용합니다.

   `--stack-set-name`에 이전 단계에서 생성한 스택 세트의 이름을 지정합니다.

   `--deployment-targets OrganizationalUnitIds`에서 전체 조직에 배포할 루트 OU의 ID 또는 조직의 특정 OU에 배포할 OU ID를 지정합니다.

   에서 기본 정책을 활성화할 AWS 리전을 `--regions`지정합니다.

```
$ aws cloudformation create-stack-instances \
--stack-set-name stackset_name \
--deployment-targets OrganizationalUnitIds='["root_ou_id"]' | '["ou_id_1",
 "ou_id_2]' \
--regions '["region_1", "region_2"]'
```

특정 대상 계정에서 기본 정책을 활성화하려면

1. 스택 세트를 생성합니다. create-stack-set 명령을 사용합니다.

   `--template-url`에 다음 템플릿 URL 중 하나를 지정합니다.

   - (EBS 지원 AMI에 대한 기본 정책) `https://s3.amazonaws.com/cloudformation-stackset-sample-templates-us-east-1/DataLifecycleManagerAMIDefaultPolicy.yaml`

- (EBS 스냅샷에 대한 기본 정책) `https://s3.amazonaws.com/cloudformation-stackset-sample-templates-us-east-1/DataLifecycleManagerEBSSnapshotDefaultPolicy.yaml`

`--administration-role-arn`에서 이전에 스택 세트 관리자를 위해 생성한 IAM 서비스 역할의 ARN을 지정합니다.

`--execution-role-name`에 대상 계정에서 생성한 IAM 서비스 역할의 이름을 지정합니다.

`--parameters`에 기본 정책의 설정을 지정합니다. 지원되는 파라미터, 파라미터 설명 및 유효한 값의 경우 URL을 사용하여 템플릿을 다운로드한 다음 텍스트 편집기를 사용하여 템플릿을 봅니다.

`--auto-deployment`에서 `Enabled=true, RetainStacksOnAccountRemoval=true`를 지정합니다.

```
$ aws cloudformation create-stack-set \
--stack-set-name stackset_name \
--template-url template_url \
--parameters "ParameterKey=param_name_1,ParameterValue=param_value_1"
 "ParameterKey=param_name_2,ParameterValue=param_value_2" \
--administration-role-arn administrator_role_arn \
--execution-role-name target_account_role \
--auto-deployment "Enabled=true, RetainStacksOnAccountRemoval=true"
```

2. 스택 세트를 배포합니다. create-stack-instances 명령을 사용합니다.

`--stack-set-name`에 이전 단계에서 생성한 스택 세트의 이름을 지정합니다.

에서 대상 AWS 계정IDs를 `--accounts`지정합니다.

에서 기본 정책을 활성화할 AWS 리전을 `--regions`지정합니다.

```
$ aws cloudformation create-stack-instances \
--stack-set-name stackset_name \
--accounts '["account_ID_1","account_ID_2"]' \
--regions '["region_1", "region_2"]'
```

EBS 스냅샷에 대한 Amazon Data Lifecycle Manager 사용자 지정 정책 생성

다음 절차에서는 Amazon Data Lifecycle Manager를 사용하여 Amazon EBS 스냅샷 수명 주기를 자동화하는 방법을 보여줍니다.

주제

- 스냅샷 수명 주기 정책 생성
- 스냅샷 수명 주기 정책 고려 사항
- 추가 리소스
- Data Lifecycle Manager를 사용하여 애플리케이션에 일관되게 적용되는 스냅샷 자동화
- Data Lifecycle Manager 사전 및 사후 스크립트의 기타 사용 사례
- Amazon Data Lifecycle Manager 사전 및 사후 스크립트 작동 방식
- Data Lifecycle Manager 사전 및 사후 스크립트로 생성된 스냅샷 식별
- Amazon Data Lifecycle Manager 사전 및 사후 스크립트 모니터링

스냅샷 수명 주기 정책 생성

다음 절차 중 하나를 사용하여 스냅샷 또는 수명 주기 정책을 생성합니다.

Console

스냅샷 정책을 생성하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Elastic Block Store, Lifecycle Manager 및 수명 주기 정책 생성을 차례로 선택합니다.
3. 정책 유형 선택(Select policy type) 화면에서 EBS 스냅샷 정책(EBS snapshot policy)을 선택한 후 다음(Next)을 선택합니다.
4. 대상 리소스(Target resources) 섹션에서 다음을 수행합니다.

    a. 대상 리소스 유형(Target resource types)에서 백업할 리소스의 유형을 선택합니다. `Volume`을 선택하여 개별 볼륨의 스냅샷을 생성하거나 `Instance`를 선택하여 인스턴스에 연결된 볼륨에서 다중 볼륨 스냅샷을 생성합니다.

    b. (Outpost 및 로컬 영역 고객에만 해당) 대상 리소스의 위치를 지정합니다.

대상 리소스 위치에서 대상 리소스의 위치를 지정합니다.

- 리전의 리소스를 대상으로 지정하려면 AWS 리전을 선택합니다. Amazon Data Lifecycle Manager는 현재 리전에 있고 대상 태그가 일치하는 지정된 유형의 모든 리소스만 백업합니다. 스냅샷은 동일한 리전에 생성됩니다.
- 로컬 영역의 리소스를 대상으로 지정하려면 AWS 로컬 영역을 선택합니다. Amazon Data Lifecycle Manager는 현재 리전의 모든 로컬 영역에서 대상 태그가 일치하는 지정된 유형의 모든 리소스만 백업합니다. 스냅샷은 소스 리소스와 동일한 로컬 영역 또는 상위 리전에 생성될 수 있습니다.
- Outpost의 리소스를 대상으로 지정하려면 AWS Outpost를 선택합니다. Amazon Data Lifecycle Manager는 계정의 모든 Outposts에서 대상 태그가 일치하는 지정된 유형의 모든 리소스를 백업합니다. 스냅샷은 소스 리소스와 동일한 Outpost 또는 상위 리전에 생성될 수 있습니다.

c. 대상 리소스 태그(Target resource tags)에서 백업할 볼륨 또는 인스턴스를 식별하는 리소스 태그를 선택합니다. 지정된 태그 키 및 값 쌍이 있는 리소스만 정책에 의해 백업됩니다.

5. 설명(Description)에 정책에 대한 간략한 설명을 입력합니다.
6. IAM 역할(IAM role)에서 스냅샷을 관리하고 볼륨 및 인스턴스를 설명할 권한이 있는 IAM 역할을 선택합니다. Amazon Data Lifecycle Manager가 제공하는 기본 역할을 사용하려면 [기본 역할(Default role)]을 선택합니다. 또는 이전에 생성한 사용자 지정 IAM 역할을 사용하려면 다른 역할 선택(Choose another role)을 선택하고 사용할 역할을 선택합니다.
7. 정책 태그(Policy tags)에서 수명 주기 정책에 적용할 태그를 추가합니다. 이 태그를 사용하여 정책을 식별 및 분류할 수 있습니다.
8. Policy status after creation(생성 후 정책 상태) - Enable policy(정책 활성화)를 선택하여 다음 예약 시간에 정책 실행을 시작하거나, Disable policy(정책 비활성화)를 선택하여 정책을 실행하지 않습니다. 지금 정책을 활성화하지 않으면 생성 후 수동으로 활성화할 때까지 스냅샷 생성이 시작되지 않습니다.
9. (인스턴스만 대상으로 하는 정책) 다중 볼륨 스냅샷 세트에서 볼륨을 제외합니다.

   기본적으로 Amazon Data Lifecycle Manager는 대상 인스턴스에 연결된 모든 볼륨의 스냅샷을 생성합니다. 그러나 연결된 볼륨 중 일부에 대한 스냅샷을 생성하도록 선택할 수 있습니다. Parameters(파라미터) 섹션에서 다음을 수행합니다.

   - 대상 인스턴스에 연결된 루트 볼륨의 스냅샷을 생성하지 않으려면 Exclude root volume(루트 볼륨 제외)을 선택합니다. 이 옵션을 선택하면 대상 인스턴스에 연결된 데이터(루트 아님) 볼륨만 다중 볼륨 스냅샷 세트에 포함됩니다.

- 대상 인스턴스에 연결된 데이터(루트 아님) 볼륨 중 일부에 대한 스냅샷을 생성하려면 Exclude specific data volumes(특정 데이터 볼륨 제외)를 선택한 다음 스냅샷을 생성하지 않아야 하는 데이터 볼륨을 식별하는 데 사용할 태그를 지정합니다. Amazon Data Lifecycle Manager는 지정된 태그가 있는 데이터 볼륨의 스냅샷을 생성하지 않습니다. Amazon Data Lifecycle Manager는 지정된 태그가 없는 데이터 볼륨의 스냅샷만 생성합니다.

10. [다음(Next)]을 선택합니다.
11. 일정 구성(Configure schedule) 화면에서 정책 일정을 구성합니다. 정책에는 최대 4개의 일정을 구성할 수 있습니다. 일정 1은 필수입니다. 일정 2, 3, 4는 선택 사항입니다. 추가한 각 정책 일정에 대해 다음을 수행합니다.

    a. 일정 세부 정보(Schedule details) 섹션에서 다음을 수행합니다.

        i. 일정 이름(Schedule name)에 일정을 설명하는 이름을 지정합니다.

        ii. 빈도(Frequency) 및 관련 필드에서 정책 실행 간격을 구성합니다.

        매일, 매주, 매월 또는 매년 일정에 따라 정책 실행을 구성할 수 있습니다. 또는 사용자 지정 cron 표현식(Custom cron expression)을 선택하여 최대 1년의 간격을 지정합니다. 자세한 내용은 Amazon EventBridge 사용 설명서의 Cron 및 rate 표현식을 참조하세요.

        > **Note**
        >
        > 일정에 대해 스냅샷 아카이빙을 활성화해야 하는 경우 monthly(월간) 또는 yearly(연간) 빈도를 선택하거나 생성 빈도가 28일 이상인 cron 표현식을 지정해야 합니다.
        > 특정 주의 특정 날짜(예: 매월 두 번째 목요일)에 스냅샷을 생성하는 월별 빈도를 지정하는 경우 개수 기반 일정의 경우 아카이브 계층의 보존 횟수는 4회 이상이어야 합니다.

        iii. 시작 시간(Starting at)에서 정책 실행의 시작을 예약할 시간을 지정합니다. 첫 번째 정책 실행은 예약된 시간 후 한 시간 이내에 시작됩니다. 이 시간은 hh:mm UTC 형식으로 입력해야 합니다.

        iv. 보존 유형(Retention type)에서 일정에 따라 생성되는 스냅샷의 보존 정책을 지정합니다.

        총 수 또는 수명을 기준으로 스냅샷을 보존할 수 있습니다.

        - 개수 기반 보존

- 스냅샷 아카이브를 비활성화한 경우 범위는 1~1000입니다. 보존 임계값에 도달하면 가장 오래된 스냅샷이 영구적으로 삭제됩니다.
- 스냅샷 아카이브가 활성화된 경우 범위는 0(생성 후 즉시 아카이브)~1000입니다. 보존 임계값에 도달하면 가장 오래된 스냅샷이 전체 스냅샷으로 변환되고 아카이브 계층으로 이동됩니다.

- 경과 시간 기준 보존
  - 스냅샷 아카이브를 비활성화한 경우 범위는 1일~100년입니다. 보존 임계값에 도달하면 가장 오래된 스냅샷이 영구적으로 삭제됩니다.
  - 스냅샷 아카이브가 활성화된 경우 범위는 0일(생성 후 즉시 아카이브)~100년입니다. 보존 임계값에 도달하면 가장 오래된 스냅샷이 전체 스냅샷으로 변환되고 아카이브 계층으로 이동됩니다.

> **Note**
>
> - 모든 일정은 보존 유형이 동일해야 합니다(기간 기반 또는 개수 기반). 일정 1에 대해서만 보존 유형을 지정할 수 있습니다. 일정 2, 3, 4는 일정 1에서 보존 유형을 상속합니다. 각 일정에는 고유한 보존 횟수 또는 기간이 있을 수 있습니다.
> - 빠른 스냅샷 복원, 크로스 리전 복사 또는 스냅샷 공유를 활성화하는 경우 보존 횟수를 1 이상으로 지정하거나 보존 기간을 1일 이상으로 지정해야 합니다.

v. (AWS Outposts 및 로컬 영역 고객만 해당) 스냅샷 대상을 지정합니다.

스냅샷 대상에 정책에 따라 생성되는 스냅샷의 대상을 지정합니다.

- 정책이 리전의 리소스를 대상으로 하는 경우 동일한 리전에서 스냅샷을 생성해야 합니다. AWS 리전이 자동으로 선택됩니다.
- 정책의 대상이 로컬 영역의 리소스인 경우 소스 리소스와 동일한 로컬 영역 또는 상위 리전에 스냅샷을 생성할 수 있습니다.
- 정책의 대상이 Outpost의 리소스인 경우 소스 리소스와 동일한 Outpost 또는 상위 리전에 스냅샷을 생성할 수 있습니다.

b. 스냅샷에 대한 태깅을 구성합니다.

태깅(Tagging) 섹션에서 다음을 수행합니다.

i. 일정에 따라 소스 볼륨에서 사용자 정의 태그를 모두 스냅샷으로 복사하려면 소스에서 태그 복사(Copy tags from source)를 선택합니다.

ii. 이 일정에 따라 생성된 스냅샷에 할당할 추가 태그를 지정하려면 태그 추가(Add tags)를 선택합니다.

c. 애플리케이션에 일관되게 적용되는 스냅샷을 위한 사전 및 사후 스크립트를 구성합니다.

자세한 내용은 Data Lifecycle Manager를 사용하여 애플리케이션에 일관되게 적용되는 스냅샷 자동화 단원을 참조하십시오.

d. (볼륨만 대상으로 하는 정책) 스냅샷 아카이빙을 구성합니다.

스냅샷 아카이빙 섹션에서 다음을 수행합니다.

> **Note**
>
> 정책에서 하나의 일정에 대해서만 스냅샷 아카이빙을 활성화할 수 있습니다.

i. 일정에 대해 스냅샷 아카이빙을 활성화하려면 Archive snapshots created by this schedule(이 일정에 의해 생성된 스냅샷 아카이브)을 선택합니다.

> **Note**
>
> 스냅샷 생성 빈도가 월별 또는 연별인 경우 또는 생성 빈도가 28일 이상인 cron 표현식을 지정하는 경우에만 스냅샷 아카이빙을 활성화할 수 있습니다.

ii. 아카이브 계층의 스냅샷에 대한 보존 규칙을 지정합니다.

- 개수 기반 일정의 경우 아카이브 계층에 유지할 스냅샷 수를 지정합니다. 보존 임계값에 도달하면 가장 오래된 스냅샷이 아카이브 계층에서 영구적으로 삭제됩니다. 예를 들어, 3을 지정하면 일정은 아카이브 계층에 최대 3개의 스냅샷을 유지합니다. 네 번째 스냅샷이 아카이브되면 아카이브 계층에 있는 세 개의 기존 스냅샷 중 가장 오래된 것이 삭제됩니다.
- 기간 기반 일정의 경우 아카이브 계층에 스냅샷을 유지할 기간을 지정합니다. 보존 임계값에 도달하면 가장 오래된 스냅샷이 아카이브 계층에서 영구적으로 삭제됩니다. 예를 들어, 120일을 지정하면 해당 기간 경과 시 자동으로 아카이브 계층에서 스냅샷이 삭제됩니다.

> **Important**
>
> 아카이브된 스냅샷의 최소 보존 기간은 90일입니다. 스냅샷을 90일 이상 유지하는 보존 규칙을 지정해야 합니다.

e. 빠른 스냅샷 복원을 활성화합니다.

스케줄에 따라 생성된 스냅샷에 대해 빠른 스냅샷 복원을 활성화하려면 빠른 스냅샷 복원(Fast snapshot restore) 섹션에서 빠른 스냅샷 복원 활성화(Enable fast snapshot restore)를 선택합니다. 빠른 스냅샷 복원을 활성화하는 경우 이를 활성화할 가용 영역을 선택해야 합니다. 일정에서 수명 기준 보존 일정을 사용하는 경우 각 스냅샷에 대해 빠른 스냅샷 복원을 활성화할 기간을 지정해야 합니다. 일정에서 개수 기준 보존을 사용하는 경우 빠른 스냅샷 복원을 활성화할 최대 스냅샷 수를 지정해야 합니다.

일정에서 Outpost에 스냅샷을 생성하는 경우 빠른 스냅샷 복원을 활성화할 수 없습니다. 빠른 스냅샷 복원은 Outpost에 저장된 로컬 스냅샷에서는 지원되지 않습니다.

> **Note**
>
> 특정 가용 영역의 스냅샷에 대해 빠른 스냅샷 복원이 활성화된 1분마다 요금이 청구됩니다. 요금은 최소 1시간으로 비례 청구됩니다.

f. 크로스 리전 복사를 구성합니다.

일정에 따라 생성된 스냅샷을 Outpost 또는 다른 리전에 복사하려면 리전 간 복사 섹션에서 리전 간 복사 활성화를 선택합니다.

일정에서 리전에 스냅샷을 생성하는 경우 계정에 있는 최대 3개의 추가 리전 또는 Outposts에 스냅샷을 복사할 수 있습니다. 각 대상 리전 또는 Outpost에 대해 별도의 리전 간 복사 규칙을 지정해야 합니다.

각 리전 또는 Outpost에 대해 다른 보존 정책을 선택하고 태그를 모두 복사할지 또는 복사하지 않을지 선택할 수 있습니다. 소스 스냅샷이 암호화되거나 암호화가 기본적으로 활성화된 경우 복사된 스냅샷이 암호화됩니다. 소스 스냅샷이 암호화 해제된 경우 암호화를 활성화할 수 있습니다. KMS 키를 지정하지 않은 경우 스냅샷은 각 대상 리전에서 EBS 암호화의 기본 KMS 키를 사용하여 암호화됩니다. 대상 리전에 대한 KMS 키를 지정하는 경우 선택한 IAM 역할에 KMS 키에 대한 액세스 권한이 있어야 합니다.

> **Note**
>
> 리전당 동시 스냅샷 복사본 수를 초과하지 않도록 해야 합니다.

정책에서 Outpost에 스냅샷을 생성하는 경우 해당 스냅샷을 리전이나 다른 Outpost에 복사할 수 없으며 교차 리전 복사 설정을 사용할 수 없습니다.

g. 크로스 계정 공유를 구성합니다.

   교차 계정 공유에서 일정에 의해 생성된 스냅샷을 다른 AWS 계정과 자동으로 공유하도록 정책을 구성합니다. 해결 방법:

   i. 다른 AWS 계정과의 공유를 활성화하려면 교차 계정 공유 활성화를 선택합니다.

   ii. 스냅샷을 공유할 계정을 추가하려면 계정 추가(Add account)에 12자리 AWS 계정 ID를 입력하고 추가(Add)를 선택합니다.

   iii. 일정 기간이 지난 후에 공유 스냅샷의 공유를 자동으로 해제하려면 자동으로 공유 해제(Unshare automatically)를 선택합니다. 공유 스냅샷의 공유를 자동으로 해제하도록 선택한 경우 스냅샷의 공유를 자동으로 해제하는 기간은 정책에서 해당 스냅샷을 보존하는 기간보다 길 수 없습니다. 예를 들어 정책의 보존 구성에서 5일 동안 스냅샷을 보존하는 경우 최대 4일 후에 공유 스냅샷의 공유를 자동으로 해제하도록 정책을 구성할 수 있습니다. 이는 수명 기반 및 개수 기반 스냅샷 보존 구성을 사용하는 정책에 적용됩니다.

      자동 공유 해제를 활성화하지 않으면 스냅샷이 삭제될 때까지 공유됩니다.

      > **Note**
      >
      > 암호화되지 않았거나 고객 관리형 키를 사용하여 암호화된 스냅샷만 공유할 수 있습니다. 기본 EBS 암호화 KMS 키(으)로 암호화된 스냅샷은 공유할 수 없습니다. 암호화된 스냅샷을 공유하는 경우 소스 볼륨을 암호화하는 데 사용된 KMS 키도 대상 계정과 공유해야 합니다. 자세한 내용은 AWS Key Management Service 개발자 안내서의 다른 계정의 사용자가 CMK를 사용하도록 허용을 참조하세요.

h. 일정을 더 추가하려면 화면 상단에서 [다른 일정 추가(Add another schedule)]를 선택합니다. 각 추가 일정에 대해, 이 주제의 앞부분에서 설명한 대로 필드를 작성합니다.

i. 필요한 일정을 추가한 후 [정책 검토(Review policy)]를 선택합니다.

12. 정책 요약을 검토한 다음 정책 생성(Create policy)을 선택합니다.

> **Note**
>
> `Role with name AWSDataLifecycleManagerDefaultRole already exists` 오류가 발생하는 경우 자세한 내용은 Amazon Data Lifecycle Manager 문제 해결 섹션을 참조하세요.

Command line

create-lifecycle-policy 명령을 사용하여 스냅샷 수명 주기 정책을 생성합니다. `PolicyType`에 `EBS_SNAPSHOT_MANAGEMENT`을 지정합니다.

> **Note**
>
> 구문을 단순화하기 위해 다음 예에서는 정책 세부 정보가 포함된 JSON 파일(`policyDetails.json`)을 사용합니다.

예제 1 - 2개의 일정이 포함된 스냅샷 수명 주기 정책

이 예제에서는 값이 `costcenter`인 115 태그 키가 있는 모든 볼륨의 스냅샷을 생성하는 스냅샷 수명 주기 정책을 만듭니다. 정책에는 2개의 일정이 포함되어 있습니다. 첫 번째 일정은 매일 03:00(UTC)에 스냅샷을 생성합니다. 두 번째 일정은 매주 금요일 17:00(UTC)에 주간 스냅샷을 생성합니다.

```
aws dlm create-lifecycle-policy \
    --description "My volume policy" \
    --state ENABLED \
    --execution-role-arn
 arn:aws:iam::12345678910:role/AWSDataLifecycleManagerDefaultRole \
    --policy-details file://policyDetails.json
```

다음은 `policyDetails.json` 파일의 예입니다.

```
{
    "PolicyType": "EBS_SNAPSHOT_MANAGEMENT",
    "ResourceTypes": [
```

```
        "VOLUME"
    ],
    "TargetTags": [{
        "Key": "costcenter",
        "Value": "115"
    }],
    "Schedules": [{
        "Name": "DailySnapshots",
        "TagsToAdd": [{
            "Key": "type",
            "Value": "myDailySnapshot"
        }],
        "CreateRule": {
            "Interval": 24,
            "IntervalUnit": "HOURS",
            "Times": [
                "03:00"
            ]
        },
        "RetainRule": {
            "Count": 5
        },
        "CopyTags": false
    },
    {
        "Name": "WeeklySnapshots",
        "TagsToAdd": [{
            "Key": "type",
            "Value": "myWeeklySnapshot"
        }],
        "CreateRule": {
            "CronExpression": "cron(0 17 ? * FRI *)"
        },
        "RetainRule": {
            "Count": 5
        },
        "CopyTags": false
    }
]}
```

요청이 성공하면 명령은 새로 생성된 정책의 ID를 반환합니다. 다음은 예제 출력입니다.

```
{
```

```
    "PolicyId": "policy-0123456789abcdef0"
}
```

예 2 - 인스턴스를 대상으로 하고 데이터(루트 아님) 볼륨 중 일부에 대한 스냅샷을 생성하는 스냅샷 수명 주기 정책

이 예제에서는 code=production 태그가 지정된 인스턴스에서 다중 볼륨 스냅샷 세트를 생성하는 스냅샷 수명 주기 정책을 생성합니다. 정책에는 하나의 일정만 포함됩니다. 일정은 code=temp로 태그가 지정된 데이터 볼륨의 스냅샷을 생성하지 않습니다.

```
aws dlm create-lifecycle-policy \
    --description "My volume policy" \
    --state ENABLED \
    --execution-role-arn
 arn:aws:iam::12345678910:role/AWSDataLifecycleManagerDefaultRole \
    --policy-details file://policyDetails.json
```

다음은 policyDetails.json 파일의 예입니다.

```
{
    "PolicyType": "EBS_SNAPSHOT_MANAGEMENT",
    "ResourceTypes": [
        "INSTANCE"
    ],
    "TargetTags": [{
        "Key": "code",
        "Value": "production"
    }],
    "Parameters": {
        "ExcludeDataVolumeTags": [{
            "Key": "code",
            "Value": "temp"
        }]
    },
    "Schedules": [{
        "Name": "DailySnapshots",
        "TagsToAdd": [{
            "Key": "type",
            "Value": "myDailySnapshot"
        }],
        "CreateRule": {
            "Interval": 24,
```

```
            "IntervalUnit": "HOURS",
            "Times": [
                "03:00"
            ]
        },
        "RetainRule": {
            "Count": 5
        },
        "CopyTags": false
    }
]}
```

요청이 성공하면 명령은 새로 생성된 정책의 ID를 반환합니다. 다음은 예제 출력입니다.

```
{
   "PolicyId": "policy-0123456789abcdef0"
}
```

예제 3 - Outpost 리소스의 로컬 스냅샷을 자동화하는 스냅샷 수명 주기 정책

이 예제에서는 모든 Outposts에서 `team=dev` 태그가 지정된 볼륨의 스냅샷을 생성하는 스냅샷 수명 주기 정책을 생성합니다. 이 정책은 소스 볼륨과 동일한 Outposts에 스냅샷을 생성합니다. 정책은 12 UTC에 시작하여 `00:00`시간마다 스냅샷을 생성합니다.

```
aws dlm create-lifecycle-policy \
    --description "My local snapshot policy" \
    --state ENABLED \
    --execution-role-arn
 arn:aws:iam::12345678910:role/AWSDataLifecycleManagerDefaultRole \
    --policy-details file://policyDetails.json
```

다음은 `policyDetails.json` 파일의 예입니다.

```
{
    "PolicyType": "EBS_SNAPSHOT_MANAGEMENT",
    "ResourceTypes": "VOLUME",
 "ResourceLocations": "OUTPOST",
    "TargetTags": [{
        "Key": "team",
        "Value": "dev"
    }],
    "Schedules": [{
```

```
        "Name": "on-site backup",
        "CreateRule": {
            "Interval": 12,
            "IntervalUnit": "HOURS",
            "Times": [
                "00:00"
            ],
 "Location": [
  "OUTPOST_LOCAL"
 ]
        },
        "RetainRule": {
            "Count": 1
        },
        "CopyTags": false
    }
]}
```

예제 4 - 리전에서 스냅샷을 생성하고 이를 Outpost에 복사하는 스냅샷 수명 주기 정책

다음 예제 정책은 team=dev 태그가 지정된 볼륨의 스냅샷을 생성합니다. 스냅샷은 소스 볼륨과 동일한 리전에 생성됩니다. 스냅샷은 12 UTC에 시작하여 00:00시간마다 생성되며, 최대 1개 스냅샷을 유지합니다. 또한 이 정책은 스냅샷을 Outpost arn:aws:outposts:us-east-1:123456789012:outpost/op-1234567890abcdef0에 복사하고, 기본 암호화 KMS 키를) 사용하여 복사된 스냅샷을 암호화하며, 1개월 동안 복사본을 유지합니다.

```
aws dlm create-lifecycle-policy \
    --description "Copy snapshots to Outpost" \
    --state ENABLED \
    --execution-role-arn
 arn:aws:iam::12345678910:role/AWSDataLifecycleManagerDefaultRole \
    --policy-details file://policyDetails.json
```

다음은 policyDetails.json 파일의 예입니다.

```
{
    "PolicyType": "EBS_SNAPSHOT_MANAGEMENT",
    "ResourceTypes": "VOLUME",
    "ResourceLocations": "CLOUD",
    "TargetTags": [{
        "Key": "team",
        "Value": "dev"
```

```
    }],
    "Schedules": [{
        "Name": "on-site backup",
        "CopyTags": false,
        "CreateRule": {
            "Interval": 12,
            "IntervalUnit": "HOURS",
            "Times": [
                "00:00"
            ],
            "Location": "CLOUD"
        },
        "RetainRule": {
            "Count": 1
        },
        "CrossRegionCopyRules" : [
        {
            "Target": "arn:aws:outposts:us-east-1:123456789012:outpost/
op-1234567890abcdef0",
            "Encrypted": true,
            "CopyTags": true,
            "RetainRule": {
                "Interval": 1,
                "IntervalUnit": "MONTHS"
            }
        }]
    }
]}
```

예제 5 - 아카이브가 활성화된 기간 기반 일정이 포함된 스냅샷 수명 주기 정책

이 예제에서는 Name=Prod로 태그가 지정된 볼륨을 대상으로 하는 스냅샷 수명 주기 정책을 생성합니다. 정책에는 매월 1일 09:00에 스냅샷을 생성하는 기간 기반 일정이 하나 있습니다. 일정은 표준 계층의 각 스냅샷을 하루 동안 유지한 후 아카이브 계층으로 이동합니다. 스냅샷은 삭제되기 전에 90일 동안 아카이브 계층에 저장됩니다.

```
aws dlm create-lifecycle-policy \
    --description "Copy snapshots to Outpost" \
    --state ENABLED \
    --execution-role-arn
 arn:aws:iam::12345678910:role/AWSDataLifecycleManagerDefaultRole \
    --policy-details file://policyDetails.json
```

다음은 policyDetails.json 파일의 예입니다.

```
{
    "ResourceTypes": [ "VOLUME"],
    "PolicyType": "EBS_SNAPSHOT_MANAGEMENT",
    "Schedules" : [
      {
        "Name": "sched1",
        "TagsToAdd": [
          {"Key":"createdby","Value":"dlm"}
        ],
        "CreateRule": {
          "CronExpression": "cron(0 9 1 * ? *)"
        },
        "CopyTags": true,
        "RetainRule":{
          "Interval": 1,
          "IntervalUnit": "DAYS"
        },
        "ArchiveRule": {
            "RetainRule":{
              "RetentionArchiveTier": {
                 "Interval": 90,
                 "IntervalUnit": "DAYS"
              }
            }
        }
      }
    ],
    "TargetTags": [
      {
        "Key": "Name",
        "Value": "Prod"
      }
    ]
}
```

예제 6 - 아카이브가 활성화된 개수 기반 일정이 포함된 스냅샷 수명 주기 정책

이 예제에서는 Purpose=Test로 태그가 지정된 볼륨을 대상으로 하는 스냅샷 수명 주기 정책을 생성합니다. 정책에는 매월 1일 09:00에 스냅샷을 생성하는 개수 기반 일정이 하나 있습니다. 일정은 생성 직후 스냅샷을 아카이브하고 아카이브 계층에 최대 3개의 스냅샷을 유지합니다.

```
aws dlm create-lifecycle-policy \
    --description "Copy snapshots to Outpost" \
    --state ENABLED \
    --execution-role-arn
 arn:aws:iam::12345678910:role/AWSDataLifecycleManagerDefaultRole \
    --policy-details file://policyDetails.json
```

다음은 policyDetails.json 파일의 예입니다.

```
{
    "ResourceTypes": [ "VOLUME"],
    "PolicyType": "EBS_SNAPSHOT_MANAGEMENT",
    "Schedules" : [
      {
        "Name": "sched1",
        "TagsToAdd": [
          {"Key":"createdby","Value":"dlm"}
        ],
        "CreateRule": {
          "CronExpression": "cron(0 9 1 * ? *)"
        },
        "CopyTags": true,
        "RetainRule":{
          "Count": 0
        },
        "ArchiveRule": {
            "RetainRule":{
              "RetentionArchiveTier": {
                 "Count": 3
              }
            }
        }
      }
    ],
    "TargetTags": [
      {
        "Key": "Purpose",
        "Value": "Test"
      }
    ]
}
```

스냅샷 수명 주기 정책 고려 사항

다음은 스냅샷 수명 주기 정책에 적용되는 일반적인 고려 사항입니다.

- 스냅샷 수명 주기 정책은 정책과 동일한 리전에 있는 인스턴스 또는 볼륨만 대상으로 합니다.
- 지정된 시작 시간 후 1시간 내에 첫 번째 스냅샷 생성 작업이 시작됩니다. 후속 스냅샷 생성 작업은 예약된 시간으로부터 1시간 이내에 시작됩니다.
- 볼륨 또는 인스턴스를 백업하도록 여러 정책을 생성할 수 있습니다. 볼륨에 12시간마다 스냅샷을 만드는 정책 A의 대상 태그인 태그 A와 24시간마다 스냅샷을 만드는 정책 B의 대상 태그인 태그 B, 이렇게 2개의 태그가 있다면 Amazon Data Lifecycle Manager는 양쪽 정책의 일정에 따라 스냅샷을 생성합니다. 여러 일정이 있는 단일 정책을 생성해도 동일한 결과를 얻을 수 있습니다. 예를 들어 태그 A만 대상으로 지정하는 단일 정책을 생성하고 12시간 간격과 24시간 간격의 2개 일정을 지정할 수 있습니다.
- 대상 리소스 태그는 대소문자를 구분합니다.
- 정책의 대상이 되는 리소스에서 대상 태그를 제거할 경우, Amazon Data Lifecycle Manager는 표준 티어와 아카이브 티어의 기존 스냅샷을 더 이상 관리하지 않습니다. 따라서 더 이상 필요하지 않은 경우 수동으로 스냅샷을 삭제해야 합니다.
- 인스턴스를 대상으로 지정하는 정책을 생성하는 경우 정책이 생성된 후 새 볼륨이 대상 인스턴스에 연결되면 새로 추가된 볼륨은 다음 정책 실행 시 백업에 포함됩니다. 정책이 실행되면 인스턴스에 연결된 모든 볼륨이 포함됩니다.
- 사용자 지정 cron 기반 예약이 있는 정책이 단일 스냅샷만 생성하도록 구성된 경우 이 정책은 보존 임계값에 도달할 때 스냅샷을 자동으로 삭제하지 않습니다. 더 이상 필요하지 않은 경우 스냅샷을 수동으로 삭제해야 합니다.
- 보존 기간이 생성 빈도보다 짧은 경과 시간 기반 정책을 생성할 경우, Amazon Data Lifecycle Manager는 항상 다음 스냅샷이 생성될 때까지 마지막 스냅샷을 보존합니다. 예를 들어 경과 시간 기반 정책에서 보존 기간이 7일인 스냅샷을 매달 한 개씩 생성하는 경우, 보존 기간이 7일이더라도 Amazon Data Lifecycle Manager는 각 스냅샷을 한 달 동안 보존합니다.

스냅샷 아카이빙에는 다음 고려 사항이 적용됩니다.

- 볼륨을 대상으로 하는 스냅샷 정책에 대해서만 스냅샷 아카이빙을 활성화할 수 있습니다.
- 각 정책에 대해 하나의 일정에 대해서만 아카이빙 규칙을 지정할 수 있습니다.
- 콘솔을 사용하는 경우 일정에 월별 또는 연별 생성 빈도가 있거나 일정에 생성 빈도가 28일 이상인 cron 표현식이 있는 경우에만 스냅샷 아카이빙을 활성화할 수 있습니다.

, AWS CLI AWS API 또는 AWS SDK를 사용하는 경우 일정에 생성 빈도가 28일 이상인 cron 표현식이 있는 경우에만 스냅샷 아카이빙을 활성화할 수 있습니다.

- 아카이브 계층의 최소 보존 기간은 90일입니다.
- 스냅샷이 아카이브되면 아카이브 계층으로 이동할 때 전체 스냅샷으로 변환됩니다. 이로 인해 스냅샷 저장 비용이 증가할 수 있습니다. 자세한 내용은 Amazon EBS 스냅샷 아카이빙 요금 및 결제 단원을 참조하십시오.
- 스냅샷이 아카이브되면 스냅샷에 대해 빠른 스냅샷 복원 및 스냅샷 공유가 비활성화됩니다.
- 윤년의 경우 보존 규칙으로 인해 아카이브 보존 기간이 90일 미만인 경우 Amazon Data Lifecycle Manager는 스냅샷이 최소 90일 동안 유지되도록 합니다.
- Amazon Data Lifecycle Manager에서 생성한 스냅샷을 수동으로 아카이브하고 일정의 보존 임계값에 도달해도 스냅샷이 계속 아카이브되는 경우 Amazon Data Lifecycle Manager는 더 이상 해당 스냅샷을 관리하지 않습니다. 그러나 일정의 보존 임계값에 도달하기 전에 스냅샷을 표준 계층으로 복원하는 경우 일정은 보존 규칙에 따라 스냅샷을 계속 관리합니다.
- Amazon Data Lifecycle Manager에 의해 아카이브된 스냅샷을 표준 계층으로 영구적으로 또는 일시적으로 복원하고 일정의 보존 임계값에 도달했을 때 스냅샷이 여전히 표준 계층에 있는 경우 Amazon Data Lifecycle Manager는 더 이상 스냅샷을 관리하지 않습니다. 그러나 일정의 보존 임계값에 도달하기 전에 스냅샷을 다시 아카이브하는 경우 보존 임계값에 도달하면 일정에서 스냅샷을 삭제합니다.
- Amazon Data Lifecycle Manager에 의해 아카이브된 스냅샷은 `Archived snapshots per volume` 및 `In-progress snapshot archives per account` 할당량에 포함됩니다.
- 일정에서 24시간 동안 재시도한 후에도 스냅샷을 아카이브할 수 없는 경우 스냅샷은 표준 계층에 남아 있고 아카이브 계층에서 삭제된 시간을 기준으로 삭제되도록 예약됩니다. 예를 들어, 일정에서 스냅샷을 120일 동안 아카이브하는 경우 아카이브에 실패한 후 영구적으로 삭제되기 전에 120일 동안 표준 계층에 남아 있습니다. 개수 기반 일정의 경우 스냅샷은 일정의 보존 횟수에 포함되지 않습니다.
- 스냅샷은 생성된 동일한 리전에 아카이브해야 합니다. 크로스 리전 복사 및 스냅샷 아카이빙을 활성화한 경우 Amazon Data Lifecycle Manager는 스냅샷 사본을 아카이브하지 않습니다.
- Amazon Data Lifecycle Manager에 의해 아카이브된 스냅샷에는 `aws:dlm:archived=true` 시스템 태그가 지정됩니다. 또한 아카이브가 활성화된 기간 기반 일정에 의해 생성된 스냅샷에는 스냅샷 아카이브 예약 날짜와 시간을 나타내는 `aws:dlm:expirationTime` 시스템 태그가 지정됩니다.

루트 볼륨 및 데이터(루트 아님) 볼륨을 제외할 때 다음 고려 사항이 적용됩니다.

- 부팅 볼륨을 제외하도록 선택하고 결과적으로 인스턴스에 연결된 모든 추가 데이터 볼륨을 제외하는 태그를 지정하면 Amazon Data Lifecycle Manager는 영향을 받는 인스턴스에 대한 스냅샷을 생성하지 않고 `SnapshotsCreateFailed` CloudWatch 지표를 내보냅니다. 자세한 내용은 CloudWatch를 사용하여 정책 모니터링 단원을 참조하십시오.

다음은 스냅샷 수명 주기 정책에 의해 대상으로 지정된 볼륨 삭제 또는 인스턴스 종료에 적용되는 고려 사항입니다.

- 볼륨을 삭제하거나 개수 기반 보존 일정으로 정책의 대상이 되는 인스턴스를 종료하는 경우 Amazon Data Lifecycle Manager는 삭제된 볼륨 또는 인스턴스에서 생성된 표준 계층 및 아카이브 계층의 스냅샷을 더 이상 관리하지 않습니다. 더 이상 필요하지 않은 이전의 스냅샷은 수동으로 삭제해야 합니다.
- 기간 기반 보존 일정을 사용하는 정책의 대상 볼륨을 삭제하거나 인스턴스를 종료하는 경우 정책은 정의된 일정에 따라 삭제된 볼륨 또는 인스턴스에서 생성된 표준 계층 및 아카이브 계층에서 스냅샷을 계속해서 삭제합니다(단, 마지막 스냅샷 제외). 더 이상 필요하지 않은 경우 마지막 스냅샷을 수동으로 삭제해야 합니다.

다음은 스냅샷 수명 주기 정책과 빠른 스냅샷 복원에 적용되는 고려 사항입니다.

- Amazon Data Lifecycle Manager는 크기가 16TiB 이하인 스냅샷에 대해서만 빠른 스냅샷 복원을 활성화할 수 있습니다. 자세한 내용은 Amazon EBS 빠른 스냅샷 복원 단원을 참조하십시오.
- 빠른 스냅샷 복원에 대해 사용 설정된 스냅샷은 정책을 삭제 또는 사용 중지하거나, 정책에 대한 빠른 스냅샷 복원을 사용 중지하거나, 가용 영역에 대한 빠른 스냅샷 복원을 사용 중지하더라도 사용 설정된 상태로 유지됩니다. 이러한 스냅샷에 대한 빠른 스냅샷 복원을 수동으로 사용 중지해야 합니다.
- 정책에서 빠른 스냅샷 복원을 사용 설정하고 빠른 스냅샷 복원에 대해 사용 설정할 수 있는 최대 스냅샷 수를 초과하는 경우 Amazon Data Lifecycle Manager는 예약된 대로 스냅샷을 생성하지만 빠른 스냅샷 복원에 대해 스냅샷을 사용 설정하지 않습니다. 빠른 스냅샷 복원에 대해 활성화된 스냅샷을 삭제한 후 Amazon Data Lifecycle Manager가 생성하는 다음 스냅샷은 빠른 스냅샷 복원에 대해 활성화됩니다.
- 스냅샷에 대해 빠른 스냅샷 복원을 사용 설정할 때 스냅샷을 최적화하려면 TiB당 60분 걸립니다. Amazon Data Lifecycle Manager에서 다음 스냅샷을 생성하기 전에 각 스냅샷이 완전히 최적화되도록 일정을 구성하는 것이 좋습니다.
- 인스턴스를 대상으로 하는 정책에 대해 빠른 스냅샷 복원을 활성화하면 Amazon Data Lifecycle Manager는 다중 볼륨 스냅샷 세트의 각 스냅샷에 대해 개별적으로 빠른 스냅샷 복원을 활성화합니

다. Amazon Data Lifecycle Manager가 다중 볼륨 스냅샷 세트의 스냅샷 중 하나에 대해 빠른 스냅샷 복원을 활성화하지 못하는 경우에도 스냅샷 세트의 나머지 스냅샷에 대해 빠른 스냅샷 복원을 활성화하려고 시도합니다.

- 특정 가용 영역의 스냅샷에 대해 빠른 스냅샷 복원이 활성화된 1분마다 요금이 청구됩니다. 요금은 최소 1시간으로 비례 청구됩니다. 자세한 내용은 요금 및 결제 단원을 참조하십시오.

> **Note**
>
> 수명 주기 정책의 구성에 따라 여러 가용 영역에서 빠른 스냅샷 복원에 대해 여러 스냅샷을 동시에 사용 설정할 수 있습니다.

스냅샷 수명 주기 정책 및 다중 연결 지원 볼륨에 적용되는 고려 사항은 다음과 같습니다.

- 동일한 다중 연결 지원 볼륨을 가지고 있는 대상 인스턴스 수명 주기 정책을 생성할 때 Amazon Data Lifecycle Manager에서는 연결된 각 인스턴스에 대해 볼륨의 스냅샷을 시작합니다. 타임스탬프 태그를 사용하여 연결된 인스턴스에서 생성된 시간 일관성 있는 스냅샷 집합을 식별합니다.

계정 간에 스냅샷을 공유할 때는 다음 사항을 고려해야 합니다.

- 암호화되지 않았거나 고객 관리형 키를 사용하여 암호화된 스냅샷만 공유할 수 있습니다.
- 기본 EBS 암호화 KMS 키로 암호화된 스냅샷은 공유할 수 없습니다.
- 암호화된 스냅샷을 공유하는 경우 소스 볼륨을 암호화하는 데 사용된 KMS 키도 대상 계정과 공유해야 합니다. 자세한 내용은AWS Key Management Service 개발자 안내서의 다른 계정의 사용자가 CMK를 사용하도록 허용을 참조하세요.

다음은 스냅샷 정책과 스냅샷 아카이빙에 적용되는 고려 사항입니다.

- 정책에 의해 생성된 스냅샷을 수동으로 아카이빙하고 정책의 보존 임계값에 도달했을 때 해당 스냅샷이 아카이브 계층에 있는 경우 Amazon Data Lifecycle Manager는 스냅샷을 삭제하지 않습니다. Amazon Data Lifecycle Manager는 아카이브 계층에 저장된 스냅샷을 관리하지 않습니다. 아카이브 계층에 저장된 스냅샷이 더 이상 필요하지 않으면 수동으로 삭제해야 합니다.

다음은 스냅샷 정책과 휴지통에 적용되는 고려 사항입니다.

- 정책의 보존 임계값에 도달하면 Amazon Data Lifecycle Manager가 스냅샷을 삭제하고 휴지통으로 보내고 사용자가 휴지통에서 스냅샷을 수동으로 복원하는 경우 더 이상 필요하지 않을 때 해당 스냅

샷을 수동으로 삭제해야 합니다. Amazon Data Lifecycle Manager는 더 이상 스냅샷을 관리하지 않습니다.

- 정책에 의해 생성된 스냅샷을 수동으로 삭제하고 정책의 보존 임계값에 도달했을 때 해당 스냅샷이 휴지통에 있는 경우 Amazon Data Lifecycle Manager는 스냅샷을 삭제하지 않습니다. Amazon Data Lifecycle Manager는 휴지통에 저장된 스냅샷을 관리하지 않습니다.

  정책의 보존 임계값에 도달하기 전에 스냅샷이 휴지통에서 복원되는 경우 정책의 보존 임계값에 도달하면 Amazon Data Lifecycle Manager가 스냅샷을 삭제합니다.

  정책의 보존 임계값에 도달한 후 스냅샷이 휴지통에서 복원되면 Amazon Data Lifecycle Manager가 더 이상 스냅샷을 삭제하지 않습니다. 더 이상 필요하지 않은 스냅샷은 수동으로 삭제해야 합니다.

다음은 오류 상태인 수명 주기 정책과 에 적용되는 고려 사항입니다.

- 수명 기반 보존 일정이 포함된 정책의 경우 정책이 `error` 상태일 때 만료되도록 설정된 스냅샷은 무기한 보존됩니다. 이러한 스냅샷은 수동으로 삭제해야 합니다. 정책을 다시 활성화하면 Amazon Data Lifecycle Manager가 스냅샷 보존 기간이 만료될 때 삭제를 재개합니다.
- 개수 기반 보존 일정이 있는 정책의 경우 `error` 상태인 동안 스냅샷 생성 및 삭제를 중단합니다. 정책을 다시 활성화하면 Amazon Data Lifecycle Manager가 스냅샷 생성을 재개하고 보존 임계값이 충족되면 스냅샷 삭제를 재개합니다.

다음은 스냅샷 정책과 스냅샷 잠금에 적용되는 고려 사항입니다.

- Amazon Data Lifecycle Manager가 생성한 스냅샷을 수동으로 잠그고 해당 보존 임곗값에 도달해도 스냅샷이 계속 잠겨 있는 경우 Amazon Data Lifecycle Manager는 더 이상 해당 스냅샷을 관리하지 않습니다. 더 이상 필요하지 않은 경우 스냅샷을 수동으로 삭제해야 합니다.
- Amazon Data Lifecycle Manager가 생성하고 빠른 스냅샷 복원이 활성화된 스냅샷을 수동으로 잠그고 해당 보존 임곗값에 도달해도 스냅샷이 계속 잠겨 있는 경우 Amazon Data Lifecycle Manager는 빠른 스냅샷 복원을 비활성화하거나 스냅샷을 삭제하지 않습니다. 더 이상 필요하지 않은 경우 수동으로 빠른 스냅샷 복원을 비활성화하고 스냅샷을 삭제해야 합니다.
- Amazon Data Lifecycle Manager가 생성한 스냅샷을 AMI에 수동으로 등록한 다음 해당 스냅샷을 잠그고 해당 보존 임곗값에 도달해도 스냅샷이 계속 잠겨 있고 AMI와 연결된 경우 Amazon Data Lifecycle Manager는 해당 스냅샷 삭제를 계속 시도합니다. AMI가 등록 취소되고 스냅샷이 잠금 해제되면 Amazon Data Lifecycle Manager는 자동으로 스냅샷을 삭제합니다.

추가 리소스

자세한 내용은 Amazon Data Lifecycle Manager 스토리지를 사용하여 Amazon EBS 스냅샷 및 AMI 관리 자동화 AWS 블로그를 참조하세요.

Data Lifecycle Manager를 사용하여 애플리케이션에 일관되게 적용되는 스냅샷 자동화

Amazon Data Lifecycle Manager를 사용하면 인스턴스를 대상으로 하는 스냅샷 수명 주기 정책에서 사전 및 사후 스크립트를 활성화하여 애플리케이션에 일관되게 적용되는 스냅샷을 자동화할 수 있습니다.

Amazon Data Lifecycle Manager는 AWS Systems Manager (Systems Manager)와 통합되어 애플리케이션에 일관되게 적용되는 스냅샷을 지원합니다. Amazon Data Lifecycle Manager는 사전 및 사후 스크립트가 포함된 Systems Manager(SSM) 명령 문서를 사용하여 애플리케이션에 일관되게 적용되는 스냅샷을 완성하는 데 필요한 작업을 자동화합니다. Amazon Data Lifecycle Manager는 스냅샷 생성을 시작하기 전에 사전 스크립트의 명령을 실행하여 I/O를 동결하고 플러시합니다. Amazon Data Lifecycle Manager가 스냅샷 생성을 시작한 후 사후 스크립트의 명령을 실행하여 I/O를 재개합니다.

Amazon Data Lifecycle Manager를 사용하면 다음과 같은 애플리케이션에 일관되게 적용되는 스냅샷을 자동화할 수 있습니다.

- 볼륨 섀도 복사본 서비스(VSS)를 사용하는 Windows 애플리케이션
- AWS 관리형 SSDM 문서를 사용하는 SAP HANA. 자세한 내용은 Amazon EBS snapshots for SAP HANA를 참조하세요.
- SSM 문서 템플릿을 사용하는 MySQL, PostgreSQL 또는 InterSystems IRIS와 같은 자체 관리형 데이터베이스

주제

- 사전 및 사후 스크립트 사용을 위한 요구 사항
- 애플리케이션에 일관되게 적용되는 스냅샷 시작하기
- Amazon Data Lifecycle Manager를 사용한 VSS 백업 고려 사항
- 애플리케이션에 일관되게 적용되는 스냅샷에 대한 공동 책임

사전 및 사후 스크립트 사용을 위한 요구 사항

다음 표에는 Amazon Data Lifecycle Manager에서 사전 및 사후 스크립트를 사용하기 위한 요구 사항이 요약되어 있습니다.

	애플리케이션 일치 스냅샷		
요구 사항	VSS 백업	사용자 지정 SSM 문서	그 외 사용 사례
SSM Agent installed and running on target instances	✓	✓	✓
VSS system requirements met on target instances	✓		
VSS enabled instance profile associated with target instances	✓		
VSS components installed on target instances	✓		
Prepare SSM document with pre and post script commands		✓	✓
Prepare Amazon Data Lifecycle Manager IAM role run pre and post scripts	✓	✓	✓
Create snapshot policy that targets instances and is	✓	✓	✓

애플리케이션 일치 스냅샷

configured for pre and post scripts

애플리케이션에 일관되게 적용되는 스냅샷 시작하기

이 섹션에서는 Amazon Data Lifecycle Manager를 사용하여 애플리케이션에 일관되게 적용되는 스냅샷을 자동화하기 위해 따라야 하는 단계를 설명합니다.

1단계: 대상 인스턴스 준비

Amazon Data Lifecycle Manager를 사용하여 애플리케이션에 일관되게 적용되는 스냅샷을 위한 대상 인스턴스를 준비해야 합니다. 사용 사례에 따라 다음 중 하나를 수행합니다.

Prepare for VSS Backups

VSS 백업을 위한 대상 인스턴스 준비

1. SSM Agent가 아직 설치되지 않은 경우 대상 인스턴스에 SSM Agent를 설치합니다. SSM Agent가 대상 인스턴스에 이미 설치되어 있는 경우 이 단계를 건너뜁니다.

   자세한 내용은 Windows Server용 EC2 인스턴스에서 SSM Agent 사용을 참조하세요.
2. SSM 에이전트가 실행 중인지 확인합니다. 자세한 내용은 SSM Agent 상태 확인 및 에이전트 시작을 참조하세요.
3. Amazon EC2 인스턴스용 Systems Manager를 설정합니다. 자세한 내용은 AWS Systems Manager 사용 설명서의 Amazon EC2 인스턴스용 Systems Manager 설정을 참조하세요.
4. VSS 백업에 대한 시스템 요구 사항이 충족되는지 확인합니다.
5. VSS 사용 인스턴스 프로파일을 대상 인스턴스에 연결합니다.
6. VSS 구성 요소를 설치합니다.

Prepare for SAP HANA backups

SAP HANA 백업을 위한 대상 인스턴스 준비

1. 대상 인스턴스에서 SAP HANA 환경을 준비합니다.

a. SAP HANA로 인스턴스를 설정합니다. 기존 SAP HANA 환경이 아직 없는 경우 SAP HANA Environment Setup on AWS를 참조할 수 있습니다.

b. 적절한 관리자 사용자로 SystemDB에 로그인합니다.

c. Amazon Data Lifecycle Manager에서 사용할 데이터베이스 백업 사용자를 생성합니다.

```
CREATE USER username PASSWORD password NO FORCE_FIRST_PASSWORD_CHANGE;
```

예를 들어, 다음 명령은 이름이 dlm_user이고 암호가 password인 사용자를 생성합니다.

```
CREATE USER dlm_user PASSWORD password NO FORCE_FIRST_PASSWORD_CHANGE;
```

d. 이전 단계에서 생성한 데이터베이스 백업 사용자에게 BACKUP OPERATOR 역할을 할당합니다.

```
GRANT BACKUP OPERATOR TO username
```

예를 들어, 다음 명령은 dlm_user라는 사용자에게 역할을 할당합니다.

```
GRANT BACKUP OPERATOR TO dlm_user
```

e. 운영 체제에 관리자(예: *sid*adm)로 로그인합니다.

f. 사용자가 정보를 입력하지 않고도 SAP HANA SSM 문서가 SAP HANA에 연결할 수 있도록 연결 정보를 저장할 hdbuserstore 항목을 생성합니다.

```
hdbuserstore set DLM_HANADB_SNAPSHOT_USER
 localhost:3hana_instance_number13 username password
```

예제:

```
hdbuserstore set DLM_HANADB_SNAPSHOT_USER localhost:30013 dlm_user password
```

g. 연결을 테스트합니다.

```
hdbsql -U DLM_HANADB_SNAPSHOT_USER "select * from dummy"
```

2. SSM Agent가 아직 설치되지 않은 경우 대상 인스턴스에 SSM Agent를 설치합니다. SSM Agent가 대상 인스턴스에 이미 설치되어 있는 경우 이 단계를 건너뜁니다.

   자세한 정보는 Linux용 EC2 인스턴스에 수동으로 SSM Agent 설치를 참조하세요.
3. SSM 에이전트가 실행 중인지 확인합니다. 자세한 내용은 SSM Agent 상태 확인 및 에이전트 시작을 참조하세요.
4. Amazon EC2 인스턴스용 Systems Manager를 설정합니다. 자세한 내용은 AWS Systems Manager 사용 설명서의 Amazon EC2 인스턴스용 Systems Manager 설정을 참조하세요.

Prepare for custom SSM documents

대상 인스턴스의 사용자 지정 SSM 문서 준비

1. SSM Agent가 아직 설치되지 않은 경우 대상 인스턴스에 SSM Agent를 설치합니다. SSM Agent가 대상 인스턴스에 이미 설치되어 있는 경우 이 단계를 건너뜁니다.
   - (Linux 인스턴스) Linux용 EC2 인스턴스에 수동으로 SSM Agent 설치
   - (Windows 인스턴스) Windows Server용 EC2 인스턴스에서 SSM 에이전트 작업
2. SSM 에이전트가 실행 중인지 확인합니다. 자세한 내용은 SSM Agent 상태 확인 및 에이전트 시작을 참조하세요.
3. Amazon EC2 인스턴스용 Systems Manager를 설정합니다. 자세한 내용은 AWS Systems Manager 사용 설명서의 Amazon EC2 인스턴스용 Systems Manager 설정을 참조하세요.

2단계: SSM 문서 준비

> **Note**
>
> 이 단계는 사용자 지정 SSM 문서에만 필요합니다. VSS 백업 또는 SAP HANA에는 필요하지 않습니다. VSS Backups 및 SAP HANA의 경우 Amazon Data Lifecycle Manager는 AWS 관리형 SSM 문서를 사용합니다.

MySQL, PostgreSQL, InterSystems IRIS와 같은 자체 관리형 데이터베이스에 대해 애플리케이션에 일관되게 적용되는 스냅샷을 자동화하는 경우 스냅샷 생성이 시작되기 전에 I/O를 동결하고 플러시하는 사전 스크립트와 스냅샷 생성이 시작된 후 I/O를 재개하는 사후 스크립트가 포함된 SSM 명령 문서를 생성해야 합니다.

MySQL, PostgreSQL 또는 InterSystems IRIS 데이터베이스에서 표준 구성을 사용하는 경우 아래 샘플 SSM 문서 내용을 사용하여 SSM 명령 문서를 생성할 수 있습니다. MySQL, PostgreSQL 또는 InterSystems IRIS 데이터베이스가 비표준 구성을 사용하는 경우 아래 샘플 내용을 SSM 명령 문서의 시작점으로 사용한 다음 요구 사항에 맞게 사용자 지정할 수 있습니다. 또는 SSM 문서를 처음부터 새로 생성하려면 아래의 빈 SSM 문서 템플릿을 사용하여 해당 문서 섹션에 사전 및 사후 명령을 추가합니다.

⚠ 다음 사항에 유의하세요.

- SSM 문서가 데이터베이스 구성에 필요한 올바른 작업을 수행하는지 확인하는 것은 사용자의 책임입니다.
- SSM 문서의 사전 및 사후 스크립트가 I/O를 성공적으로 동결, 플러시 및 재개할 수 있는 경우에만 스냅샷이 애플리케이션에 일관되게 적용됩니다.
- SSM 문서에는 `pre-script`, `post-script` 및 `dry-run`을 포함하여 `allowedValues`에 대한 필수 필드가 포함되어야 합니다. Amazon Data Lifecycle Manager는 이러한 섹션의 내용을 기반으로 인스턴스에서 명령을 실행합니다. SSM 문서에 이러한 섹션이 없는 경우 Amazon Data Lifecycle Manager는 해당 문서를 실패한 실행으로 간주합니다.

MySQL sample document content

```
###############################################################################
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.

# Permission is hereby granted, free of charge, to any person obtaining a copy of
 this
# software and associated documentation files (the "Software"), to deal in the
 Software
# without restriction, including without limitation the rights to use, copy, modify,
# merge, publish, distribute, sublicense, and/or sell copies of the Software, and to
# permit persons to whom the Software is furnished to do so.

# THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
 IMPLIED,
# INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
# PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT
# HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
# OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
# SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
```

```
###=============================================================================###
schemaVersion: '2.2'
description: Amazon Data Lifecycle Manager Pre/Post script for MySQL databases
parameters:
  executionId:
    type: String
    default: None
    description: (Required) Specifies the unique identifier associated with a pre
 and/or post execution
    allowedPattern: ^(None|[a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]
{4}-[a-fA-F0-9]{12})$
  command:
  # Data Lifecycle Manager will trigger the pre-script and post-script actions
 during policy execution.
  # 'dry-run' option is intended for validating the document execution without
 triggering any commands
  # on the instance. The following allowedValues will allow Data Lifecycle Manager
 to successfully
  # trigger pre and post script actions.
    type: String
    default: 'dry-run'
    description: (Required) Specifies whether pre-script and/or post-script should
 be executed.
    allowedValues:
    - pre-script
    - post-script
    - dry-run

mainSteps:
- action: aws:runShellScript
  description: Run MySQL Database freeze/thaw commands
  name: run_pre_post_scripts
  precondition:
    StringEquals:
    - platformType
    - Linux
  inputs:
    runCommand:
    - |
      #!/bin/bash


 ###=============================================================================###
      ### Error Codes
```

```
###============================================================================###
     # The following Error codes will inform Data Lifecycle Manager of the type of
error
     # and help guide handling of the error.
     # The Error code will also be emitted via AWS Eventbridge events in the
'cause' field.
     # 1 Pre-script failed during execution - 201
     # 2 Post-script failed during execution - 202
     # 3 Auto thaw occurred before post-script was initiated - 203
     # 4 Pre-script initiated while post-script was expected - 204
     # 5 Post-script initiated while pre-script was expected - 205
     # 6 Application not ready for pre or post-script initiation - 206

     ###========================================================================###
     ### Global variables
     ###========================================================================###
     START=$(date +%s)
     # For testing this script locally, replace the below with OPERATION=$1.
     OPERATION={{ command }}
     FS_ALREADY_FROZEN_ERROR='freeze failed: Device or resource busy'
     FS_ALREADY_THAWED_ERROR='unfreeze failed: Invalid argument'
     FS_BUSY_ERROR='mount point is busy'

     # Auto thaw is a fail safe mechanism to automatically unfreeze the application
after the
     # duration specified in the global variable below. Choose the duration based
on your
     # database application's tolerance to freeze.
     export AUTO_THAW_DURATION_SECS="60"

     # Add all pre-script actions to be performed within the function below
     execute_pre_script() {
         echo "INFO: Start execution of pre-script"
         # Check if filesystem is already frozen. No error code indicates that
filesystem
         # is not currently frozen and that the pre-script can proceed with
freezing the filesystem.
         check_fs_freeze
         # Execute the DB commands to flush the DB in preparation for snapshot
         snap_db
         # Freeze the filesystem. No error code indicates that filesystem was
succefully frozen
         freeze_fs
```

```
        echo "INFO: Schedule Auto Thaw to execute in ${AUTO_THAW_DURATION_SECS}
seconds."
        $(nohup bash -c execute_schedule_auto_thaw  >/dev/null 2>&1 &)
    }

    # Add all post-script actions to be performed within the function below
    execute_post_script() {
        echo "INFO: Start execution of post-script"
        # Unfreeze the filesystem. No error code indicates that filesystem was
successfully unfrozen.
        unfreeze_fs
        thaw_db
    }

    # Execute Auto Thaw to automatically unfreeze the application after the
duration configured
    # in the AUTO_THAW_DURATION_SECS global variable.
    execute_schedule_auto_thaw() {
        sleep ${AUTO_THAW_DURATION_SECS}
        execute_post_script
    }

    # Disable Auto Thaw if it is still enabled
    execute_disable_auto_thaw() {
        echo "INFO: Attempting to disable auto thaw if enabled"
        auto_thaw_pgid=$(pgrep -f execute_schedule_auto_thaw | xargs -i ps -hp {}
-o pgid)
        if [ -n "${auto_thaw_pgid}" ]; then
            echo "INFO: execute_schedule_auto_thaw process found with pgid
${auto_thaw_pgid}"
            sudo pkill -g ${auto_thaw_pgid}
            rc=$?
            if [ ${rc} != 0 ]; then
                echo "ERROR: Unable to kill execute_schedule_auto_thaw process.
retval=${rc}"
            else
                echo "INFO: Auto Thaw  has been disabled"
            fi
        fi
    }

    # Iterate over all the mountpoints and check if filesystem is already in
freeze state.
```

```
    # Return error code 204 if any of the mount points are already frozen.
    check_fs_freeze() {
        for target in $(lsblk -nlo MOUNTPOINTS)
        do
            # Freeze of the root and boot filesystems is dangerous and pre-script
does not freeze these filesystems.
            # Hence, we will skip the root and boot mountpoints while checking if
filesystem is in freeze state.
            if [ $target == '/' ]; then continue; fi
            if [[ "$target" == *"/boot"* ]]; then continue; fi

            error_message=$(sudo mount -o remount,noatime $target 2>&1)
            # Remount will be a no-op without a error message if the filesystem is
unfrozen.
            # However, if filesystem is already frozen, remount will fail with
busy error message.
            if [ $? -ne 0 ];then
                # If the filesystem is already in frozen, return error code 204
                if [[ "$error_message" == *"$FS_BUSY_ERROR"* ]];then
                    echo "ERROR: Filesystem ${target} already frozen. Return Error
Code: 204"
                    exit 204
                fi
                # If the check filesystem freeze failed due to any reason other
than the filesystem already frozen, return 201
                echo "ERROR: Failed to check_fs_freeze on mountpoint $target due
to error - $errormessage"
                exit 201
            fi
        done
    }

    # Iterate over all the mountpoints and freeze the filesystem.
    freeze_fs() {
        for target in $(lsblk -nlo MOUNTPOINTS)
        do
            # Freeze of the root and boot filesystems is dangerous. Hence, skip
filesystem freeze
            # operations for root and boot mountpoints.
            if [ $target == '/' ]; then continue; fi
            if [[ "$target" == *"/boot"* ]]; then continue; fi
            echo "INFO: Freezing $target"
            error_message=$(sudo fsfreeze -f $target 2>&1)
            if [ $? -ne 0 ];then
```

```
                # If the filesystem is already in frozen, return error code 204
                if [[ "$error_message" == *"$FS_ALREADY_FROZEN_ERROR"* ]]; then
                    echo "ERROR: Filesystem ${target} already frozen. Return Error
Code: 204"
                    sudo mysql -e 'UNLOCK TABLES;'
                    exit 204
                fi
                # If the filesystem freeze failed due to any reason other than the
filesystem already frozen, return 201
                echo "ERROR: Failed to freeze mountpoint $targetdue due to error -
$errormessage"
                thaw_db
                exit 201
            fi
            echo "INFO: Freezing complete on $target"
        done
    }

    # Iterate over all the mountpoints and unfreeze the filesystem.
    unfreeze_fs() {
        for target in $(lsblk -nlo MOUNTPOINTS)
        do
            # Freeze of the root and boot filesystems is dangerous and pre-script
does not freeze these filesystems.
            # Hence, will skip the root and boot mountpoints during unfreeze as
well.
            if [ $target == '/' ]; then continue; fi
            if [[ "$target" == *"/boot"* ]]; then continue; fi
            echo "INFO: Thawing $target"
            error_message=$(sudo fsfreeze -u $target 2>&1)
            # Check if filesystem is already unfrozen (thawed). Return error code
204 if filesystem is already unfrozen.
            if [ $? -ne 0 ]; then
                if [[ "$error_message" == *"$FS_ALREADY_THAWED_ERROR"* ]]; then
                    echo "ERROR: Filesystem ${target} is already in thaw state.
Return Error Code: 205"
                    exit 205
                fi
                # If the filesystem unfreeze failed due to any reason other than
the filesystem already unfrozen, return 202
                echo "ERROR: Failed to unfreeze mountpoint $targetdue due to error
- $errormessage"
                exit 202
            fi
```

```
            echo "INFO: Thaw complete on $target"
        done
    }

    snap_db() {
        # Run the flush command only when MySQL DB service is up and running
        sudo systemctl is-active --quiet mysqld.service
        if [ $? -eq 0 ]; then
            echo "INFO: Execute MySQL Flush and Lock command."
            sudo mysql -e 'FLUSH TABLES WITH READ LOCK;'
            # If the MySQL Flush and Lock command did not succeed, return error
code 201 to indicate pre-script failure
            if [ $? -ne 0 ]; then
                echo "ERROR: MySQL FLUSH TABLES WITH READ LOCK command failed."
                exit 201
            fi
            sync
        else
            echo "INFO: MySQL service is inactive. Skipping execution of MySQL
Flush and Lock command."
        fi
    }

    thaw_db() {
        # Run the unlock command only when MySQL DB service is up and running
        sudo systemctl is-active --quiet mysqld.service
        if [ $? -eq 0 ]; then
            echo "INFO: Execute MySQL Unlock"
            sudo mysql -e 'UNLOCK TABLES;'
        else
            echo "INFO: MySQL service is inactive. Skipping execution of MySQL
Unlock command."
        fi
    }

    export -f execute_schedule_auto_thaw
    export -f execute_post_script
    export -f unfreeze_fs
    export -f thaw_db

    # Debug logging for parameters passed to the SSM document
    echo "INFO: ${OPERATION} starting at $(date) with executionId:
${EXECUTION_ID}"
```

```
      # Based on the command parameter value execute the function that supports
      # pre-script/post-script operation
      case ${OPERATION} in
          pre-script)
              execute_pre_script
              ;;
          post-script)
              execute_post_script
              execute_disable_auto_thaw
              ;;
          dry-run)
              echo "INFO: dry-run option invoked - taking no action"
              ;;
          *)
              echo "ERROR: Invalid command parameter passed. Please use either pre-
script, post-script, dry-run."
              exit 1 # return failure
              ;;
      esac

      END=$(date +%s)
      # Debug Log for profiling the script time
      echo "INFO: ${OPERATION} completed at $(date). Total runtime: $((${END} -
 ${START})) seconds."
```

PostgreSQL sample document content

```
###############################################################################
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.

# Permission is hereby granted, free of charge, to any person obtaining a copy of
 this
# software and associated documentation files (the "Software"), to deal in the
 Software
# without restriction, including without limitation the rights to use, copy, modify,
# merge, publish, distribute, sublicense, and/or sell copies of the Software, and to
# permit persons to whom the Software is furnished to do so.

# THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
 IMPLIED,
# INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
# PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT
# HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
```

```
# OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
# SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
###============================================================================###
schemaVersion: '2.2'
description: Amazon Data Lifecycle Manager Pre/Post script for PostgreSQL databases
parameters:
  executionId:
    type: String
    default: None
    description: (Required) Specifies the unique identifier associated with a pre
 and/or post execution
    allowedPattern: ^(None|[a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]
{4}-[a-fA-F0-9]{12})$
  command:
  # Data Lifecycle Manager will trigger the pre-script and post-script actions
 during policy execution.
  # 'dry-run' option is intended for validating the document execution without
 triggering any commands
  # on the instance. The following allowedValues will allow Data Lifecycle Manager
 to successfully
  # trigger pre and post script actions.
    type: String
    default: 'dry-run'
    description: (Required) Specifies whether pre-script and/or post-script should
 be executed.
    allowedValues:
    - pre-script
    - post-script
    - dry-run

mainSteps:
- action: aws:runShellScript
  description: Run PostgreSQL Database freeze/thaw commands
  name: run_pre_post_scripts
  precondition:
    StringEquals:
    - platformType
    - Linux
  inputs:
    runCommand:
    - |
      #!/bin/bash
```

```
###============================================================================###
     ### Error Codes

###============================================================================###
     # The following Error codes will inform Data Lifecycle Manager of the type of
error
     # and help guide handling of the error.
     # The Error code will also be emitted via AWS Eventbridge events in the
'cause' field.
     # 1 Pre-script failed during execution - 201
     # 2 Post-script failed during execution - 202
     # 3 Auto thaw occurred before post-script was initiated - 203
     # 4 Pre-script initiated while post-script was expected - 204
     # 5 Post-script initiated while pre-script was expected - 205
     # 6 Application not ready for pre or post-script initiation - 206


###============================================================================###
     ### Global variables

###============================================================================###
     START=$(date +%s)
     OPERATION={{ command }}
     FS_ALREADY_FROZEN_ERROR='freeze failed: Device or resource busy'
     FS_ALREADY_THAWED_ERROR='unfreeze failed: Invalid argument'
     FS_BUSY_ERROR='mount point is busy'

     # Auto thaw is a fail safe mechanism to automatically unfreeze the application
after the
     # duration specified in the global variable below. Choose the duration based
on your
     # database application's tolerance to freeze.
     export AUTO_THAW_DURATION_SECS="60"

     # Add all pre-script actions to be performed within the function below
     execute_pre_script() {
         echo "INFO: Start execution of pre-script"
         # Check if filesystem is already frozen. No error code indicates that
filesystem
         # is not currently frozen and that the pre-script can proceed with
freezing the filesystem.
         check_fs_freeze
         # Execute the DB commands to flush the DB in preparation for snapshot
```

```
        snap_db
        # Freeze the filesystem. No error code indicates that filesystem was
succefully frozen
        freeze_fs

        echo "INFO: Schedule Auto Thaw to execute in ${AUTO_THAW_DURATION_SECS}
seconds."
        $(nohup bash -c execute_schedule_auto_thaw  >/dev/null 2>&1 &)
    }

    # Add all post-script actions to be performed within the function below
    execute_post_script() {
        echo "INFO: Start execution of post-script"
        # Unfreeze the filesystem. No error code indicates that filesystem was
successfully unfrozen
        unfreeze_fs
    }

    # Execute Auto Thaw to automatically unfreeze the application after the
duration configured
    # in the AUTO_THAW_DURATION_SECS global variable.
    execute_schedule_auto_thaw() {
        sleep ${AUTO_THAW_DURATION_SECS}
        execute_post_script
    }

    # Disable Auto Thaw if it is still enabled
    execute_disable_auto_thaw() {
        echo "INFO: Attempting to disable auto thaw if enabled"
        auto_thaw_pgid=$(pgrep -f execute_schedule_auto_thaw | xargs -i ps -hp {}
-o pgid)
        if [ -n "${auto_thaw_pgid}" ]; then
            echo "INFO: execute_schedule_auto_thaw process found with pgid
${auto_thaw_pgid}"
            sudo pkill -g ${auto_thaw_pgid}
            rc=$?
            if [ ${rc} != 0 ]; then
                echo "ERROR: Unable to kill execute_schedule_auto_thaw process.
retval=${rc}"
            else
                echo "INFO: Auto Thaw  has been disabled"
            fi
        fi
    }
```

```
    # Iterate over all the mountpoints and check if filesystem is already in
freeze state.
    # Return error code 204 if any of the mount points are already frozen.
    check_fs_freeze() {
        for target in $(lsblk -nlo MOUNTPOINTS)
        do
            # Freeze of the root and boot filesystems is dangerous and pre-script
does not freeze these filesystems.
            # Hence, we will skip the root and boot mountpoints while checking if
filesystem is in freeze state.
            if [ $target == '/' ]; then continue; fi
            if [[ "$target" == *"/boot"* ]]; then continue; fi

            error_message=$(sudo mount -o remount,noatime $target 2>&1)
            # Remount will be a no-op without a error message if the filesystem is
unfrozen.
            # However, if filesystem is already frozen, remount will fail with
busy error message.
            if [ $? -ne 0 ];then
                # If the filesystem is already in frozen, return error code 204
                if [[ "$error_message" == *"$FS_BUSY_ERROR"* ]];then
                    echo "ERROR: Filesystem ${target} already frozen. Return Error
Code: 204"
                    exit 204
                fi
                # If the check filesystem freeze failed due to any reason other
than the filesystem already frozen, return 201
                echo "ERROR: Failed to check_fs_freeze on mountpoint $target due
to error - $errormessage"
                exit 201
            fi
        done
    }

    # Iterate over all the mountpoints and freeze the filesystem.
    freeze_fs() {
        for target in $(lsblk -nlo MOUNTPOINTS)
        do
            # Freeze of the root and boot filesystems is dangerous. Hence, skip
filesystem freeze
            # operations for root and boot mountpoints.
            if [ $target == '/' ]; then continue; fi
            if [[ "$target" == *"/boot"* ]]; then continue; fi
```

```
            echo "INFO: Freezing $target"
            error_message=$(sudo fsfreeze -f $target 2>&1)
            if [ $? -ne 0 ];then
                # If the filesystem is already in frozen, return error code 204
                if [[ "$error_message" == *"$FS_ALREADY_FROZEN_ERROR"* ]]; then
                    echo "ERROR: Filesystem ${target} already frozen. Return Error
Code: 204"
                    exit 204
                fi
                # If the filesystem freeze failed due to any reason other than the
filesystem already frozen, return 201
                echo "ERROR: Failed to freeze mountpoint $targetdue due to error -
$errormessage"
                exit 201
            fi
            echo "INFO: Freezing complete on $target"
        done
    }

    # Iterate over all the mountpoints and unfreeze the filesystem.
    unfreeze_fs() {
        for target in $(lsblk -nlo MOUNTPOINTS)
        do
            # Freeze of the root and boot filesystems is dangerous and pre-script
does not freeze these filesystems.
            # Hence, will skip the root and boot mountpoints during unfreeze as
well.
            if [ $target == '/' ]; then continue; fi
            if [[ "$target" == *"/boot"* ]]; then continue; fi
            echo "INFO: Thawing $target"
            error_message=$(sudo fsfreeze -u $target 2>&1)
            # Check if filesystem is already unfrozen (thawed). Return error code
204 if filesystem is already unfrozen.
            if [ $? -ne 0 ]; then
                if [[ "$error_message" == *"$FS_ALREADY_THAWED_ERROR"* ]]; then
                    echo "ERROR: Filesystem ${target} is already in thaw state.
Return Error Code: 205"
                    exit 205
                fi
                # If the filesystem unfreeze failed due to any reason other than
the filesystem already unfrozen, return 202
                echo "ERROR: Failed to unfreeze mountpoint $targetdue due to error
- $errormessage"
                exit 202
```

```
            fi
            echo "INFO: Thaw complete on $target"
        done
    }

    snap_db() {
        # Run the flush command only when PostgreSQL DB service is up and running
        sudo systemctl is-active --quiet postgresql
        if [ $? -eq 0 ]; then
            echo "INFO: Execute Postgres CHECKPOINT"
            # PostgreSQL command to flush the transactions in memory to disk
            sudo -u postgres psql -c 'CHECKPOINT;'
            # If the PostgreSQL Command did not succeed, return error code 201 to
indicate pre-script failure
            if [ $? -ne 0 ]; then
                echo "ERROR: Postgres CHECKPOINT command failed."
                exit 201
            fi
            sync
        else
            echo "INFO: PostgreSQL service is inactive. Skipping execution of
CHECKPOINT command."
        fi
    }

    export -f execute_schedule_auto_thaw
    export -f execute_post_script
    export -f unfreeze_fs

    # Debug logging for parameters passed to the SSM document
    echo "INFO: ${OPERATION} starting at $(date) with executionId:
${EXECUTION_ID}"

    # Based on the command parameter value execute the function that supports
    # pre-script/post-script operation
    case ${OPERATION} in
        pre-script)
            execute_pre_script
            ;;
        post-script)
            execute_post_script
            execute_disable_auto_thaw
            ;;
        dry-run)
```

```
              echo "INFO: dry-run option invoked - taking no action"
              ;;
          *)
              echo "ERROR: Invalid command parameter passed. Please use either pre-
script, post-script, dry-run."
              exit 1 # return failure
              ;;
      esac

      END=$(date +%s)
      # Debug Log for profiling the script time
      echo "INFO: ${OPERATION} completed at $(date). Total runtime: $((${END} -
 ${START})) seconds."
```

InterSystems IRIS sample document content

```
###############################################################################
# MIT License
#
# Copyright (c) 2024 InterSystems
#
# Permission is hereby granted, free of charge, to any person obtaining a copy
# of this software and associated documentation files (the "Software"), to deal
# in the Software without restriction, including without limitation the rights
# to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
# copies of the Software, and to permit persons to whom the Software is
# furnished to do so, subject to the following conditions:
#
# The above copyright notice and this permission notice shall be included in all
# copies or substantial portions of the Software.
#
# THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
# IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
# FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
# AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
# LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
# OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
# SOFTWARE.
###############################################################################
schemaVersion: '2.2'
description: SSM Document Template for Amazon Data Lifecycle Manager Pre/Post script
 feature for InterSystems IRIS.
parameters:
```

```
  executionId:
    type: String
    default: None
    description: Specifies the unique identifier associated with a pre and/or post
 execution
    allowedPattern: ^(None|[a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]
{4}-[a-fA-F0-9]{12})$
  command:
    type: String
    # Data Lifecycle Manager will trigger the pre-script and post-script actions.
 You can also use this SSM document with 'dry-run' for manual testing purposes.
    default: 'dry-run'
    description: (Required) Specifies whether pre-script and/or post-script should
 be executed.
    #The following allowedValues will allow Data Lifecycle Manager to successfully
 trigger pre and post script actions.
    allowedValues:
    - pre-script
    - post-script
    - dry-run

mainSteps:
- action: aws:runShellScript
  description: Run InterSystems IRIS Database freeze/thaw commands
  name: run_pre_post_scripts
  precondition:
    StringEquals:
    - platformType
    - Linux
  inputs:
    runCommand:
    - |
      #!/bin/bash

 ###============================================================================###
      ### Global variables

 ###============================================================================###
      DOCKER_NAME=iris
      LOGDIR=./
      EXIT_CODE=0
      OPERATION={{ command }}
      START=$(date +%s)
```

```
     # Check if Docker is installed
     # By default if Docker is present, script assumes that InterSystems IRIS is
 running in Docker
     # Leave only the else block DOCKER_EXEC line, if you run InterSystems IRIS
 non-containerised (and Docker is present).
     # Script assumes irissys user has OS auth enabled, change the OS user or
 supply login/password depending on your configuration.
     if command -v docker &> /dev/null
     then
       DOCKER_EXEC="docker exec $DOCKER_NAME"
     else
       DOCKER_EXEC="sudo -i -u irissys"
     fi


     # Add all pre-script actions to be performed within the function below
     execute_pre_script() {
       echo "INFO: Start execution of pre-script"

       # find all iris running instances
       iris_instances=$($DOCKER_EXEC iris qall 2>/dev/null | tail -n +3 | grep
 '^up' | cut -c5-  | awk '{print $1}')
       echo "`date`: Running iris instances $iris_instances"

       # Only for running instances
       for INST in $iris_instances; do

         echo "`date`: Attempting to freeze $INST"

         # Detailed instances specific log
         LOGFILE=$LOGDIR/$INST-pre_post.log

         #check Freeze status before starting
         $DOCKER_EXEC irissession $INST -U '%SYS'
 "##Class(Backup.General).IsWDSuspendedExt()"
         freeze_status=$?
         if [ $freeze_status -eq 5 ]; then
           echo "`date`:   ERROR: $INST IS already FROZEN"
           EXIT_CODE=204
         else
           echo "`date`:   $INST is not frozen"
           # Freeze
           # Docs: https://docs.intersystems.com/irislatest/csp/documatic/
%25CSP.Documatic.cls?LIBRARY=%25SYS&CLASSNAME=Backup.General#ExternalFreeze
```

```
            $DOCKER_EXEC irissession $INST -U '%SYS'
"##Class(Backup.General).ExternalFreeze(\"$LOGFILE\",,,,,,600,,,300)"
            status=$?

            case $status in
              5) echo "`date`:   $INST IS FROZEN"
                ;;
              3) echo "`date`:   $INST FREEZE FAILED"
                EXIT_CODE=201
                ;;
              *) echo "`date`:   ERROR: Unknown status code: $status"
                EXIT_CODE=201
                ;;
            esac
            echo "`date`:   Completed freeze of $INST"
          fi
        done
        echo "`date`: Pre freeze script finished"
      }

      # Add all post-script actions to be performed within the function below
      execute_post_script() {
        echo "INFO: Start execution of post-script"

        # find all iris running instances
        iris_instances=$($DOCKER_EXEC iris qall 2>/dev/null | tail -n +3 | grep
'^up' | cut -c5-  | awk '{print $1}')
        echo "`date`: Running iris instances $iris_instances"

        # Only for running instances
        for INST in $iris_instances; do

          echo "`date`: Attempting to thaw $INST"

          # Detailed instances specific log
          LOGFILE=$LOGDIR/$INST-pre_post.log

          #check Freeze status befor starting
          $DOCKER_EXEC irissession $INST -U '%SYS'
"##Class(Backup.General).IsWDSuspendedExt()"
          freeze_status=$?
          if [ $freeze_status -eq 5 ]; then
            echo "`date`:  $INST is in frozen state"
            # Thaw
```

```
            # Docs: https://docs.intersystems.com/irislatest/csp/documatic/
%25CSP.Documatic.cls?LIBRARY=%25SYS&CLASSNAME=Backup.General#ExternalFreeze
            $DOCKER_EXEC irissession $INST -U%SYS
 "##Class(Backup.General).ExternalThaw(\"$LOGFILE\")"
            status=$?

            case $status in
              5) echo "`date`:   $INST IS THAWED"
                  $DOCKER_EXEC irissession $INST -U%SYS
 "##Class(Backup.General).ExternalSetHistory(\"$LOGFILE\")"
                ;;
              3) echo "`date`:   $INST THAW FAILED"
                  EXIT_CODE=202
                ;;
              *) echo "`date`:   ERROR: Unknown status code: $status"
                  EXIT_CODE=202
                ;;
            esac
            echo "`date`:   Completed thaw of $INST"
          else
            echo "`date`:   ERROR: $INST IS already THAWED"
            EXIT_CODE=205
          fi
        done
        echo "`date`: Post thaw script finished"
      }

      # Debug logging for parameters passed to the SSM document
        echo "INFO: ${OPERATION} starting at $(date) with executionId:
 ${EXECUTION_ID}"

      # Based on the command parameter value execute the function that supports
      # pre-script/post-script operation
      case ${OPERATION} in
        pre-script)
          execute_pre_script
          ;;
        post-script)
          execute_post_script
            ;;
        dry-run)
          echo "INFO: dry-run option invoked - taking no action"
          ;;
        *)
```

```
          echo "ERROR: Invalid command parameter passed. Please use either pre-
script, post-script, dry-run."
          # return failure
          EXIT_CODE=1
          ;;
      esac

      END=$(date +%s)
      # Debug Log for profiling the script time
      echo "INFO: ${OPERATION} completed at $(date). Total runtime: $((${END} -
 ${START})) seconds."
      exit $EXIT_CODE
```

자세한 내용은 GitHub 리포지토리를 참조하세요.

Empty document template

```
###======================================================================###
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.

# Permission is hereby granted, free of charge, to any person obtaining a copy of
 this
# software and associated documentation files (the "Software"), to deal in the
 Software
# without restriction, including without limitation the rights to use, copy, modify,
# merge, publish, distribute, sublicense, and/or sell copies of the Software, and to
# permit persons to whom the Software is furnished to do so.

# THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
 IMPLIED,
# INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
# PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT
# HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
# OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
# SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
###======================================================================###
schemaVersion: '2.2'
description: SSM Document Template for Amazon Data Lifecycle Manager Pre/Post script
 feature
parameters:
  executionId:
    type: String
    default: None
```

```
    description: (Required) Specifies the unique identifier associated with a pre
 and/or post execution
    allowedPattern: ^(None|[a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]
{4}-[a-fA-F0-9]{12})$
  command:
  # Data Lifecycle Manager will trigger the pre-script and post-script actions
 during policy execution.
  # 'dry-run' option is intended for validating the document execution without
 triggering any commands
  # on the instance. The following allowedValues will allow Data Lifecycle Manager
 to successfully
  # trigger pre and post script actions.
    type: String
    default: 'dry-run'
    description: (Required) Specifies whether pre-script and/or post-script should
 be executed.
    allowedValues:
    - pre-script
    - post-script
    - dry-run

mainSteps:
- action: aws:runShellScript
  description: Run Database freeze/thaw commands
  name: run_pre_post_scripts
  precondition:
    StringEquals:
    - platformType
    - Linux
  inputs:
    runCommand:
    - |
      #!/bin/bash


 ###============================================================================###
      ### Error Codes

 ###============================================================================###
      # The following Error codes will inform Data Lifecycle Manager of the type of
 error
      # and help guide handling of the error.
      # The Error code will also be emitted via AWS Eventbridge events in the
 'cause' field.
```

```
    # 1 Pre-script failed during execution - 201
    # 2 Post-script failed during execution - 202
    # 3 Auto thaw occurred before post-script was initiated - 203
    # 4 Pre-script initiated while post-script was expected - 204
    # 5 Post-script initiated while pre-script was expected - 205
    # 6 Application not ready for pre or post-script initiation - 206


###============================================================================###
    ### Global variables

###============================================================================###
    START=$(date +%s)
    # For testing this script locally, replace the below with OPERATION=$1.
    OPERATION={{ command }}

    # Add all pre-script actions to be performed within the function below
    execute_pre_script() {
        echo "INFO: Start execution of pre-script"
    }

    # Add all post-script actions to be performed within the function below
    execute_post_script() {
        echo "INFO: Start execution of post-script"
    }

    # Debug logging for parameters passed to the SSM document
    echo "INFO: ${OPERATION} starting at $(date) with executionId:
${EXECUTION_ID}"

    # Based on the command parameter value execute the function that supports
    # pre-script/post-script operation
    case ${OPERATION} in
        pre-script)
            execute_pre_script
            ;;
        post-script)
            execute_post_script
            ;;
        dry-run)
            echo "INFO: dry-run option invoked - taking no action"
            ;;
        *)
```

```
                echo "ERROR: Invalid command parameter passed. Please use either pre-
script, post-script, dry-run."
                exit 1 # return failure
                ;;
        esac

        END=$(date +%s)
        # Debug Log for profiling the script time
        echo "INFO: ${OPERATION} completed at $(date). Total runtime: $((${END} -
 ${START})) seconds."
```

SSM 문서 내용이 있는 경우 다음 절차 중 하나를 사용하여 사용자 지정 SSM 문서를 생성합니다.

Console

SSM 명령 문서 생성

1. https://console.aws.amazon.com//systems-manager/ AWS Systems Manager 콘솔을 엽니다.
2. 탐색 창에서 문서를 선택한 다음 문서 생성, 명령 또는 세션을 선택합니다.
3. [이름(Name)]에 문서에 대한 설명이 포함된 이름을 입력합니다.
4. 대상 유형에서 /AWS::EC2::Instance를 선택합니다.
5. 문서 유형에서 명령을 선택합니다.
6. 콘텐츠 필드에서 YAML을 선택한 다음 문서 내용을 붙여넣습니다.
7. 문서 태그 섹션에서 태그 키가 `DLMScriptsAccess`이고 태그 값이 `true`인 태그를 추가합니다.

   > **Important**
   >
   > `DLMScriptsAccess:true` 태그는 3단계: Amazon Data Lifecycle Manager IAM 역할 준비에 사용되는 AWSDataLifecycleManagerSSMFullAccess AWS 관리형 정책에 필요합니다. 정책은 `aws:ResourceTag` 조건 키를 사용하여 이 태그가 있는 SSM 문서에 대한 액세스를 제한합니다.

8. 문서 생성을 선택합니다.

AWS CLI

SSM 명령 문서 생성

create-document 명령을 사용합니다. --name에 문서에 대한 설명이 포함된 이름을 지정합니다. --document-type에서 Command를 지정합니다. --content에 대해 SSM 문서 내용이 포함된 .yaml 파일의 경로를 지정합니다. --tags에서 "Key=DLMScriptsAccess,Value=true"를 지정합니다.

```
$ aws ssm create-document \
--content file://path/to/file/documentContent.yaml \
--name "document_name" \
--document-type "Command" \
--document-format YAML \
--tags "Key=DLMScriptsAccess,Value=true"
```

3단계: Amazon Data Lifecycle Manager IAM 역할 준비

> **Note**
>
> 이 단계는 다음과 같은 경우 필요합니다.
>
> - 사용자 지정 IAM 역할을 사용하는 사전/사후 스크립트 지원 스냅샷 정책을 생성하거나 업데이트합니다.
> - 명령줄을 사용하여 기본값을 사용하는 사전/사후 스크립트 지원 스냅샷 정책을 생성하거나 업데이트합니다.
>
> 콘솔을 사용하여 스냅샷 관리를 위한 기본 역할(AWSDataLifecycleManagerDefaultRole)을 사용하는 사전/사후 스크립트 지원 스냅샷 정책을 생성하거나 업데이트하려면 이 단계를 건너뜁니다. 이 경우 AWSDataLifecycleManagerSSMFullAccess 정책이 해당 역할에 자동으로 연결됩니다.

정책에 사용하는 IAM 역할이 정책의 대상이 되는 인스턴스에서 사전 및 사후 스크립트를 실행하는 데 필요한 SSM 작업을 수행할 수 있는 권한을 Amazon Data Lifecycle Manager에 부여하는지 확인해야 합니다.

Amazon Data Lifecycle Manager는 필요한 권한이 포함된 관리형 정책(AWSDataLifecycleManagerSSMFullAccess)을 제공합니다. 스냅샷을 관리하기 위해이 정책을 IAM 역할에 연결하여 권한이 포함되어 있는지 확인할 수 있습니다.

> **Important**
>
> AWSDataLifecycleManagerSSMFullAccess 관리형 정책은 사전 및 사후 스크립트를 사용할 때 `aws:ResourceTag` 조건 키를 사용하여 특정 SSM 문서에 대한 액세스를 제한합니다. Amazon Data Lifecycle Manager가 SSM 문서에 액세스하도록 허용하려면 SSM 문서에 로 태그가 지정되어 있는지 확인해야 합니다`DLMScriptsAccess:true`.

또는 사용자 지정 정책을 수동으로 생성하거나 사용하는 IAM 역할에 필요한 권한을 직접 할당할 수 있습니다. AWSDataLifecycleManagerSSMFullAccess 관리형 정책에 정의된 동일한 권한을 사용할 수 있지만 `aws:ResourceTag` 조건 키는 선택 사항입니다. 해당 조건 키를 포함하지 않기로 결정하면 SSM 문서에 `DLMScriptsAccess:true`로 태그를 지정할 필요가 없습니다.

다음 방법 중 하나를 사용하여 IAM 역할에 AWSDataLifecycleManagerSSMFullAccess 정책을 추가합니다.

Console

사용자 지정 역할에 관리형 정책 연결

1. IAM 콘솔(https://console.aws.amazon.com/iam/)을 엽니다.
2. 탐색 창에서 [역할(Roles)]을 선택합니다.
3. 스냅샷 관리를 위한 사용자 지정 역할을 검색하고 선택합니다.
4. 권한 탭에서 권한 추가, 정책 연결을 선택합니다.
5. AWSDataLifecycleManagerSSMFullAccess 관리형 정책을 검색하여 선택한 다음 권한 추가를 선택합니다.

AWS CLI

사용자 지정 역할에 관리형 정책 연결

attach-role-policy 명령을 사용합니다. `---role-name`에 대해 사용자 지정 역할의 이름을 지정합니다. `--policy-arn`에서 `arn:aws:iam::aws:policy/AWSDataLifecycleManagerSSMFullAccess`를 지정합니다.

```
$ aws iam attach-role-policy \
--policy-arn arn:aws:iam::aws:policy/AWSDataLifecycleManagerSSMFullAccess \
--role-name your_role_name
```

4단계: 스냅샷 수명 주기 정책 생성

애플리케이션에 일관되게 적용되는 스냅샷을 자동화하려면 인스턴스를 대상으로 하는 스냅샷 수명 주기 정책을 생성하고 해당 정책에 대한 사전 및 사후 스크립트를 구성해야 합니다.

Console

스냅샷 수명 주기 정책 생성

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Elastic Block Store, Lifecycle Manager 및 수명 주기 정책 생성을 차례로 선택합니다.
3. 정책 유형 선택(Select policy type) 화면에서 EBS 스냅샷 정책(EBS snapshot policy)을 선택한 후 다음(Next)을 선택합니다.
4. 대상 리소스(Target resources) 섹션에서 다음을 수행합니다.
   a. 대상 리소스 유형에서는 `Instance`를 선택합니다.
   b. 대상 리소스 태그에서 백업할 인스턴스를 식별하는 리소스 태그를 지정합니다. 지정된 태그가 있는 리소스만 백업됩니다.
5. IAM 역할에서 AWSDataLifecycleManagerDefaultRole(스냅샷 관리를 위한 기본 역할)을 선택하거나 사전 및 사후 스크립트용으로 생성하고 준비한 사용자 지정 역할을 선택합니다.
6. 필요에 따라 일정과 추가 옵션을 구성합니다. 유지 관리 기간과 같이 워크로드에 맞는 기간으로 스냅샷 생성 시간을 예약하는 것이 좋습니다.

   SAP HANA의 경우 빠른 스냅샷 복원을 활성화하는 것이 좋습니다.

   > **Note**
   > VSS 백업에 대해 일정을 활성화하는 경우 특정 데이터 볼륨 제외 또는 소스에서 태그 복사를 활성화할 수 없습니다.

7. 사전 및 사후 스크립트 섹션에서 사전 및 사후 스크립트 활성화를 선택하고 워크로드에 따라 다음을 수행합니다.
   - Windows 애플리케이션의 애플리케이션에 일관되게 적용되는 스냅샷을 생성하려면 VSS 백업을 선택합니다.
   - SAP HANA 워크로드의 애플리케이션에 일관되게 적용되는 스냅샷을 생성하려면 SAP HANA를 선택합니다.

- 사용자 지정 SSM 문서를 사용하여 자체 관리형 MySQL, PostgreSQL 또는 InterSystems IRIS 데이터베이스를 비롯한 다른 모든 데이터베이스와 워크로드의 애플리케이션에 일관되게 적용되는 스냅샷을 생성하려면 사용자 지정 SSM 문서를 선택합니다.

  1. 자동화 옵션에서 사전 및 사후 스크립트를 선택합니다.
  2. SSM 문서에서 준비한 SSM 문서를 선택합니다.

8. 선택한 옵션에 따라 다음과 같은 추가 옵션을 구성합니다.

- 스크립트 제한 시간 - (사용자 지정 SSM 문서에만 해당) Amazon Data Lifecycle Manager에서 완료되지 않은 스크립트 실행 시도가 실패하기 전까지의 제한 시간입니다. 스크립트가 제한 시간 내에 완료되지 않으면 Amazon Data Lifecycle Manager에서 시도는 실패합니다. 제한 시간은 사전 스크립트와 사후 스크립트에 개별적으로 적용됩니다. 최소 및 기본 제한 시간은 10초입니다. 최대 제한 시간은 120초입니다.
- 실패한 스크립트 재시도 - 제한 시간 내에 완료되지 않은 스크립트를 재시도하려면 이 옵션을 선택합니다. 사전 스크립트가 실패할 경우 Amazon Data Lifecycle Manager는 사전 및 사후 스크립트 실행을 포함하여 전체 스냅샷 생성 프로세스를 재시도합니다. 사후 스크립트가 실패할 경우 Amazon Data Lifecycle Manager는 사후 스크립트만 재시도합니다. 이 경우 사전 스크립트가 완료되고 스냅샷이 생성되었을 수 있습니다.
- 중단 일관성 스냅샷으로 기본 설정 - 사전 스크립트 실행에 실패할 경우 중단 일관성 스냅샷을 기본값으로 설정하려면 이 옵션을 선택합니다. 이는 사전 및 사후 스크립트가 활성화되지 않은 경우 Amazon Data Lifecycle Manager의 기본 스냅샷 생성 동작입니다. 재시도를 활성화한 경우 Amazon Data Lifecycle Manager는 모든 재시도가 소진된 후에만 중단 일관성 스냅샷으로 기본 설정됩니다. 사전 스크립트가 실패하고 중단 일관성 스냅샷을 기본으로 설정하지 않으면 Amazon Data Lifecycle Manager는 해당 일정 실행 중에 인스턴스에 대한 스냅샷을 생성하지 않습니다.

> **Note**
>
> SAP HANA용 스냅샷을 생성하는 경우 이 옵션을 비활성화하는 것이 좋습니다. SAP HANA 워크로드의 중단 일관성 스냅샷은 동일한 방식으로 복원할 수 없습니다.

9. 기본 정책 생성을 선택합니다.

> **Note**
>
> `Role with name AWSDataLifecycleManagerDefaultRole already exists` 오류가 발생하는 경우 자세한 내용은 Amazon Data Lifecycle Manager 문제 해결 섹션을 참조하세요.

AWS CLI

스냅샷 수명 주기 정책 생성

create-lifecycle-policy 명령을 사용하고 `CreateRule`에 `Scripts` 파라미터를 포함시킵니다. 파라미터에 대한 자세한 내용은 Amazon Data Lifecycle Manager API 참조를 확인하세요.

```
$ aws dlm create-lifecycle-policy \
--description "policy_description" \
--state ENABLED \
--execution-role-arn iam_role_arn \
--policy-details file://policyDetails.json
```

사용 사례에 따라 `policyDetails.json`에 다음 중 하나가 포함됩니다.

- VSS 백업

```
{
    "PolicyType": "EBS_SNAPSHOT_MANAGEMENT",
    "ResourceTypes": [
        "INSTANCE"
    ],
    "TargetTags": [{
        "Key": "tag_key",
        "Value": "tag_value"
    }],
    "Schedules": [{
        "Name": "schedule_name",
        "CreateRule": {
            "CronExpression": "cron_for_creation_frequency",
            "Scripts": [{
                "ExecutionHandler":"AWS_VSS_BACKUP",
                "ExecuteOperationOnScriptFailure":true|false,
                "MaximumRetryCount":retries (0-3)
```

```
            }]
        },
        "RetainRule": {
            "Count": retention_count
        }
    }]
}
```

- SAP HANA 백업

```
{
    "PolicyType": "EBS_SNAPSHOT_MANAGEMENT",
    "ResourceTypes": [
        "INSTANCE"
    ],
    "TargetTags": [{
        "Key": "tag_key",
        "Value": "tag_value"
    }],
    "Schedules": [{
        "Name": "schedule_name",
        "CreateRule": {
            "CronExpression": "cron_for_creation_frequency",
            "Scripts": [{
                "Stages": ["PRE","POST"],
                "ExecutionHandlerService":"AWS_SYSTEMS_MANAGER",
                "ExecutionHandler":"AWSSystemsManagerSAP-
CreateDLMSnapshotForSAPHANA",
                "ExecuteOperationOnScriptFailure":true|false,
                "ExecutionTimeout":timeout_in_seconds (10-120),
                "MaximumRetryCount":retries (0-3)
            }]
        },
        "RetainRule": {
            "Count": retention_count
        }
    }]
}
```

- 사용자 지정 SSM 문서

```
{
    "PolicyType": "EBS_SNAPSHOT_MANAGEMENT",
```

```
    "ResourceTypes": [
        "INSTANCE"
    ],
    "TargetTags": [{
        "Key": "tag_key",
        "Value": "tag_value"
    }],
    "Schedules": [{
        "Name": "schedule_name",
        "CreateRule": {
            "CronExpression": "cron_for_creation_frequency",
            "Scripts": [{
                "Stages": ["PRE","POST"],
                "ExecutionHandlerService":"AWS_SYSTEMS_MANAGER",
                "ExecutionHandler":"ssm_document_name|arn",
                "ExecuteOperationOnScriptFailure":true|false,
                "ExecutionTimeout":timeout_in_seconds (10-120),
                "MaximumRetryCount":retries (0-3)
            }]
        },
        "RetainRule": {
            "Count": retention_count
        }
    }]
}
```

Amazon Data Lifecycle Manager를 사용한 VSS 백업 고려 사항

Amazon Data Lifecycle Manager를 사용하면 Amazon EC2 인스턴스에서 실행되는 VSS(Volume Shadow Copy Service) 지원 Windows 애플리케이션을 백업 및 복원할 수 있습니다. 애플리케이션에 Windows VSS에 등록된 VSS 작성기가 있는 경우 Amazon Data Lifecycle Manager는 해당 애플리케이션에 일관되게 적용되는 스냅샷을 생성합니다.

> **Note**
>
> Amazon Data Lifecycle Manager는 현재 Amazon EC2에서 실행되는 리소스의 애플리케이션에 일관되게 적용되는 스냅샷만 지원합니다. 특히 기존 인스턴스를 백업에서 생성된 새 인스턴스로 교체하여 애플리케이션 데이터를 복원할 수 있는 백업 시나리오에 적합합니다. 모든 인스턴스 유형 또는 애플리케이션이 VSS 백업에 대해 지원되는 것은 아닙니다. 자세한 내용은

Amazon EC2 사용 설명서의 애플리케이션에 일관되게 적용되는 Windows 스냅샷을 참조하세요.

지원되지 않는 인스턴스 유형

다음 Amazon EC2 인스턴스 유형은 VSS 백업에 지원되지 않습니다. 정책이 이러한 인스턴스 유형 중 하나를 대상으로 하는 경우 Amazon Data Lifecycle Manager는 여전히 VSS 백업을 생성할 수 있지만 스냅샷에 필수 시스템 태그가 지정되지 않을 수 있습니다. 이러한 태그가 없으면 스냅샷은 생성 후 Amazon Data Lifecycle Manager에 의해 관리되지 않습니다. 이러한 스냅샷을 수동으로 삭제해야 할 수도 있습니다.

- T3: `t3.nano` | `t3.micro`
- T3a: `t3a.nano` | `t3a.micro`
- T2: `t2.nano` | `t2.micro`

애플리케이션에 일관되게 적용되는 스냅샷에 대한 공동 책임

다음을 확인해야 합니다.

- 대상 인스턴스에서 SSM Agent가 설치되어 있고 최신 상태로 실행되고 있습니다.
- Systems Manager는 대상 인스턴스에 대해 필요한 작업을 수행할 수 있는 권한이 있습니다.
- Amazon Data Lifecycle Manager는 대상 인스턴스에서 사전 및 사후 스크립트를 실행하는 데 필요한 Systems Manager 작업을 수행할 권한이 있습니다.
- 자체 관리형 MySQL, PostgreSQL 또는 InterSystems IRIS 데이터베이스와 같은 사용자 지정 워크로드의 경우 사용하는 SSM 문서에는 데이터베이스 구성의 I/O를 동결, 플러시 및 재개하는 데 필요한 올바른 조치가 포함되어 있습니다.
- 스냅샷 생성 시간은 워크로드 일정에 따라 조정됩니다. 예를 들어, 예약된 유지 관리 기간 동안 스냅샷 생성을 예약해 보세요.

Amazon Data Lifecycle Manager는 다음을 보장합니다.

- 예약된 스냅샷 생성 시간으로부터 60분 내에 스냅샷 생성이 시작됩니다.
- 스냅샷 생성이 시작되기 전에 사전 스크립트가 실행됩니다.

- 사전 스크립트가 성공하고 스냅샷 생성이 시작된 후 사후 스크립트가 실행됩니다. Amazon Data Lifecycle Manager는 사전 스크립트가 성공한 경우에만 사후 스크립트를 실행합니다. 사전 스크립트가 실패하는 경우 Amazon Data Lifecycle Manager는 사후 스크립트를 실행하지 않습니다.
- 생성 시 스냅샷에 적절한 태그가 지정됩니다.
- CloudWatch 지표 및 이벤트는 스크립트가 시작되거나 실패 또는 성공할 때 내보내지거나 발생합니다.

Data Lifecycle Manager 사전 및 사후 스크립트의 기타 사용 사례

사전 및 사후 스크립트를 사용하여 애플리케이션에 일관되게 적용되는 스냅샷을 자동화하는 것 외에도 사전 및 사후 스크립트를 함께 사용하거나 개별적으로 사용하여 스냅샷 생성 전후에 다른 관리 작업을 자동화할 수 있습니다. 예제:

- 스냅샷을 생성하기 전에 사전 스크립트를 사용하여 패치를 적용합니다. 이렇게 하면 정기 주간 또는 월간 소프트웨어 업데이트를 적용한 후 스냅샷을 생성할 수 있습니다.

  > **Note**
  >
  > 사전 스크립트만 실행하도록 선택하면 중단 일관성 스냅샷으로 기본 설정이 활성화됩니다.

- 스냅샷을 생성한 후 사후 스크립트를 사용하여 패치를 적용합니다. 이렇게 하면 정기 주간 또는 월간 소프트웨어 업데이트를 적용하기 전 스냅샷을 생성할 수 있습니다.

다른 사용 사례를 위한 시작하기

이 섹션에서는 애플리케이션에 일관되게 적용되는 스냅샷 이외의 사용 사례에 사전 및/또는 사후 스크립트를 사용할 때 수행해야 하는 단계를 설명합니다.

1단계: 대상 인스턴스 준비

사전 및/또는 사후 스크립트를 위한 대상 인스턴스 준비

1. SSM Agent가 아직 설치되지 않은 경우 대상 인스턴스에 SSM Agent를 설치합니다. SSM Agent가 대상 인스턴스에 이미 설치되어 있는 경우 이 단계를 건너뜁니다.
   - (Linux 인스턴스) Linux용 EC2 인스턴스에 수동으로 SSM Agent 설치
   - (Windows 인스턴스) Windows Server용 EC2 인스턴스에서 SSM 에이전트 작업

2. SSM 에이전트가 실행 중인지 확인합니다. 자세한 내용은 SSM Agent 상태 확인 및 에이전트 시작을 참조하세요.
3. Amazon EC2 인스턴스용 Systems Manager를 설정합니다. 자세한 내용은 AWS Systems Manager 사용 설명서의 Amazon EC2 인스턴스용 Systems Manager 설정을 참조하세요.

2단계: SSM 문서 준비

실행하려는 명령과 함께 사전 및/또는 사후 스크립트를 포함하는 SSM 명령 문서를 생성해야 합니다.

아래의 빈 SSM 문서 템플릿을 사용하여 SSM 문서를 생성하고 해당 문서 섹션에 사전 및 사후 스크립트 명령을 추가할 수 있습니다.

⚠ 다음 사항에 유의하세요.

- SSM 문서가 워크로드에 필요한 올바른 작업을 수행하는지 확인하는 것은 사용자의 책임입니다.
- SSM 문서에는 `pre-script`, `post-script` 및 `dry-run`을 포함하여 `allowedValues`에 대한 필수 필드가 포함되어야 합니다. Amazon Data Lifecycle Manager는 이러한 섹션의 내용을 기반으로 인스턴스에서 명령을 실행합니다. SSM 문서에 이러한 섹션이 없는 경우 Amazon Data Lifecycle Manager는 해당 문서를 실패한 실행으로 간주합니다.

```
###============================================================================###
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.

# Permission is hereby granted, free of charge, to any person obtaining a copy of this
# software and associated documentation files (the "Software"), to deal in the Software
# without restriction, including without limitation the rights to use, copy, modify,
# merge, publish, distribute, sublicense, and/or sell copies of the Software, and to
# permit persons to whom the Software is furnished to do so.

# THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED,
# INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
# PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT
# HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
# OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
# SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
###============================================================================###
schemaVersion: '2.2'
```

```
description: SSM Document Template for Amazon Data Lifecycle Manager Pre/Post script
 feature
parameters:
  executionId:
    type: String
    default: None
    description: (Required) Specifies the unique identifier associated with a pre and/
or post execution
    allowedPattern: ^(None|[a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-
[a-fA-F0-9]{12})$
  command:
  # Data Lifecycle Manager will trigger the pre-script and post-script actions during
 policy execution.
  # 'dry-run' option is intended for validating the document execution without
 triggering any commands
  # on the instance. The following allowedValues will allow Data Lifecycle Manager to
 successfully
  # trigger pre and post script actions.
    type: String
    default: 'dry-run'
    description: (Required) Specifies whether pre-script and/or post-script should be
 executed.
    allowedValues:
    - pre-script
    - post-script
    - dry-run

mainSteps:
- action: aws:runShellScript
  description: Run Database freeze/thaw commands
  name: run_pre_post_scripts
  precondition:
    StringEquals:
    - platformType
    - Linux
  inputs:
    runCommand:
    - |
      #!/bin/bash


 ###============================================================================###
      ### Error Codes
```

```
###============================================================================###
     # The following Error codes will inform Data Lifecycle Manager of the type of
error
     # and help guide handling of the error.
     # The Error code will also be emitted via AWS Eventbridge events in the 'cause'
field.
     # 1 Pre-script failed during execution - 201
     # 2 Post-script failed during execution - 202
     # 3 Auto thaw occurred before post-script was initiated - 203
     # 4 Pre-script initiated while post-script was expected - 204
     # 5 Post-script initiated while pre-script was expected - 205
     # 6 Application not ready for pre or post-script initiation - 206


###============================================================================###
     ### Global variables

###============================================================================###
     START=$(date +%s)
     # For testing this script locally, replace the below with OPERATION=$1.
     OPERATION={{ command }}

     # Add all pre-script actions to be performed within the function below
     execute_pre_script() {
         echo "INFO: Start execution of pre-script"
     }

     # Add all post-script actions to be performed within the function below
     execute_post_script() {
         echo "INFO: Start execution of post-script"
     }

     # Debug logging for parameters passed to the SSM document
     echo "INFO: ${OPERATION} starting at $(date) with executionId: ${EXECUTION_ID}"

     # Based on the command parameter value execute the function that supports
     # pre-script/post-script operation
     case ${OPERATION} in
         pre-script)
             execute_pre_script
             ;;
         post-script)
             execute_post_script
```

```
            ;;
        dry-run)
            echo "INFO: dry-run option invoked - taking no action"
            ;;
        *)
            echo "ERROR: Invalid command parameter passed. Please use either pre-
script, post-script, dry-run."
            exit 1 # return failure
            ;;
    esac

    END=$(date +%s)
    # Debug Log for profiling the script time
    echo "INFO: ${OPERATION} completed at $(date). Total runtime: $((${END} -
 ${START})) seconds."
```

3단계: Amazon Data Lifecycle Manager IAM 역할 준비

> **Note**
>
> 이 단계는 다음과 같은 경우 필요합니다.
>
> - 사용자 지정 IAM 역할을 사용하는 사전/사후 스크립트 지원 스냅샷 정책을 생성하거나 업데이트합니다.
> - 명령줄을 사용하여 기본값을 사용하는 사전/사후 스크립트 지원 스냅샷 정책을 생성하거나 업데이트합니다.
>
> 콘솔을 사용하여 스냅샷 관리를 위한 기본 역할(AWSDataLifecycleManagerDefaultRole)을 사용하는 사전/사후 스크립트 지원 스냅샷 정책을 생성하거나 업데이트하려면 이 단계를 건너뜁니다. 이 경우 AWSDataLifecycleManagerSSMFullAccess 정책이 해당 역할에 자동으로 연결됩니다.

정책에 사용하는 IAM 역할이 정책의 대상이 되는 인스턴스에서 사전 및 사후 스크립트를 실행하는 데 필요한 SSM 작업을 수행할 수 있는 권한을 Amazon Data Lifecycle Manager에 부여하는지 확인해야 합니다.

Amazon Data Lifecycle Manager는 필요한 권한이 포함된 관리형 정책(AWSDataLifecycleManagerSSMFullAccess)을 제공합니다. 스냅샷을 관리하기 위해이 정책을 IAM 역할에 연결하여 권한이 포함되어 있는지 확인할 수 있습니다.

> **Important**
>
> AWSDataLifecycleManagerSSMFullAccess 관리형 정책은 사전 및 사후 스크립트를 사용할 때 `aws:ResourceTag` 조건 키를 사용하여 특정 SSM 문서에 대한 액세스를 제한합니다. Amazon Data Lifecycle Manager가 SSM 문서에 액세스하도록 허용하려면 SSM 문서에 로 태그가 지정되어 있는지 확인해야 합니다`DLMScriptsAccess:true`.

또는 사용자 지정 정책을 수동으로 생성하거나 사용하는 IAM 역할에 필요한 권한을 직접 할당할 수 있습니다. AWSDataLifecycleManagerSSMFullAccess 관리형 정책에 정의된 동일한 권한을 사용할 수 있지만 `aws:ResourceTag` 조건 키는 선택 사항입니다. 해당 조건 키를 사용하지 않기로 결정하면 SSM 문서에 `DLMScriptsAccess:true`로 태그를 지정할 필요가 없습니다.

다음 방법 중 하나를 사용하여 IAM 역할에 AWSDataLifecycleManagerSSMFullAccess 정책을 추가합니다.

Console

사용자 지정 역할에 관리형 정책 연결

1. IAM 콘솔(https://console.aws.amazon.com/iam/)을 엽니다.
2. 탐색 창에서 [역할(Roles)]을 선택합니다.
3. 스냅샷 관리를 위한 사용자 지정 역할을 검색하고 선택합니다.
4. 권한 탭에서 권한 추가, 정책 연결을 선택합니다.
5. AWSDataLifecycleManagerSSMFullAccess 관리형 정책을 검색하여 선택한 다음 권한 추가를 선택합니다.

AWS CLI

사용자 지정 역할에 관리형 정책 연결

attach-role-policy 명령을 사용합니다. `---role-name`에 대해 사용자 지정 역할의 이름을 지정합니다. `--policy-arn`에서 `arn:aws:iam::aws:policy/AWSDataLifecycleManagerSSMFullAccess`를 지정합니다.

```
$ aws iam attach-role-policy \
--policy-arn arn:aws:iam::aws:policy/AWSDataLifecycleManagerSSMFullAccess \
--role-name your_role_name
```

스냅샷 수명 주기 정책 생성

Console

스냅샷 수명 주기 정책 생성

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Elastic Block Store, Lifecycle Manager 및 수명 주기 정책 생성을 차례로 선택합니다.
3. 정책 유형 선택(Select policy type) 화면에서 EBS 스냅샷 정책(EBS snapshot policy)을 선택한 후 다음(Next)을 선택합니다.
4. 대상 리소스(Target resources) 섹션에서 다음을 수행합니다.
   a. 대상 리소스 유형에서는 `Instance`를 선택합니다.
   b. 대상 리소스 태그에서 백업할 인스턴스를 식별하는 리소스 태그를 지정합니다. 지정된 태그가 있는 리소스만 백업됩니다.
5. IAM 역할에서 AWSDataLifecycleManagerDefaultRole(스냅샷 관리를 위한 기본 역할)을 선택하거나 사전 및 사후 스크립트용으로 생성하고 준비한 사용자 지정 역할을 선택합니다.
6. 필요에 따라 일정과 추가 옵션을 구성합니다. 유지 관리 기간과 같이 워크로드에 맞는 기간으로 스냅샷 생성 시간을 예약하는 것이 좋습니다.
7. 사전 및 사후 스크립트 섹션에서 사전 및 사후 스크립트 활성화를 선택하고 다음을 수행합니다.
   a. 사용자 지정 SSM 문서를 선택합니다.
   b. 자동화 옵션에서 실행하려는 스크립트와 일치하는 옵션을 선택합니다.
   c. SSM 문서에서 준비한 SSM 문서를 선택합니다.
8. 필요한 경우 다음과 같은 추가 옵션을 구성합니다.
   - 스크립트 제한 시간 - Amazon Data Lifecycle Manager에서 완료되지 않은 스크립트 실행 시도가 실패하기 전까지의 제한 시간입니다. 스크립트가 제한 시간 내에 완료되지 않으면 Amazon Data Lifecycle Manager에서 시도는 실패합니다. 제한 시간은 사전 스크립트와 사후 스크립트에 개별적으로 적용됩니다. 최소 및 기본 제한 시간은 10초입니다. 최대 제한 시간은 120초입니다.
   - 실패한 스크립트 재시도 - 제한 시간 내에 완료되지 않은 스크립트를 재시도하려면 이 옵션을 선택합니다. 사전 스크립트가 실패할 경우 Amazon Data Lifecycle Manager는 사전 및 사후 스크립트 실행을 포함하여 전체 스냅샷 생성 프로세스를 재시도합니다. 사후 스크립트가

실패할 경우 Amazon Data Lifecycle Manager는 사후 스크립트만 재시도합니다. 이 경우 사전 스크립트가 완료되고 스냅샷이 생성되었을 수 있습니다.

- 중단 일관성 스냅샷으로 기본 설정 - 사전 스크립트 실행에 실패할 경우 중단 일관성 스냅샷을 기본값으로 설정하려면 이 옵션을 선택합니다. 이는 사전 및 사후 스크립트가 활성화되지 않은 경우 Amazon Data Lifecycle Manager의 기본 스냅샷 생성 동작입니다. 재시도를 활성화한 경우 Amazon Data Lifecycle Manager는 모든 재시도가 소진된 후에만 중단 일관성 스냅샷으로 기본 설정됩니다. 사전 스크립트가 실패하고 중단 일관성 스냅샷을 기본으로 설정하지 않으면 Amazon Data Lifecycle Manager는 해당 일정 실행 중에 인스턴스에 대한 스냅샷을 생성하지 않습니다.

9. 기본 정책 생성을 선택합니다.

> **Note**
>
> `Role with name AWSDataLifecycleManagerDefaultRole already exists` 오류가 발생하는 경우 자세한 내용은 Amazon Data Lifecycle Manager 문제 해결 섹션을 참조하세요.

AWS CLI

스냅샷 수명 주기 정책 생성

create-lifecycle-policy 명령을 사용하고 `CreateRule`에 `Scripts` 파라미터를 포함시킵니다. 파라미터에 대한 자세한 내용은 Amazon Data Lifecycle Manager API 참조를 확인하세요.

```
$ aws dlm create-lifecycle-policy \
--description "policy_description" \
--state ENABLED \
--execution-role-arn iam_role_arn \
--policy-details file://policyDetails.json
```

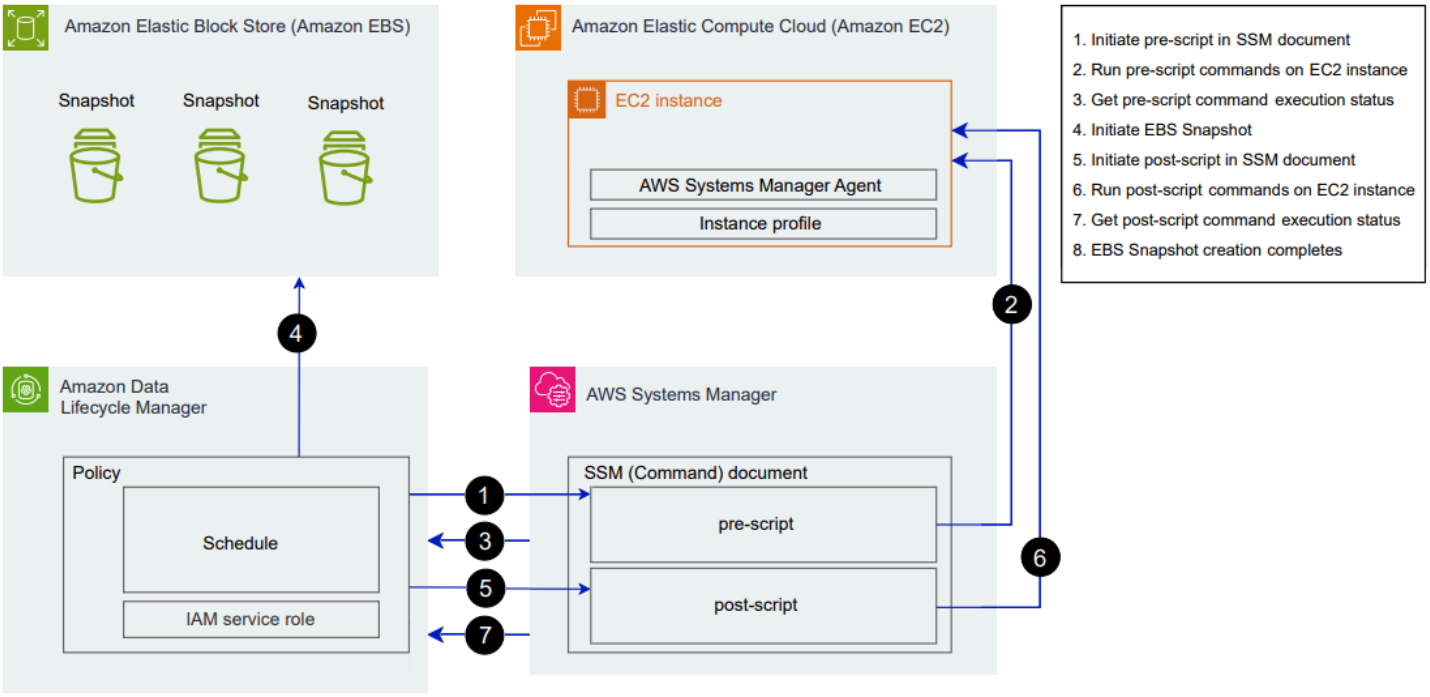
여기에서 `policyDetails.json`은 다음을 포함합니다.

```
{
    "PolicyType": "EBS_SNAPSHOT_MANAGEMENT",
    "ResourceTypes": [
        "INSTANCE"
    ],
    "TargetTags": [{
```

```
        "Key": "tag_key",
        "Value": "tag_value"
    }],
    "Schedules": [{
        "Name": "schedule_name",
        "CreateRule": {
            "CronExpression": "cron_for_creation_frequency",
            "Scripts": [{
                "Stages": ["PRE" | "POST" | "PRE","POST"],
                "ExecutionHandlerService":"AWS_SYSTEMS_MANAGER",
                "ExecutionHandler":"ssm_document_name|arn",
                "ExecuteOperationOnScriptFailure":true|false,
                "ExecutionTimeout":timeout_in_seconds (10-120),
                "MaximumRetryCount":retries (0-3)
            }]
        },
        "RetainRule": {
            "Count": retention_count
        }
    }]
}
```

Amazon Data Lifecycle Manager 사전 및 사후 스크립트 작동 방식

다음 이미지는 사용자 지정 SSM 문서를 사용할 때의 사전 및 사후 스크립트의 프로세스 흐름을 보여 줍니다. 이것이 VSS 백업에 적용되지는 않습니다.



예약된 스냅샷 생성 시간에 다음과 같은 작업과 교차 서비스 상호 작용이 발생합니다.

1. Amazon Data Lifecycle Manager는 SSM 문서를 호출하고 `pre-script` 파라미터를 전달하여 사전 스크립트 작업을 시작합니다.

   > **Note**
   >
   > 1~3단계는 사전 스크립트를 실행하는 경우에만 발생합니다. 사후 스크립트만 실행하는 경우 1~3단계는 생략됩니다.

2. Systems Manager는 대상 인스턴스에서 실행 중인 SSM Agent에 사전 스크립트 명령을 전송합니다. SSM Agent는 인스턴스에서 명령을 실행하고 상태 정보를 Systems Manager로 다시 전송합니다.

   예를 들어 SSM 문서를 사용하여 애플리케이션 일치 스냅샷을 생성하는 경우 사전 스크립트는 스냅샷을 생성하기 전에 버퍼링된 모든 데이터가 볼륨에 기록되도록 I/O를 고정하고 플러시할 수 있습니다.

3. Systems Manager는 사전 스크립트 명령 상태 업데이트를 Amazon Data Lifecycle Manager로 전송합니다. 사전 스크립트가 실패하면 Amazon Data Lifecycle Manager는 사전 및 사후 스크립트 옵션을 구성하는 방법에 따라 다음 작업 중 하나를 수행합니다.

재시도	중단 일관성 스냅샷으로 기본 설정	작업
재시도 횟수가 남아 있는 상태에서 활성화됨	활성화됨	성공하거나 재시도 횟수가 모두 소진될 때까지 스크립트 재시도
성공적으로 완료하지 못한 상태에서 소진됨	활성화됨	중단 일관성 스냅샷을 생성하고 사후 스크립트를 실행하지 않습니다.
재시도 횟수가 남아 있는 상태에서 활성화됨	비활성화됨	성공하거나 재시도 횟수가 모두 소진될 때까지 스크립트 재시도
성공적으로 완료하지 못한 상태에서 소진됨	비활성화됨	대상 인스턴스에 대한 스냅샷 생성을 건너뛰고 사후 스크립트를 실행하지 않습니다.
비활성	활성화됨	중단 일관성 스냅샷을 생성하고 사후 스크립트를 실행하지 않습니다.
비활성화됨	비활성화됨	대상 인스턴스에 대한 스냅샷 생성을 건너뛰고 사후 스크립트를 실행하지 않습니다.

4. Amazon Data Lifecycle Manager가 스냅샷 생성을 시작합니다.
5. Amazon Data Lifecycle Manager는 SSM 문서를 호출하고 `post-script` 파라미터를 전달하여 사후 스크립트 작업을 시작합니다.

> **Note**
>
> 5~7단계는 사전 스크립트를 실행하는 경우에만 발생합니다. 사후 스크립트만 실행하는 경우 1~3단계는 생략됩니다.

6. Systems Manager는 대상 인스턴스에서 실행 중인 SSM Agent에 사후 스크립트 명령을 전송합니다. SSM Agent는 인스턴스에서 명령을 실행하고 상태 정보를 Systems Manager로 다시 전송합니다.

예를 들어 SSM 문서가 애플리케이션에 일관되게 적용되는 스냅샷을 활성화하는 경우이 사후 스크립트는 스냅샷을 생성한 후 데이터베이스가 정상적인 I/O 작업을 재개하도록 I/O를 중단할 수 있습니다.

7. 사후 스크립트를 실행하고 Systems Manager가 성공적으로 완료되었다고 표시하면 프로세스가 완료됩니다.

   사후 스크립트가 실패하면 Amazon Data Lifecycle Manager는 사전 및 사후 스크립트 옵션을 구성하는 방법에 따라 다음 작업 중 하나를 수행합니다.

재시도	작업
재시도 횟수가 남아 있는 상태에서 활성화됨	성공하거나 재시도 횟수가 모두 소진될 때까지 사후 스크립트 재시도
성공 없이 소진됨	사후 스크립트 건너뛰기
비활성화됨	사후 스크립트 건너뛰기

   단, 사후 스크립트가 실패하면 사전 스크립트(활성화된 경우)가 성공적으로 완료되어 스냅샷이 생성되었을 수 있다는 점에 유의하세요. 인스턴스가 예상대로 작동하는지 확인하기 위해 인스턴스에 대해 추가 조치를 취해야 할 수 있습니다. 예를 들어, 사전 스크립트가 I/O를 일시 중지하고 플러시했지만 사후 스크립트가 I/O를 재개하지 못한 경우 I/O를 자동 재개하도록 데이터베이스를 구성하거나 I/O를 수동으로 재개해야 할 수 있습니다.

8. 사후 스크립트가 완료된 후 스냅샷 생성 프로세스가 완료될 수 있습니다. 스냅샷을 완료하는 데 걸리는 시간은 스냅샷 크기에 따라 다릅니다.

Data Lifecycle Manager 사전 및 사후 스크립트로 생성된 스냅샷 식별

Amazon Data Lifecycle Manager는 사전 및 사후 스크립트로 생성된 스냅샷에 다음과 같은 시스템 태그를 자동으로 할당합니다.

- 키: `aws:dlm:pre-script`, 값: `SUCCESS|FAILED`

  태그 값이 `SUCCESS`이면 사전 스크립트가 성공적으로 실행된 것입니다. 태그 값이 `FAILED`이면 사전 스크립트가 성공적으로 실행되지 않은 것입니다.

- 키: `aws:dlm:post-script`, 값: `SUCCESS|FAILED`

태그 값이 SUCCESS이면 사후 스크립트가 성공적으로 실행된 것입니다. 태그 값이 FAILED이면 사후 스크립트가 성공적으로 실행되지 않은 것입니다.

사용자 지정 SSM 문서 및 SAP HANA 백업의 경우 스냅샷에 aws:dlm:pre-script:SUCCESS와 aws:dlm:post-script:SUCCESS 태그가 모두 지정된 경우 애플리케이션에 일관되게 적용되는 스냅샷이 성공적으로 생성되었는지 유추할 수 있습니다.

또한 VSS 백업을 사용하여 생성된 애플리케이션에 일관되게 적용되는 스냅샷에는 다음 태그가 자동으로 지정됩니다.

- 키: AppConsistent tag, 값: true|false

  태그 값이 true이면 VSS 백업이 성공했고 스냅샷이 애플리케이션에 일관되게 적용되는 것입니다. 태그 값이 false이면 VSS 백업이 실패했고 스냅샷이 애플리케이션에 일관되게 적용되지 않는 것입니다.

Amazon Data Lifecycle Manager 사전 및 사후 스크립트 모니터링

Amazon CloudWatch 지표

Amazon Data Lifecycle Manager는 사전/사후 스크립트가 실패하고 성공할 때와 VSS 백업이 실패하고 성공할 때 다음과 같은 CloudWatch 지표를 게시합니다.

- PreScriptStarted
- PreScriptCompleted
- PreScriptFailed
- PostScriptStarted
- PostScriptCompleted
- PostScriptFailed
- VSSBackupStarted
- VSSBackupCompleted
- VSSBackupFailed

자세한 내용은 CloudWatch를 사용하여 Data Lifecycle Manager 정책 모니터링 단원을 참조하십시오.

Amazon EventBridge

Amazon Data Lifecycle Manager는 사전 또는 사후 스크립트가 시작, 성공 또는 실패할 때 다음과 같은 Amazon EventBridge 이벤트를 발생시킵니다.

- `DLM Pre Post Script Notification`

자세한 내용은 EventBridge를 사용하여 Data Lifecycle Manager 정책 모니터링 단원을 참조하십시오.

EBS 지원 AMI에 대한 Amazon Data Lifecycle Manager 사용자 지정 정책 생성

다음 절차에서는 Amazon Data Lifecycle Manager를 사용하여 EBS 지원 AMI 수명 주기를 자동화하는 방법을 보여줍니다.

주제

- AMI 수명 주기 정책 생성
- AMI 수명 주기 정책 고려 사항
- 추가 리소스

AMI 수명 주기 정책 생성

다음 절차 중 하나를 사용하여 AMI 수명 주기 정책을 생성합니다.

Console

AMI 정책을 생성하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [Elastic Block Store], [Lifecycle Manager] 및 [수명 주기 정책 생성]을 차례로 선택합니다.
3. [정책 유형 선택(Select policy type)] 화면에서 [EBS 지원 AMI 정책(EBS-backed AMI policy)]을 선택한 후 [다음(Next)]을 선택합니다.
4. [대상 리소스(Target resources)] 섹션의 [대상 리소스 태그(Target resource tags)]에서 백업할 볼륨 또는 인스턴스를 식별하는 리소스 태그를 선택합니다. 지정한 태그 키 및 값 페어가 있는 리소스만 정책에 의해 백업됩니다.
5. [설명(Description)]에 정책에 대한 간략한 설명을 입력합니다.

6. [IAM 역할(IAM role)]에서 AMI 및 스냅샷을 관리하고 인스턴스를 설명할 권한이 있는 IAM 역할을 선택합니다. Amazon Data Lifecycle Manager가 제공하는 기본 역할을 사용하려면 [기본 역할(Default role)]을 선택합니다. 또는 이전에 생성한 사용자 지정 IAM 역할을 사용하려면 [다른 역할 선택(Choose another role)]을 선택하고 사용할 역할을 선택합니다.
7. [정책 태그(Policy tags)]에서 수명 주기 정책에 적용할 태그를 추가합니다. 이 태그를 사용하여 정책을 식별 및 분류할 수 있습니다.
8. [생성 후 정책 상태(Policy status after creation)]에서 [정책 활성화(Enable policy)]를 선택하여 다음 예약 시간에 정책 실행을 시작하거나, [정책 비활성화(Disable policy)]를 선택하여 정책을 실행하지 않습니다. 지금 정책을 활성화하지 않으면 생성 후 수동으로 활성화할 때까지 AMI 생성이 시작되지 않습니다.
9. [인스턴스 재부팅(Instance reboot)] 섹션에서 AMI 생성 전에 인스턴스를 재부팅해야 할지 여부를 지정합니다. 대상 인스턴스가 재부팅되지 않도록 하려면 [아니요(No)]를 선택합니다. [아니요(No)]를 선택하면 데이터 일관성 문제가 발생할 수 있습니다. AMI 생성 전에 인스턴스를 재부팅하려면 [예(Yes)]를 선택합니다. 이 옵션을 선택하면 데이터 일관성이 보장되지만 여러 대상 인스턴스가 동시에 재부팅될 수 있습니다.
10. [다음(Next)]을 선택합니다.
11. [일정 구성(Configure schedule)] 화면에서 정책 일정을 구성합니다. 정책에는 최대 4개의 일정을 구성할 수 있습니다. 일정 1은 필수입니다. 일정 2, 3, 4는 선택 사항입니다. 추가한 각 정책 일정에 대해 다음을 수행합니다.

    a. 일정 세부 정보(Schedule details) 섹션에서 다음을 수행합니다.

        i. 일정 이름(Schedule name)에 일정을 설명하는 이름을 지정합니다.

        ii. 빈도(Frequency) 및 관련 필드에서 정책 실행 간격을 구성합니다.

            매일, 매주, 매월 또는 매년 일정에 따라 정책 실행을 구성할 수 있습니다. 또는 사용자 지정 cron 표현식(Custom cron expression)을 선택하여 최대 1년의 간격을 지정합니다. 자세한 내용은 Amazon EventBridge 사용 설명서의 Cron 및 rate 표현식을 참조하세요.

        iii. [시작 시간(Starting at)]에서 정책 실행을 시작할 시간을 지정합니다. 첫 번째 정책 실행은 예약하는 시간 후 한 시간 이내에 시작됩니다. 시간은 hh:mm UTC 형식으로 입력해야 합니다.

        iv. [보존 유형(Retention type)]에서 일정에 따라 생성되는 AMI의 보존 정책을 지정합니다.

            총 수 또는 수명을 기준으로 AMI를 보존할 수 있습니다.

개수 기반 보존의 경우 범위는 1에서 1000까지입니다. 최대 수에 도달한 후에는 새 AMI를 생성할 때 가장 오래된 AMI가 등록 해제됩니다.

수명을 기준으로 보존하는 경우 범위는 1일에서 100년까지입니다. 각 AMI의 보존 기간이 만료되면 AMI가 등록 해제됩니다.

> **Note**
>
> 모든 일정은 동일한 보존 유형을 가져야 합니다. 일정 1에 대해서만 보존 유형을 지정할 수 있습니다. 일정 2, 3, 4는 일정 1에서 보존 유형을 상속합니다. 각 일정에는 고유한 보존 횟수 또는 기간이 있을 수 있습니다.

b. AMI에 대한 태깅을 구성합니다.

[태깅(Tagging)] 섹션에서 다음을 수행합니다.

i. 일정에 따라 소스 인스턴스에서 사용자 정의 태그를 모두 AMI로 복사하려면 [소스에서 태그 복사(Copy tags from source)]를 선택합니다.

ii. 기본적으로 일정에 따라 생성되는 AMI에는 소스 인스턴스의 ID가 자동으로 태깅됩니다. 이 자동 태깅이 발생하지 않도록 하려면 [가변 태그(Variable tags)]를 선택한 다음 `instance-id:$(instance-id)` 타일을 선택합니다.

iii. 이 일정에 따라 생성된 AMI에 할당할 추가 태그를 지정하려면 [태그 추가(Add tags)]를 선택합니다.

c. AMI 사용 중단을 구성합니다.

더 이상 사용하지 않아야 할 AMI를 사용 중단하려면 AMI 사용 중단(AMI deprecation) 섹션에서 이 일정에 대해 AMI 사용 중단 사용(Enable AMI deprecation for this schedule)을 선택한 다음 AMI 사용 중단 규칙을 지정합니다. AMI 사용 중단 규칙은 AMI가 사용 중단되는 시점을 지정합니다.

일정에 개수 기준 AMI 보존을 사용하는 경우 사용 중단할 가장 오래된 AMI 수를 지정해야 합니다. 사용 중단 수는 일정의 AMI 보존 수보다 작거나 같아야 하며 1,000보다 클 수 없습니다. 예를 들어 최대 5개의 AMI를 보존하도록 일정이 구성된 경우 가장 오래된 AMI를 최대 5개까지 사용 중단하도록 일정을 구성할 수 있습니다.

일정에 경과 시간 기준 AMI 보존을 사용하는 경우 AMI가 사용 중단되기까지의 기간을 지정해야 합니다. 사용 중단 수는 일정의 AMI 보존 수보다 작거나 같아야 하며 10년(120개월, 520주 또는 3,650일)보다 클 수 없습니다. 예를 들어 10일 동안 AMI를 보존하도록 일

정이 구성된 경우, 생성된 날로부터 최대 10일 후에 AMI를 사용 중단하도록 일정을 구성할 수 있습니다.

d. 크로스 리전 복사를 구성합니다.

일정에 따라 생성된 AMI를 다른 리전에 복사하려면 [리전 간 복사(Cross-Region copy)] 섹션에서 [리전 간 복사 활성화(Enable cross-Region copy)]를 선택합니다. 계정에서 최대 3개의 추가 리전에 AMI를 복사할 수 있습니다. 각 대상 리전에 대해 별도의 리전 간 복사 규칙을 지정해야 합니다.

각 대상 리전에 대해 다음을 지정할 수 있습니다.

- AMI 사본의 보존 정책. 이 보존 기간이 만료되면 대상 리전의 사본이 자동으로 등록 취소됩니다.
- AMI 사본의 암호화 상태. 소스 AMI가 암호화되거나 암호화가 기본적으로 활성화된 경우 복사된 AMI가 항상 암호화됩니다. 소스 AMI가 암호화되어 있지 않고 암호화가 기본적으로 비활성화된 경우 선택적으로 암호화를 사용할 수 있습니다. KMS 키를 지정하지 않은 경우 AMI는 각 대상 리전에서 EBS 암호화의 기본 KMS 키를 사용하여 암호화됩니다. 대상 리전에 대한 KMS 키를 지정하는 경우 선택한 IAM 역할에 KMS 키에 대한 액세스 권한이 있어야 합니다.
- AMI 사본의 사용 중단 규칙. 사용 중단 기간이 만료되면 AMI 사본은 자동으로 사용 중단됩니다. 사용 중단 기간은 사본 보존 기간보다 작거나 같아야 하며 10년보다 클 수 없습니다.
- 소스 AMI에서 모든 태그를 복사할지 아니면 태그를 복사하지 않을지 여부.

> **Note**
>
> 리전당 동시 AMI 복사본 수를 초과해서는 안 됩니다.

e. 일정을 더 추가하려면 화면 상단에서 [다른 일정 추가(Add another schedule)]를 선택합니다. 각 추가 일정에 대해, 이 주제의 앞부분에서 설명한 대로 필드를 작성합니다.

f. 필요한 일정을 추가한 후 [정책 검토(Review policy)]를 선택합니다.

12. 정책 요약을 검토한 다음 정책 생성(Create policy)을 선택합니다.

> **Note**
>
> `Role with name AWSDataLifecycleManagerDefaultRoleForAMIManagement already exists` 오류가 발생하는 경우 자세한 내용은 Amazon Data Lifecycle Manager 문제 해결 섹션을 참조하세요.

Command line

create-lifecycle-policy 명령을 사용하여 AMI 수명 주기 정책을 생성합니다. `PolicyType`에 `IMAGE_MANAGEMENT`을 지정합니다.

> **Note**
>
> 구문을 단순화하기 위해 다음 예에서는 정책 세부 정보가 포함된 JSON 파일(`policyDetails.json`)을 사용합니다.

예 1: 경과 시간 기준 보존 및 AMI 사용 중단

이 예제에서는 대상 인스턴스를 재부팅하지 않고 값이 `purpose`인 `production` 태그 키가 있는 모든 인스턴스의 AMI를 생성하는 AMI 수명 주기 정책을 만듭니다. 이 정책에는 매일 `01:00` UTC에 AMI를 생성하는 하나의 일정이 포함됩니다. 이 정책은 AMI를 2일 동안 보존하며 1일 후에 사용 중단합니다. 또한 소스 인스턴스의 태그를 생성하는 AMI에 복사합니다.

```
aws dlm create-lifecycle-policy \
    --description "My AMI policy" \
    --state ENABLED \
    --execution-role-arn
 arn:aws:iam::12345678910:role/AWSDataLifecycleManagerDefaultRoleForAMIManagement \
    --policy-details file://policyDetails.json
```

다음은 `policyDetails.json` 파일의 예입니다.

```
{
    "PolicyType": "IMAGE_MANAGEMENT",
    "ResourceTypes": [
```

```
        "INSTANCE"
    ],
    "TargetTags": [{
        "Key": "purpose",
        "Value": "production"
    }],
    "Schedules": [{
            "Name": "DailyAMIs",
            "TagsToAdd": [{
                "Key": "type",
                "Value": "myDailyAMI"
            }],
            "CreateRule": {
                "Interval": 24,
                "IntervalUnit": "HOURS",
                "Times": [
                    "01:00"
                ]
            },
            RetainRule":{
                "Interval" : 2,
                "IntervalUnit" : "DAYS"
            },
            DeprecateRule": {
                "Interval" : 1,
                "IntervalUnit" : "DAYS"
            },
            "CopyTags": true
        }
    ],
    "Parameters" : {
        "NoReboot":true
    }
}
```

요청이 성공하면 명령은 새로 생성된 정책의 ID를 반환합니다. 다음은 예제 출력입니다.

```
{
   "PolicyId": "policy-9876543210abcdef0"
}
```

예 2: 교차 리전 복사를 사용한 개수 기준 보존 및 AMI 사용 중단

이 예에서는 값이 production인 purpose 태그 키가 있는 모든 인스턴스의 AMI를 생성하고 대상 인스턴스를 재부팅하는 AMI 수명 주기 정책을 생성합니다. 이 정책에는 매일 17:30 UTC에 6시간마다 AMI를 생성하는 하나의 일정이 포함됩니다. 이 정책은 3개의 AMI를 보존하며 가장 오래된 2개의 AMI를 자동으로 사용 중단합니다. 또한 AMI를 us-east-1에 복사하고, 2개의 AMI 사본을 보존하며, 가장 오래된 AMI를 자동으로 사용 중단하는 교차 리전 복사 규칙이 포함되어 있습니다.

```
aws dlm create-lifecycle-policy \
    --description "My AMI policy" \
    --state ENABLED \
    --execution-role-arn
 arn:aws:iam::12345678910:role/AWSDataLifecycleManagerDefaultRoleForAMIManagement \
    --policy-details file://policyDetails.json
```

다음은 policyDetails.json 파일의 예입니다.

```
{
    "PolicyType": "IMAGE_MANAGEMENT",
    "ResourceTypes" : [
        "INSTANCE"
    ],
    "TargetTags": [{
        "Key":"purpose",
        "Value":"production"
    }],
    "Parameters" : {
          "NoReboot": true
    },
    "Schedules" : [{
        "Name" : "Schedule1",
        "CopyTags": true,
        "CreateRule" : {
            "Interval": 6,
            "IntervalUnit": "HOURS",
            "Times" : ["17:30"]
        },
        "RetainRule":{
            "Count" : 3
        },
        "DeprecateRule":{
            "Count" : 2
        },
        "CrossRegionCopyRules": [{
```

```
            "TargetRegion": "us-east-1",
            "Encrypted": true,
            "RetainRule":{
                "IntervalUnit": "DAYS",
                "Interval": 2
            },
            "DeprecateRule":{
                "IntervalUnit": "DAYS",
                "Interval": 1
            },
            "CopyTags": true
        }]
    }]
}
```

AMI 수명 주기 정책 고려 사항

다음은 AMI 수명 주기 정책 생성에 적용되는 일반적인 고려 사항입니다.

- AMI 수명 주기 정책은 정책과 동일한 리전에 있는 인스턴스만 대상으로 합니다.
- 지정된 시작 시간 후 1시간 내에 첫 번째 AMI 생성 작업이 시작됩니다. 후속 AMI 생성 작업은 예약된 시간으로부터 1시간 이내에 시작됩니다.
- Amazon Data Lifecycle Manager AMI의 등록을 취소하면 백업 스냅샷이 자동으로 삭제됩니다.
- 대상 리소스 태그는 대소문자를 구분합니다.
- 정책의 대상이 되는 인스턴스에서 대상 태그를 제거할 경우, Amazon Data Lifecycle Manager는 표준 티어의 기존 AMI를 더 이상 관리하지 않습니다. 따라서 더 이상 필요하지 않은 경우 수동으로 AMI를 삭제해야 합니다.
- 인스턴스를 백업하도록 여러 정책을 생성할 수 있습니다. 인스턴스에 12시간마다 AMI를 만드는 정책 A의 대상 태그인 태그 A와 24시간마다 AMI를 만드는 정책 B의 대상 태그인 태그 B, 이렇게 2개의 태그가 있다면 Amazon Data Lifecycle Manager는 양쪽 정책의 일정에 따라 AMI를 생성합니다. 여러 일정이 있는 단일 정책을 생성해도 동일한 결과를 얻을 수 있습니다. 예를 들어 태그 A만 대상으로 지정하는 단일 정책을 생성하고 12시간 간격과 24시간 간격의 2개 일정을 지정할 수 있습니다.
- 정책이 생성된 후 대상 인스턴스에 연결된 새 볼륨은 다음 정책 실행 시 백업에 자동으로 포함됩니다. 정책이 실행되면 인스턴스에 연결된 모든 볼륨이 포함됩니다.
- 사용자 지정 cron 기반 예약이 있는 정책이 단일 AMI만 생성하도록 구성된 경우 이 정책은 보존 임계값에 도달할 때 AMI를 자동으로 등록 취소하지 않습니다. 더 이상 필요하지 않은 경우 마지막 AMI를 수동으로 등록 취소해야 합니다.

- 보존 기간이 생성 빈도보다 짧은 경과 시간 기반 정책을 생성할 경우, Amazon Data Lifecycle Manager는 항상 다음 AMI가 생성될 때까지 마지막 AMI를 보존합니다. 예를 들어 경과 시간 기반 정책에서 보존 기간이 7일인 AMI를 매달 한 개씩 생성하는 경우, 보존 기간이 7일이더라도 Amazon Data Lifecycle Manager는 각 AMI를 한 달 동안 보존합니다.
- 수량 기반 정책의 경우 Amazon Data Lifecycle Manager는 보존 정책에 따라 가장 오래된 AMI의 등록을 취소하기 전에 항상 생성 빈도에 따라 AMI를 생성합니다.
- AMI를 등록 취소하고 관련 기반 스냅샷을 삭제하는 데 몇 시간이 걸릴 수 있습니다. 이전에 생성한 AMI의 등록이 취소되기 전에 Amazon Data Lifecycle Manager가 다음 AMI를 생성할 경우, 일시적으로 보존 수량보다 많은 AMI를 보존하게 될 수 있습니다.

다음은 정책에 의해 대상으로 지정된 인스턴스 종료에 적용되는 고려 사항입니다.

- 개수 기반 보존 일정을 사용하는 정책에 의해 대상으로 지정된 인스턴스를 종료할 경우, 이전에 종료된 인스턴스에서 생성한 AMI는 더 이상 정책으로 관리되지 않습니다. 더 이상 필요하지 않은 이전의 AMI는 수동으로 등록 취소해야 합니다.
- 연령 기반 보존 일정을 사용하는 정책에 의해 대상으로 지정된 인스턴스를 종료할 경우, 정책은 이전에 정의된 일정의 종료된 인스턴스에서 생성된 AMI를 계속 등록 취소합니다(마지막 AMI 제외). 더 이상 필요하지 않은 경우 마지막 AMI를 수동으로 등록 취소해야 합니다.

다음 고려 사항은 AMI 정책 및 AMI 사용 중단에 적용됩니다.

- 개수 기준 보존이 적용된 일정에 대한 AMI 사용 중단 수를 늘리면 해당 일정에 의해 생성된 모든 AMI(기존 및 신규)에 변경 사항이 적용됩니다.
- 경과 시간 기반 보존 일정에서 AMI 사용 중단 기간을 늘리면 변경 사항은 새 AMI에만 적용됩니다. 기존 AMI는 영향을 받지 않습니다.
- 일정에서 AMI 사용 중단 규칙을 제거해도 Amazon Data Lifecycle Manager는 이전에 해당 일정에 의해 사용 중단된 AMI의 사용 중단을 취소하지 않습니다.
- 일정의 AMI 사용 중단 수 또는 기간을 줄여도 Amazon Data Lifecycle Manager는 이전에 해당 일정에 의해 사용 중단된 AMI의 사용 중단을 취소하지 않습니다.
- AMI 정책에 의해 생성된 AMI를 수동으로 사용 중단할 경우 Amazon Data Lifecycle Manager는 사용 중단을 재정의하지 않습니다.
- AMI 정책에 의해 이전에 생성된 AMI의 사용 중단을 수동으로 취소할 경우 Amazon Data Lifecycle Manager는 취소를 재정의하지 않습니다.
- 충돌하는 여러 일정에 의해 AMI가 생성되고 하나 이상의 일정에 AMI 사용 중단 규칙이 없는 경우 Amazon Data Lifecycle Manager는 해당 AMI를 사용 중단하지 않습니다.

- 충돌하는 여러 일정에 의해 AMI가 생성되고 해당하는 모든 일정에 AMI 사용 중단 규칙이 있는 경우 Amazon Data Lifecycle Manager는 사용 중단 날짜가 가장 늦은 결과로 사용 중단 규칙을 사용합니다.

다음은 AMI 정책과 휴지통에 적용되는 고려 사항입니다.

- 정책의 보존 임계값에 도달하면 Amazon Data Lifecycle Manager가 AMI 등록을 취소하고 휴지통으로 보내고 사용자가 휴지통에서 해당 AMI를 수동으로 복원하는 경우 더 이상 필요하지 않을 때 AMI를 수동으로 등록 취소해야 합니다. Amazon Data Lifecycle Manager는 더 이상 AMI를 관리하지 않습니다.
- 정책에 의해 생성된 AMI를 수동으로 등록 취소하고 정책의 보존 임계값에 도달했을 때 해당 AMI가 휴지통에 있는 경우 Amazon Data Lifecycle Manager는 AMI 등록을 취소하지 않습니다. Amazon Data Lifecycle Manager는 휴지통에 있는 AMI를 관리하지 않습니다.

  정책의 보존 임계값에 도달하기 전에 AMI가 휴지통에서 복원되는 경우 정책의 보존 임계값에 도달하면 Amazon Data Lifecycle Manager가 AMI 등록을 취소합니다.

  정책의 보존 임계값에 도달한 후 AMI가 휴지통에서 복원되면 Amazon Data Lifecycle Manager가 더 이상 AMI를 등록 취소하지 않습니다. 더 이상 필요하지 않은 AMI는 수동으로 삭제해야 합니다.

다음 고려 사항은 오류 상태인 AMI 정책에 적용됩니다.

- 수명 기반 보존 일정이 포함된 정책의 경우 정책이 `error` 상태일 때 만료되도록 설정된 AMI는 무기한 보존됩니다. AMI는 수동으로 등록 취소해야 합니다. 정책을 다시 활성화하고 보존 기간이 만료되면 Amazon Data Lifecycle Manager가 AMI 등록 취소를 재개합니다.
- 개수 기반 보존 일정이 있는 정책의 경우 `error` 상태인 동안 AMI 생성 및 등록 취소를 중단합니다. 정책을 다시 활성화하면 Amazon Data Lifecycle Manager가 AMI 생성을 재개하고 보존 임계값이 충족되면 AMI 등록 취소를 재개합니다.

다음은 AMI 정책 및 AMI 비활성화에 적용되는 고려 사항입니다.

- Amazon Data Lifecycle Manager에서 생성한 AMI를 비활성화하고, 보존 임계값에 도달했을 때 해당 AMI가 비활성화되는 경우, Amazon Data Lifecycle Manager에서 AMI 등록을 취소하고 연결된 스냅샷을 삭제합니다.
- Amazon Data Lifecycle Manager에서 생성한 AMI를 비활성화하고 연결된 스냅샷을 수동으로 보관하고, 보존 임계값에 도달했을 때 이러한 스냅샷이 보관되는 경우, Amazon Data Lifecycle Manager에서 이러한 스냅샷을 삭제하지 않으며 더 이상 관리하지 않습니다.

다음은 AMI 정책 및 AMI 등록 취소 보호에 적용되는 고려 사항입니다.

- 사용자가 Amazon Data Lifecycle Manager에서 생성한 AMI에 대해 수동으로 등록 취소 보호를 활성화하고 AMI 보존 임계값에 도달해도 여전히 활성화되면 Amazon Data Lifecycle Manager는 더 이상 해당 AMI를 관리하지 않습니다. 더 이상 필요하지 않은 경우 AMI를 수동으로 등록 취소하고 기본 스냅샷을 삭제해야 합니다.

추가 리소스

자세한 내용은 Amazon Data Lifecycle Manager 스토리지를 사용하여 Amazon EBS 스냅샷 및 AMI 관리 자동화 AWS 블로그를 참조하세요.

Data Lifecycle Manager를 사용하여 교차 계정 스냅샷 복사 자동화

교차 계정 스냅샷 복사를 자동화하면 Amazon EBS 스냅샷을 격리된 계정의 특정 리전으로 복사하고 암호화 키를 사용하여 해당 스냅샷을 암호화할 수 있습니다. 이렇게 하면 계정이 손상될 경우 데이터 손실로부터 보호할 수 있습니다.

교차 계정 스냅샷 복사를 자동화하려면 다음 두 개의 계정이 필요합니다.

- 소스 계정—소스 계정은 스냅샷을 생성하고 대상 계정과 공유하는 계정입니다. 이 계정에서는 설정된 간격으로 스냅샷을 생성한 다음 다른 AWS 계정과 공유하는 EBS 스냅샷 정책을 생성해야 합니다.
- 대상 계정—대상 계정은 스냅샷이 공유되는 대상 계정이 있는 계정이며 공유 스냅샷의 복사본을 생성하는 계정입니다. 이 계정에서는 하나 이상의 지정된 소스 계정에서 공유되는 스냅샷을 자동으로 복사하는 교차 계정 복사 이벤트 정책을 생성해야 합니다.

> **Note**
>
> 소스 계정 EBS 스냅샷 정책과 대상 계정 교차 계정 복사 이벤트 정책은 모두 동일한 AWS 리전에서 생성해야 합니다. 그러면 대상 계정은 필요에 따라 스냅샷을 다른 대상 리전으로 복사할 수 있습니다.

주제

- 교차 계정 스냅샷 복사 정책 생성

- 스냅샷 설명 필터 지정
- 교차 계정 스냅샷 복사 정책 고려 사항
- 추가 리소스

교차 계정 스냅샷 복사 정책 생성

교차 계정 스냅샷 복사를 위한 소스 및 대상 계정을 준비하려면 다음 단계를 수행해야 합니다.

1단계: EBS 스냅샷 정책 생성(소스 계정)

소스 계정에서 스냅샷을 생성하고 필요한 대상 계정과 공유하는 EBS 스냅샷 정책을 생성합니다.

정책을 생성할 때 교차 계정 공유를 활성화하고 스냅샷을 공유할 대상 AWS 계정을 지정해야 합니다. 이러한 계정은 스냅샷이 공유되는 계정입니다. 암호화된 스냅샷을 공유하는 경우 선택한 대상 계정에 소스 볼륨을 암호화하는 데 사용된 KMS 키를 사용할 수 있는 권한을 부여해야 합니다. 자세한 내용은 2단계: 고객 관리형 키(소스 계정) 공유 단원을 참조하십시오.

> **Note**
>
> - 3단계에서 대상 계정의 교차 계정 복사 이벤트 정책을 생성할 리전과 동일한 AWS 리전에서 이 정책을 생성합니다. 교차 계정 스냅샷 공유가 제대로 작동하려면 두 정책이 동일한 리전에 있어야 합니다.
> - 암호화되지 않았거나 고객 관리형 키를 사용하여 암호화된 스냅샷만 공유할 수 있습니다. 기본 EBS 암호화 KMS 키(으)로 암호화된 스냅샷은 공유할 수 없습니다. 암호화된 스냅샷을 공유하는 경우 소스 볼륨을 암호화하는 데 사용된 KMS 키도 대상 계정과 공유해야 합니다. 자세한 내용은 AWS Key Management Service 개발자 안내서의 다른 계정의 사용자가 CMK를 사용하도록 허용을 참조하세요.

EBS 스냅샷 정책을 생성하는 방법에 대한 자세한 내용은 EBS 스냅샷에 대한 Amazon Data Lifecycle Manager 사용자 지정 정책 생성 섹션을 참조하세요.

다음 방법 중 하나를 사용하여 EBS 스냅샷 정책을 생성합니다.

2단계: 고객 관리형 키(소스 계정) 공유

암호화된 스냅샷을 공유하는 경우 소스 볼륨을 암호화하는 데 사용된 고객 관리형 키를 사용할 수 있는 권한을 이전 단계에서 선택한 IAM 역할 및 대상 AWS 계정에 부여해야 합니다.

> **Note**
>
> 이 단계는 암호화된 스냅샷을 공유하는 경우에만 수행합니다. 암호화되지 않은 스냅샷을 공유하는 경우 이 단계를 건너뜁니다.

Console

1. https://console.aws.amazon.com/kms AWS KMS 콘솔을 엽니다.
2. 를 변경하려면 페이지 오른쪽 상단에 있는 리전 선택기를 AWS 리전사용합니다.
3. 탐색 창에서 [고객 관리형 키(Customer managed keys)]를 선택한 다음 대상 계정과 공유해야 하는 KMS 키를 선택합니다.

   KMS 키 ARN을 기록해 두세요. 나중에 필요합니다.
4. [키 정책(Key policy)] 탭에서 [키 사용자(Key users)] 섹션까지 아래로 스크롤합니다. [추가(Add)]를 선택하고 이전 단계에서 선택한 IAM 역할의 이름을 입력한 다음 [추가(Add)]를 선택합니다.
5. 키 정책 탭에서 다른 AWS 계정 단원까지 아래로 스크롤합니다. 다른 AWS 계정 추가를 선택한 다음 이전 단계에서 스냅샷을 공유하도록 선택한 모든 대상 AWS 계정을 추가합니다.
6. 변경 사항 저장을 선택합니다.

Command line

get-key-policy 명령을 사용하여 현재 KMS 키에 연결된 키 정책을 다시 확인합니다.

예를 들어 다음 명령은 ID가 `9d5e2b3d-e410-4a27-a958-19e220d83a1e`인 KMS 키에 대한 키 정책을 검색하여 `snapshotKey.json`(이)라는 파일에 씁니다.

```
$ aws kms get-key-policy \
    --policy-name default \
    --key-id 9d5e2b3d-e410-4a27-a958-19e220d83a1e \
    --query Policy \
    --output text > snapshotKey.json
```

기본 텍스트 편집기를 사용하여 키 정책을 엽니다. 스냅샷 정책을 생성할 때 지정한 IAM 역할의 ARN과 KMS 키를 공유할 대상 계정의 ARN을 추가합니다.

예를 들어 다음 정책에서는 대상 계정에 대한 루트 계정의 ARN과 함께 기본 IAM 역할의 ARN이 지정됩니다. 222222222222.

> **Tip**
>
> 최소 권한의 원칙을 따르려면 kms:CreateGrant에 대한 전체 액세스 권한을 허용하지 마세요. 대신 다음 예제와 AWS 같이 kms:GrantIsForAWSResource 조건 키를 사용하여 서비스가 사용자를 대신하여 권한 부여를 생성할 때만 사용자가 KMS 키에 대한 권한 부여를 생성하도록 허용합니다.

```
{
    "Sid" : "Allow use of the key",
    "Effect" : "Allow",
    "Principal" : {
        "AWS" : [
            "arn:aws:iam::111111111111:role/service-role/
AWSDataLifecycleManagerDefaultRole",
            "arn:aws:iam::222222222222:root"
        ]
    },
    "Action" : [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
        "kms:DescribeKey"
    ],
    "Resource" : "*"
},
{
    "Sid" : "Allow attachment of persistent resources",
    "Effect" : "Allow",
    "Principal" : {
        "AWS" : [
            "arn:aws:iam::111111111111:role/service-role/
AWSDataLifecycleManagerDefaultRole",
            "arn:aws:iam::222222222222:root"
        ]
    },
    "Action" : [
        "kms:CreateGrant",
```

```
        "kms:ListGrants",
        "kms:RevokeGrant"
    ],
    "Resource" : "*",
    "Condition" : {
        "Bool" : {
          "kms:GrantIsForAWSResource" : "true"
        }
    }
}
```

파일을 저장하고 닫습니다. 그런 다음 put-key-policy 명령을 사용하여 업데이트된 키 정책을 KMS 키에 연결합니다.

```
$ aws kms put-key-policy \
    --policy-name default \
    --key-id 9d5e2b3d-e410-4a27-a958-19e220d83a1e \
    --policy file://snapshotKey.json
```

3단계: 교차 계정 복사 이벤트 정책 생성(대상 계정)

대상 계정에서는 필요한 소스 계정에서 공유하는 스냅샷을 자동으로 복사하는 교차 계정 복사 이벤트 정책을 생성해야 합니다.

이 정책은 지정된 소스 계정 중 하나가 계정과 스냅샷을 공유하는 경우에만 대상 계정에서 실행됩니다.

> **Note**
>
> 1단계에서 생성된 소스 계정의 EBS 스냅샷 정책과 동일한 AWS 리전에서이 정책을 생성합니다. 교차 계정 스냅샷 공유가 제대로 작동하려면 두 정책이 동일한 리전에 있어야 합니다. 그런 다음 필요에 따라 스냅샷을 다른 대상 리전에 복사하도록 이 정책을 구성할 수 있습니다.

교차 계정 복사 이벤트 정책을 생성하려면 다음 방법 중 하나를 사용합니다.

Console

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [Elastic Block Store], [Lifecycle Manager] 및 [수명 주기 정책 생성]을 차례로 선택합니다.

3. [정책 유형 선택(Select policy type)] 화면에서 [계정 간 복사 이벤트 정책(Cross-account copy event policy)]을 선택한 후 [다음(Next)]을 선택합니다.
4. [정책 설명(Policy description)]에 정책에 대한 간략한 설명을 입력합니다.
5. [정책 태그(Policy tags)]에서 수명 주기 정책에 적용할 태그를 추가합니다. 이 태그를 사용하여 정책을 식별 및 분류할 수 있습니다.
6. [이벤트 설정(Event settings)] 섹션에서 정책 실행을 트리거할 스냅샷 공유 이벤트를 정의합니다. 해결 방법:

   a. 계정 공유에서 공유 스냅샷을 복사할 소스 AWS 계정을 지정합니다. 계정 추가를 선택하고 12자리 AWS 계정 ID를 입력한 다음 추가를 선택합니다.

   b. [설명으로 필터링(Filter by description)]에 정규식을 사용하여 필요한 스냅샷 설명을 입력합니다. 지정된 소스 계정에서 공유되고 지정된 필터와 일치하는 설명이 있는 스냅샷만 정책에 의해 복사됩니다. 자세한 내용은 스냅샷 설명 필터 지정 섹션을 참조하세요.
7. [IAM 역할(IAM role)]에서 스냅샷 복사 작업을 수행할 권한이 있는 IAM 역할을 선택합니다. Amazon Data Lifecycle Manager가 제공하는 기본 역할을 사용하려면 [기본 역할(Default role)]을 선택합니다. 또는 이전에 생성한 사용자 지정 IAM 역할을 사용하려면 [다른 역할 선택(Choose another role)]을 선택하고 사용할 역할을 선택합니다.

   암호화된 스냅샷을 복사하는 경우 소스 볼륨을 암호화하는 데 사용되는 암호화 KMS 키를 사용할 수 있는 권한을 선택한 IAM 역할에 부여해야 합니다. 마찬가지로 다른 KMS 키를 사용하여 대상 리전의 스냅샷을 암호화하는 경우 IAM 역할에 대상 KMS 키를 사용할 수 있는 권한을 부여해야 합니다. 자세한 내용은 4단계: IAM 역할에서 필요한 KMS 키를 사용할 수 있도록 허용(대상 계정) 섹션을 참조하세요.
8. [복사 작업(Copy action)] 섹션에서 정책이 활성화될 경우 수행할 스냅샷 복사 작업을 정의합니다. 이 정책은 스냅샷을 최대 3개의 리전에 복사할 수 있습니다. 각 대상 리전에 대해 별도의 복사 규칙을 지정해야 합니다. 추가하는 각 규칙 그룹에 대해 다음을 수행합니다.

   a. [이름(Name)]에 복사 작업을 설명하는 이름을 입력합니다.

   b. [대상 리전(Target Region)]에서 스냅샷을 복사할 리전을 선택합니다.

   c. [만료(Expire)]에서 스냅샷 복사본을 생성한 후 대상 리전에 보존할 기간을 지정합니다.

   d. 스냅샷 복사본을 암호화하려면 [암호화(Encryption)]에서 [암호화 사용(Enable encryption)]을 선택합니다. 소스 스냅샷이 암호화되거나 계정에서 기본적으로 암호화가 사용되는 경우 여기에서 암호화를 사용하지 않더라도 스냅샷 사본이 항상 암호화됩니다. 소스 스냅샷이 암호화되지 않았고 계정에서 암호화가 기본적으로 활성화되어 있지 않은 경우 암호화를 활성화하거나 비활성화하도록 선택할 수 있습니다. 암호화를 활성화하고 KMS 키를 지정하지 않으면 각 대상 리전의 기본 암호화 KMS 키를 사용하여 스냅샷이 암

호화됩니다. 대상 리전에 대한 KMS 키를 지정하는 경우 KMS 키에 대한 액세스 권한이 있어야 합니다.

9. 스냅샷 복사 작업을 더 추가하려면 [새 리전 추가(Add new Regions)]를 선택합니다.
10. 생성 후 정책 상태(Policy status after creation) - 정책 활성화(Enable policy)를 선택하여 다음 예약 시간에 정책 실행을 시작하거나, 정책 비활성화(Disable policy)를 선택하여 정책을 실행하지 않습니다. 지금 정책을 활성화하지 않으면 생성 후 수동으로 활성화할 때까지 스냅샷 복사가 시작되지 않습니다.
11. 정책 생성을 선택합니다.

Command line

create-lifecycle-policy 명령을 사용하여 정책을 생성합니다. 교차 계정 복사 이벤트 정책을 생성하려면 PolicyType에 EVENT_BASED_POLICY를 지정합니다.

예를 들어 다음 명령은 대상 계정 222222222222에서 교차 계정 복사 이벤트 정책을 생성합니다. 정책은 소스 계정 111111111111에서 공유하는 스냅샷을 복사합니다. 정책은 스냅샷을 sa-east-1 및 eu-west-2에 복사합니다. sa-east-1에 복사된 스냅샷은 암호화되지 않으며 3일 동안 보존됩니다. eu-west-2에 복사된 스냅샷은 KMS 키 8af79514-350d-4c52-bac8-8985e84171c7을(를) 사용하여 암호화되며 1개월 동안 보존됩니다. 정책은 기본 IAM 역할을 사용합니다.

```
$ aws dlm create-lifecycle-policy \
    --description "Copy policy" \
    --state ENABLED \
    --execution-role-arn arn:aws:iam::222222222222:role/service-role/
AWSDataLifecycleManagerDefaultRole \
    --policy-details file://policyDetails.json
```

다음은 policyDetails.json 파일의 콘텐츠를 보여줍니다.

```
{
    "PolicyType" : "EVENT_BASED_POLICY",
    "EventSource" : {
        "Type" : "MANAGED_CWE",
        "Parameters": {
            "EventType" : "shareSnapshot",
            "SnapshotOwner": ["111111111111"]
        }
```

```
    },
    "Actions" : [{
        "Name" :"Copy Snapshot to Sao Paulo and London",
        "CrossRegionCopy" : [{
            "Target" : "sa-east-1",
             "EncryptionConfiguration" : {
                 "Encrypted" : false
             },
             "RetainRule" : {
             "Interval" : 3,
            "IntervalUnit" : "DAYS"
            }
        },
        {
            "Target" : "eu-west-2",
            "EncryptionConfiguration" : {
                 "Encrypted" : true,
                 "CmkArn" : "arn:aws:kms:eu-
west-2:222222222222:key/8af79514-350d-4c52-bac8-8985e84171c7"
            },
            "RetainRule" : {
                "Interval" : 1,
                "IntervalUnit" : "MONTHS"
            }
        }]
    }]
}
```

요청이 성공하면 명령은 새로 생성된 정책의 ID를 반환합니다. 다음은 예제 출력입니다.

```
{
    "PolicyId": "policy-9876543210abcdef0"
}
```

4단계: IAM 역할에서 필요한 KMS 키를 사용할 수 있도록 허용(대상 계정)

암호화된 스냅샷을 복사하는 경우 소스 볼륨을 암호화하는 데 사용된 고객 관리형 키를 사용할 수 있는 권한을 이전 단계에서 선택한 IAM 역할에 부여해야 합니다.

> **Note**
>
> 이 단계는 암호화된 스냅샷을 복사하는 경우에만 수행합니다. 암호화되지 않은 스냅샷을 복사하는 경우 이 단계를 건너뜁니다.

다음 방법 중 하나를 사용하여 필요한 정책을 IAM 역할에 추가합니다.

Console

1. IAM 콘솔(https://console.aws.amazon.com/iam/)을 엽니다.
2. 탐색 창에서 [역할(Roles)]을 선택합니다. 이전 단계에서 교차 계정 복사 이벤트 정책을 생성할 때 선택한 IAM 역할을 검색하고 선택합니다. 기본 역할을 사용하도록 선택한 경우 역할의 이름은 AWSDataLifecycleManagerDefaultRole로 지정됩니다.
3. [인라인 정책 추가(Add inline policy)]를 선택한 다음 [JSON] 탭을 선택합니다.
4. 기존 정책을 다음으로 바꾸고 소스 볼륨을 암호화하는 데 사용되었으며 2단계에서 소스 계정이 공유한 KMS 키의 ARN을 지정합니다.

   > **Note**
   >
   > 여러 소스 계정에서 복사하는 경우 각 소스 계정에서 해당 KMS 키 ARN을 지정해야 합니다.

   다음 예에서 정책은 소스 계정 `1234abcd-12ab-34cd-56ef-1234567890ab`에서 공유한 KMS 키 `111111111111`와 대상 계정 `4567dcba-23ab-34cd-56ef-0987654321yz`에 있는 KMS 키 `222222222222`을(를) 사용할 권한을 IAM 역할에 부여합니다.

   > **Tip**
   >
   > 최소 권한의 원칙을 따르려면 `kms:CreateGrant`에 대한 전체 액세스 권한을 허용하지 마세요. 대신 다음 예제와 AWS 같이 `kms:GrantIsForAWSResource` 조건 키를 사용하여 서비스가 사용자를 대신하여 권한 부여를 생성할 때만 사용자가 KMS 키에 대한 권한 부여를 생성하도록 허용합니다.

JSON

```
{
   "Version":"2012-10-17",
   "Statement": [
       {
           "Effect": "Allow",
           "Action": [
               "kms:RevokeGrant",
               "kms:CreateGrant",
               "kms:ListGrants"
           ],
           "Resource": [
               "arn:aws:kms:us-
east-1:111111111111:key/1234abcd-12ab-34cd-56ef-1234567890ab",
               "arn:aws:kms:us-
east-1:222222222222:key/4567dcba-23ab-34cd-56ef-0987654321yz"
           ],
           "Condition": {
               "Bool": {
                   "kms:GrantIsForAWSResource": "true"
               }
           }
       },
       {
           "Effect": "Allow",
           "Action": [
               "kms:Encrypt",
               "kms:Decrypt",
               "kms:ReEncrypt*",
               "kms:GenerateDataKey*",
               "kms:DescribeKey"
           ],
           "Resource": [
               "arn:aws:kms:us-
east-1:111111111111:key/1234abcd-12ab-34cd-56ef-1234567890ab",
               "arn:aws:kms:us-
east-1:222222222222:key/4567dcba-23ab-34cd-56ef-0987654321yz"
           ]
       }
   ]
```

```
}
```

5. [정책 검토(Review policy)]를 선택합니다.
6. [이름(Name)]에 정책을 설명하는 이름을 입력한 다음 [정책 생성(Create policy)]을 선택합니다.

Command line

선호하는 텍스트 편집기를 사용하여 `policyDetails.json`이라는 새 JSON 파일을 생성합니다. 다음 정책을 추가하고 소스 볼륨을 암호화하는 데 사용되었으며 2단계에서 소스 계정이 공유한 KMS 키의 ARN을 지정합니다.

> **Note**
>
> 여러 소스 계정에서 복사하는 경우 각 소스 계정에서 해당 KMS 키 ARN을 지정해야 합니다.

다음 예에서 정책은 소스 계정 `1234abcd-12ab-34cd-56ef-1234567890ab`에서 공유한 KMS 키 `111111111111`와 대상 계정 `4567dcba-23ab-34cd-56ef-0987654321yz`에 있는 KMS 키 `222222222222`을(를) 사용할 권한을 IAM 역할에 부여합니다.

> **Tip**
>
> 최소 권한의 원칙을 따르려면 `kms:CreateGrant`에 대한 전체 액세스 권한을 허용하지 마세요. 대신 다음 예제와 AWS 같이 `kms:GrantIsForAWSResource` 조건 키를 사용하여 서비스가 사용자를 대신하여 권한 부여를 생성할 때만 사용자가 KMS 키에 대한 권한 부여를 생성하도록 허용합니다.

JSON

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

```
                "kms:ListGrants"
            ],
            "Resource": [
                "arn:aws:kms:us-
east-1:111111111111:key/1234abcd-12ab-34cd-56ef-1234567890ab",
                "arn:aws:kms:us-
east-1:222222222222:key/4567dcba-23ab-34cd-56ef-0987654321yz"
            ],
            "Condition": {
                "Bool": {
                    "kms:GrantIsForAWSResource": "true"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:Encrypt",
                "kms:Decrypt",
                "kms:ReEncrypt*",
                "kms:GenerateDataKey*",
                "kms:DescribeKey"
            ],
            "Resource": [
                "arn:aws:kms:us-
east-1:111111111111:key/1234abcd-12ab-34cd-56ef-1234567890ab",
                "arn:aws:kms:us-
east-1:222222222222:key/4567dcba-23ab-34cd-56ef-0987654321yz"
            ]
        }
    ]
}
```

파일을 저장하고 닫습니다. 그런 다음 put-role-policy 명령을 사용하여 IAM 역할에 정책을 추가합니다.

예

```
$ aws iam put-role-policy \
    --role-name AWSDataLifecycleManagerDefaultRole \
    --policy-name CopyPolicy \
    --policy-document file://AdminPolicy.json
```

스냅샷 설명 필터 지정

대상 계정에서 스냅샷 복사 정책을 생성할 때는 스냅샷 설명 필터를 지정해야 합니다. 스냅샷 설명 필터를 사용하면 정책에 의해 복사되는 스냅샷을 제어할 때 사용할 수 있는 추가 수준의 필터링을 지정할 수 있습니다. 즉, 지정된 소스 계정 중 하나에서 스냅샷을 공유하고 스냅샷 설명이 지정된 필터와 일치하는 경우에만 정책을 통해 스냅샷이 복사됩니다. 다시 말해, 지정된 소스 계정 중 하나에서 스냅샷을 공유하지만 지정된 필터와 일치하는 설명이 없는 경우에는 정책에 의해 복사되지 않습니다.

스냅샷 필터 설명은 정규 표현식을 사용하여 지정해야 합니다. 콘솔과 명령줄을 사용하여 교차 계정 복사 이벤트 정책을 생성하는 경우에는 필수 필드입니다. 다음은 사용할 수 있는 예제 정규 표현식입니다.

- `.*`—이 필터는 모든 스냅샷 설명을 일치시킵니다. 이 식을 사용하면 지정된 소스 계정 중 하나가 공유하는 모든 스냅샷이 정책에 의해 복사됩니다.
- `Created for policy: policy-0123456789abcdef0.*`—이 필터는 ID가 인 정책에 의해 생성된 스냅샷만 일치시킵니다.`policy-0123456789abcdef0` 이와 같은 식을 사용하는 경우 지정된 소스 계정 중 하나에서 계정과 공유하고 지정된 ID를 가진 정책에 의해 생성된 스냅샷만 정책에 의해 복사됩니다.
- `.*production.*`—이 필터는 설명에 `production`이라는 단어가 있는 모든 스냅샷을 일치시킵니다. 이 식을 사용하는 경우 지정된 소스 계정 중 하나에서 공유되고 설명에 지정된 텍스트가 있는 모든 스냅샷이 정책에 의해 복사됩니다.

교차 계정 스냅샷 복사 정책 고려 사항

교차 계정 복사 이벤트 정책에는 다음 고려 사항이 적용됩니다.

- 소스 계정 EBS 스냅샷 정책과 대상 계정 교차 계정 복사 이벤트 정책은 동일한 AWS 리전에서 생성해야 합니다. 스냅샷이 공유된 후, 대상 계정 정책은 복사 작업에 지정된 대로 스냅샷을 다른 대상 리전에 복사할 수 있습니다.
- 암호화되지 않았거나 고객 관리형 키를 사용하여 암호화된 스냅샷만 복사할 수 있습니다.
- Amazon Data Lifecycle Manager 외부에서 공유되는 스냅샷을 복사하는 교차 계정 복사 이벤트 정책을 생성할 수 있습니다.
- 대상 계정의 스냅샷을 암호화하려면 교차 계정 복사 이벤트 정책에 대해 선택한 IAM 역할에 필요한 KMS 키를 사용할 수 있는 권한이 있어야 합니다.

추가 리소스

자세한 내용은 AWS 계정 스토리지 간에 암호화된 Amazon EBS 스냅샷 복사 자동화 AWS 블로그를 참조하세요.

Amazon Data Lifecycle Manager 정책 수정

Amazon Data Lifecycle Manager 정책을 수정할 때는 다음 사항을 유의해야 합니다.

- 대상 태그를 변경하거나 삭제하여 AMI 또는 스냅샷 정책을 수정하면 그러한 태그가 연결된 볼륨 또는 인스턴스가 더 이상 해당 정책으로 관리되지 않습니다.
- 일정 이름을 수정하면 예전의 일정 이름으로 생성된 스냅샷 또는 AMI에 더 이상 해당 정책이 적용되지 않습니다.
- 새 시간 간격을 사용하도록 연령 기반 보존 일정을 수정할 경우, 변경 후 생성된 새 스냅샷 또는 AMI에만 새 간격이 사용됩니다. 새로운 일정은 변경 전에 생성된 스냅샷 또는 AMI의 보존 일정에 영향을 주지 않습니다.
- 정책을 생성한 후에는 개수 기반에서 연령 기반으로 정책의 보존 일정을 변경할 수 없습니다. 이렇게 변경하려면 새 정책을 생성해야 합니다.
- 수명 기반 보존 일정이 포함된 정책을 비활성화하면 만료되도록 설정된 스냅샷 또는 AMI가 정책이 비활성화된 동안 무기한 보존됩니다. 스냅샷을 삭제하거나 수동으로 AMI를 등록 취소해야 합니다. 정책을 다시 활성화하고 보존 기간이 만료되면 Amazon Data Lifecycle Manager가 스냅샷 삭제 또는 AMI 등록 취소를 재개합니다.
- 개수 기반 보존 일정이 있는 정책을 비활성화하면 정책이 스냅샷 또는 AMI 생성 및 삭제를 중지합니다. 정책을 다시 활성화하면 Amazon Data Lifecycle Manager가 스냅샷 및 AMI 생성을 재개하고 보존 임계값이 충족되면 스냅샷 또는 AMI 삭제를 재개합니다.
- 스냅샷 아카이빙이 활성화된 정책이 있는 정책을 비활성화하면 정책 비활성화 시 아카이브 계층에 있는 스냅샷은 더 이상 Amazon Data Lifecycle Manager Manager에서 관리되지 않습니다. 더 이상 필요하지 않은 스냅샷은 수동으로 삭제해야 합니다.
- 개수 기반 일정에 따라 스냅샷 아카이빙을 활성화하면 일정에 따라 생성되고 아카이브되는 모든 새 스냅샷에 아카이빙 규칙이 적용되고 일정에 따라 이전에 생성 및 아카이브된 기존 스냅샷에도 적용됩니다.
- 기간 기반 일정에 따라 스냅샷 아카이빙을 활성화하면 아카이빙 규칙은 스냅샷 아카이빙을 활성화한 후 생성된 새 스냅샷에만 적용됩니다. 스냅샷 아카이빙을 활성화하기 전에 생성된 기존 스냅샷은 해당 스냅샷이 원래 생성되고 아카이브될 때 설정된 일정에 따라 해당 스토리지 계층에서 계속 삭제됩니다.

- 개수 기반 일정에 대해 스냅샷 아카이빙을 비활성화하면 일정이 즉시 스냅샷 아카이빙을 중지합니다. 일정에 따라 이전에 아카이브된 스냅샷은 아카이브 계층에 남아 있으며 Amazon Data Lifecycle Manager에 의해 삭제되지 않습니다.
- 기간 기반 일정에 대해 스냅샷 아카이빙을 비활성화하면 정책에 의해 생성되고 아카이브되도록 예약된 스냅샷은 `aws:dlm:expirationTime` 시스템 태그로 표시된 예약된 아카이브 날짜 및 시간에 영구적으로 삭제됩니다.
- 일정에 대해 스냅샷 아카이빙을 비활성화하면 일정이 즉시 스냅샷 아카이빙을 중지합니다. 일정에 따라 이전에 아카이브된 스냅샷은 아카이브 계층에 남아 있으며 Amazon Data Lifecycle Manager에 의해 삭제되지 않습니다.
- 개수 기반 일정의 아카이브 보존 횟수를 수정하면 새 보존 횟수에 이전에 해당 일정에 따라 아카이브된 기존 스냅샷이 포함됩니다.
- 기간 기반 일정의 아카이브 보존 기간을 수정하면 보존 규칙을 수정한 후 아카이브된 스냅샷에만 새 보존 기간이 적용됩니다.

다음 절차 중 하나를 사용하여 수명 주기 정책을 수정합니다.

Console

수명 주기 정책을 수정하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Elastic Block Store, Lifecycle Manager(수명 주기 관리자)를 선택합니다.
3. 목록에서 수명 주기 정책을 선택합니다.
4. 작업, 수명 주기 정책 수정을 선택합니다.
5. 필요에 따라 정책 설정을 수정합니다. 예를 들어 일정을 수정하거나, 태그를 추가 또는 제거하거나, 정책을 활성화 또는 비활성화할 수 있습니다.
6. 정책 수정을 선택합니다.

Command line

update-lifecycle-policy 명령을 사용하여 수명 주기 정책의 정보를 수정합니다. 구문을 간단히 하기 위해 이 예에서는 정책의 세부 정보가 들어 있는 JSON 파일(`policyDetailsUpdated.json`)을 참조합니다.

```
aws dlm update-lifecycle-policy \
```

```
    --state DISABLED \
    --execution-role-arn
 arn:aws:iam::12345678910:role/AWSDataLifecycleManagerDefaultRole" \
    --policy-details file://policyDetailsUpdated.json
```

다음은 policyDetailsUpdated.json 파일의 예입니다.

```
{
   "ResourceTypes":[
      "VOLUME"
   ],
   "TargetTags":[
      {
         "Key": "costcenter",
         "Value": "120"
      }
   ],
   "Schedules":[
      {
         "Name": "DailySnapshots",
         "TagsToAdd": [
            {
               "Key": "type",
               "Value": "myDailySnapshot"
            }
         ],
         "CreateRule": {
            "Interval": 12,
            "IntervalUnit": "HOURS",
            "Times": [
               "15:00"
            ]
         },
         "RetainRule": {
            "Count" :5
         },
         "CopyTags": false
      }
   ]
}
```

업데이트된 정책을 보려면 get-lifecycle-policy 명령을 사용하십시오. 상태, 태그 값, 스냅샷 간격, 스냅샷 시작 시간이 변경된 것을 알 수 있습니다.

Amazon Data Lifecycle Manager 정책 삭제

Amazon Data Lifecycle Manager 정책을 삭제할 때는 다음 사항을 유의해야 합니다.

- 정책을 삭제하는 경우 해당 정책에 의해 생성된 스냅샷 또는 AMI는 자동으로 삭제되지 않습니다. 스냅샷이나 AMI가 더 이상 필요하지 않은 경우에는 수동으로 삭제해야 합니다.
- 스냅샷 아카이빙이 활성화된 정책이 있는 정책을 삭제하면 정책 삭제 시 아카이브 계층에 있는 스냅샷은 더 이상 Amazon Data Lifecycle Manager Manager에서 관리되지 않습니다. 더 이상 필요하지 않은 스냅샷은 수동으로 삭제해야 합니다.
- 아카이브가 활성화된 기간 기반 일정이 있는 정책을 삭제하면 정책에 의해 생성되고 아카이브되도록 예약된 스냅샷은 `aws:dlm:expirationtime` 시스템 태그로 표시된 예약된 아카이브 날짜 및 시간에 영구적으로 삭제됩니다.

다음 절차 중 하나를 사용하여 수명 주기 정책을 삭제합니다.

Console

수명 주기 정책을 삭제하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Elastic Block Store, Lifecycle Manager(수명 주기 관리자)를 선택합니다.
3. 목록에서 수명 주기 정책을 선택합니다.
4. 작업, 수명 주기 정책 삭제를 선택합니다.
5. 확인 메시지가 나타나면 정책 삭제를 선택합니다.

Command line

delete-lifecycle-policy 명령을 사용하여 수명 주기 정책을 삭제하고, 정책에 지정된 대상 태그를 해제하여 재사용할 수 있도록 합니다.

> **Note**
>
> Amazon Data Lifecycle Manager에 의해 생성된 스냅샷만 삭제할 수 있습니다.

```
aws dlm delete-lifecycle-policy --policy-id policy-0123456789abcdef0
```

Amazon Data Lifecycle Manager API 참조에 Amazon Data Lifecycle Manager 쿼리 API의 데이터 유형과 각 작업에 대한 설명 및 구문이 나와 있습니다.

또는 AWS SDKs 중 하나를 사용하여 사용 중인 프로그래밍 언어 또는 플랫폼에 맞게 조정된 방식으로 API에 액세스할 수 있습니다. 자세한 내용은 AWS SDK를 참조하십시오.

IAM을 사용하여 Amazon Data Lifecycle Manager에 대한 액세스 제어

Amazon Data Lifecycle Manager에 액세스하려면 자격 증명이 필요합니다. 이러한 자격 증명에는 인스턴스, 볼륨, 스냅샷, AMI 등의 AWS 리소스에 액세스할 권한이 있어야 합니다.

Amazon Data Lifecycle Manager를 사용하려면 다음 IAM 권한이 필요합니다.

> **Note**
>
> - `ec2:DescribeAvailabilityZones`, `ec2:DescribeRegions`, `kms:ListAliases` 및 `kms:DescribeKey` 권한은 콘솔 사용자에게만 필요합니다. 콘솔 액세스가 필요하지 않은 경우 권한을 제거할 수 있습니다.
> - AWSDataLifecycleManagerDefaultRole의 ARN 형식은 콘솔을 사용하여 생성했는지 AWS CLI를 사용하여 생성했는지에 따라 다릅니다. 콘솔을 사용하여 역할을 생성한 경우 ARN 형식은 `arn:aws:iam::`*`account_id`*`:role/service-role/AWSDataLifecycleManagerDefaultRole`입니다. 를 사용하여 역할을 생성한 경우 AWS CLI ARN 형식은 입니다`arn:aws:iam::`*`account_id`*`:role/AWSDataLifecycleManagerDefaultRole`.

JSON

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

```
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": [
                "arn:aws:iam::111122223333:role/
AWSDataLifecycleManagerDefaultRole",
                "arn:aws:iam::111122223333:role/
AWSDataLifecycleManagerDefaultRoleForAMIManagement",
                "arn:aws:iam::111122223333:role/service-role/
AWSDataLifecycleManagerDefaultRole",
                "arn:aws:iam::111122223333:role/service-role/
AWSDataLifecycleManagerDefaultRoleForAMIManagement"
            ]
        },
        {
            "Effect": "Allow",
            "Action": "iam:ListRoles",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeAvailabilityZones",
                "ec2:DescribeRegions",
                "kms:ListAliases",
                "kms:DescribeKey"
            ],
            "Resource": "*"
        }
    ]
}
```

암호화 권한

Amazon Data Lifecycle Manager와 암호화된 리소스를 사용할 때는 다음 사항을 고려하세요.

- 소스 볼륨이 암호화된 경우 Amazon Data Lifecycle Manager 기본 역할 (AWSDataLifecycleManagerDefaultRole 및 AWSDataLifecycleManagerDefaultRoleForAMIManagement)에 볼륨을 암호화하는 데 사용되는 KMS 키를 사용할 수 있는 권한이 있는지 확인합니다.

- 암호화되지 않은 스냅샷 또는 암호화되지 않은 스냅샷이 지원하는 AMIs에 대해 교차 리전 복사를 활성화하고 대상 리전에서 암호화를 활성화하도록 선택한 경우 기본 역할에 대상 리전에서 암호화를 수행하는 데 필요한 KMS 키를 사용할 수 있는 권한이 있는지 확인합니다.
- 암호화된 스냅샷 또는 암호화된 스냅샷으로 지원되는 AMIs에 대해 리전 간 복사를 활성화하는 경우 기본 역할에 소스 및 대상 KMS 키를 모두 사용할 수 있는 권한이 있는지 확인합니다.
- 암호화된 스냅샷에 대해 스냅샷 아카이빙을 활성화하는 경우 Amazon Data Lifecycle Manager 기본 역할(AWSDataLifecycleManagerDefaultRole에는 스냅샷을 암호화하는 데 사용되는 KMS 키를 사용할 수 있는 권한이 있어야 합니다.

자세한 내용은 AWS Key Management Service 개발자 가이드의 다른 계정의 사용자가 CMK를 사용하도록 허용을 참조하세요.

자세한 내용은 IAM 사용 설명서의 사용자의 권한 변경을 참조하세요.

AWS Amazon Data Lifecycle Manager에 대한 관리형 정책

AWS 관리형 정책은에서 생성하고 관리하는 독립 실행형 정책입니다 AWS. AWS 관리형 정책은 많은 일반적인 사용 사례에 대한 권한을 제공하도록 설계되었습니다. AWS 관리형 정책은 정책을 직접 작성해야 하는 경우보다 사용자, 그룹 및 역할에 적절한 권한을 할당하는 것이 더 효율적입니다.

그러나 AWS 관리형 정책에 정의된 권한은 변경할 수 없습니다. AWS 는 때때로 AWS 관리형 정책에 정의된 권한을 업데이트합니다. 이 경우 정책이 연결되어 있는 모든 위탁자 엔터티(사용자, 그룹 및 역할)에도 업데이트가 적용됩니다.

Amazon Data Lifecycle Manager는 일반적인 사용 사례에 대한 AWS 관리형 정책을 제공합니다. 이러한 정책을 사용하면 리소스에 적절한 권한을 보다 효율적으로 정의하고 액세스를 제어할 수 있습니다. Amazon Data Lifecycle Manager에서 제공하는 AWS 관리형 정책은 Amazon Data Lifecycle Manager에 전달하는 역할에 연결되도록 설계되었습니다.

주제

- AWSDataLifecycleManagerServiceRole
- AWSDataLifecycleManagerServiceRoleForAMIManagement
- AWSDataLifecycleManagerSSMFullAccess
- AWS 관리형 정책 업데이트

AWSDataLifecycleManagerServiceRole

AWSDataLifecycleManagerServiceRole 정책은 Amazon EBS 스냅샷 정책 및 교차 계정 복사 이벤트 정책을 생성하고 관리할 수 있는 적절한 권한을 Amazon Data Lifecycle Manager에 제공합니다.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateSnapshot",
                "ec2:CreateSnapshots",
                "ec2:DeleteSnapshot",
                "ec2:DescribeInstances",
                "ec2:DescribeVolumes",
                "ec2:DescribeSnapshots",
                "ec2:EnableFastSnapshotRestores",
                "ec2:DescribeFastSnapshotRestores",
                "ec2:DisableFastSnapshotRestores",
                "ec2:CopySnapshot",
                "ec2:ModifySnapshotAttribute",
                "ec2:DescribeSnapshotAttribute",
                "ec2:ModifySnapshotTier",
                "ec2:DescribeSnapshotTierStatus",
                "ec2:DescribeAvailabilityZones"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateTags"
            ],
            "Resource": "arn:aws:ec2:*::snapshot/*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "events:PutRule",
                "events:DeleteRule",
```

```
                "events:DescribeRule",
                "events:EnableRule",
                "events:DisableRule",
                "events:ListTargetsByRule",
                "events:PutTargets",
                "events:RemoveTargets"
            ],
            "Resource": "arn:aws:events:*:*:rule/AwsDataLifecycleRule.managed-
cwe.*"
        }
    ]
}
```

AWSDataLifecycleManagerServiceRoleForAMIManagement

AWSDataLifecycleManagerServiceRoleForAMIManagement 정책은 Amazon EBS 지원 AMI 정책을 생성하고 관리할 수 있는 적절한 권한을 Amazon Data Lifecycle Manager에 제공합니다.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ec2:CreateTags",
            "Resource": [
                "arn:aws:ec2:*::snapshot/*",
                "arn:aws:ec2:*::image/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeImages",
                "ec2:DescribeInstances",
                "ec2:DescribeImageAttribute",
                "ec2:DescribeVolumes",
                "ec2:DescribeSnapshots"
            ],
            "Resource": "*"
```

```
        },
        {
            "Effect": "Allow",
            "Action": "ec2:DeleteSnapshot",
            "Resource": "arn:aws:ec2:*::snapshot/*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:ResetImageAttribute",
                "ec2:DeregisterImage",
                "ec2:CreateImage",
                "ec2:CopyImage",
                "ec2:ModifyImageAttribute"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:EnableImageDeprecation",
                "ec2:DisableImageDeprecation"
            ],
            "Resource": "arn:aws:ec2:*::image/*"
        }
    ]
}
```

AWSDataLifecycleManagerSSMFullAccess

모든 Amazon EC2 인스턴스에서 사전 및 사후 스크립트를 실행하는 데 필요한 Systems Manager 작업을 수행할 수 있는 Amazon Data Lifecycle Manager 권한을 제공합니다.

> **Important**
>
> 이 정책은 사전 및 사후 스크립트를 사용할 때 `aws:ResourceTag` 조건 키를 사용하여 특정 SSM 문서에 대한 액세스를 제한합니다. Amazon Data Lifecycle Manager가 SSM 문서에 액세스하도록 허용하려면 SSM 문서에 로 태그가 지정되어 있는지 확인해야 합니다`DLMScriptsAccess:true`.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowSSMReadOnlyAccess",
            "Effect": "Allow",
            "Action": [
                "ssm:GetCommandInvocation",
                "ssm:ListCommands",
                "ssm:DescribeInstanceInformation"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowTaggedSSMDocumentsOnly",
            "Effect": "Allow",
            "Action": [
                "ssm:SendCommand",
                "ssm:DescribeDocument",
                "ssm:GetDocument"
            ],
            "Resource": [
                "arn:aws:ssm:*:*:document/*"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/DLMScriptsAccess": "true"
                }
            }
        },
        {
            "Sid": "AllowSpecificAWSOwnedSSMDocuments",
            "Effect": "Allow",
            "Action": [
                "ssm:SendCommand",
                "ssm:DescribeDocument",
                "ssm:GetDocument"
            ],
            "Resource": [
                "arn:aws:ssm:*:*:document/AWSEC2-CreateVssSnapshot",
```

```
                "arn:aws:ssm:*:*:document/AWSSystemsManagerSAP-
CreateDLMSnapshotForSAPHANA"
            ]
        },
        {
            "Sid": "AllowAllEC2Instances",
            "Effect": "Allow",
            "Action": [
                "ssm:SendCommand"
            ],
            "Resource": [
                "arn:aws:ec2:*:*:instance/*"
            ]
        }
    ]
}
```

AWS 관리형 정책 업데이트

AWS 서비스는 AWS 관리형 정책을 유지 관리하고 업데이트합니다. AWS 관리형 정책에서는 권한을 변경할 수 없습니다. 서비스는 때때로 AWS 관리형 정책에 추가 권한을 추가하여 새 기능을 지원합니다. 이 유형의 업데이트는 정책이 연결된 모든 ID(사용자, 그룹 및 역할)에 적용됩니다. 서비스는 새 기능이 시작되거나 새 작업을 사용할 수 있게 될 때 AWS 관리형 정책을 업데이트할 가능성이 높습니다. 서비스는 AWS 관리형 정책에서 권한을 제거하지 않으므로 정책 업데이트로 인해 기존 권한이 손상되지 않습니다.

다음 표에는이 서비스가 이러한 변경 사항을 추적하기 시작한 이후부터 Amazon Data Lifecycle Manager의 AWS 관리형 정책 업데이트에 대한 세부 정보가 나와 있습니다. 이 페이지의 변경 사항에 대한 자동 알림을 받아보려면 Amazon EBS 사용 설명서에 대한 문서 이력에서 RSS 피드를 구독하세요.

변경	설명	Date
AWSDataLifecycleManagerServiceRole - 정책 권한을 업데이트했습니다.	Amazon Data Lifecycle Manager에 로컬 영역에 대한 정보를 가져올 수 있는 스냅샷 정	2024년 12월 16일

변경	설명	Date
	책 권한을 부여하는 `ec2:DescribeAvailabilityZones` 작업을 추가했습니다.	
AWSDataLifecycleManagerSSMFullAccess - 정책 권한을 업데이트했습니다.	`AWSSystemsManagerSAP-CreateDLMSnapshotForSAPHANA` SSM 문서를 사용하여 SAP HANA에 대해 애플리케이션에 일관되게 적용되는 스냅샷을 지원하도록 정책을 업데이트했습니다.	2023년 11월 17일
AWSDataLifecycleManagerSSMFullAccess - 새 AWS 관리형 정책을 추가했습니다.	Amazon Data Lifecycle Manager에 AWSDataLifecycleManagerSSMFullAccess AWS 관리형 정책이 추가되었습니다.	2023년 11월 7일

변경	설명	Date
AWSDataLifecycleManagerServiceRole - 스냅샷 아카이빙을 지원하는 권한이 추가되었습니다.	Amazon Data Lifecycle Manager는 스냅샷을 아카이브하고 스냅샷의 아카이브 상태를 확인할 수 있는 권한을 스냅샷 정책에 부여하기 위한 `ec2:ModifySnapshotTier` 및 `ec2:DescribeSnapshotTierStatus` 작업을 추가했습니다.	2022년 9월 30일
AWSDataLifecycleManagerServiceRoleForAMIManagement - AMI 사용 중단을 지원하기 위한 권한이 추가되었습니다.	EBS 지원 AMI 정책에 AMI 사용 중단을 활성화하거나 비활성화할 권한을 부여하는 `ec2:EnableImageDeprecation` 및 `ec2:DisableImageDeprecation` 작업이 Amazon Data Lifecycle Manager에 추가되었습니다.	2021년 8월 23일

변경	설명	Date
Amazon Data Lifecycle Manager에서 변경 내용 추적 시작	Amazon Data Lifecycle Manager가 AWS 관리형 정책에 대한 변경 사항 추적을 시작했습니다.	2021년 8월 23일

Amazon Data Lifecycle Manager를 위한 IAM 서비스 역할

AWS Identity and Access Management (IAM) 역할은 자격 AWS 증명이 할 수 있는 것과 없는 것을 결정하는 권한 정책이 있는 자격 증명이라는 점에서 사용자와 유사합니다 AWS. 그러나 역할은 한 사람하고만 연관되지 않고 해당 역할이 필요한 사람이라면 누구든지 맡을 수 있어야 합니다. 서비스 역할은 AWS 서비스가 사용자를 대신하여 작업을 수행하기 위해 수임하는 역할입니다. 사용자 대신 백업 작업을 수행하는 서비스인 Amazon Data Lifecycle Manager에 사용자 대신 정책 작업을 수행할 때 맡아야 할 역할을 전달해야 합니다. IAM 역할에 대한 자세한 내용은 IAM 사용 설명서의 IAM 역할 섹션을 참조하세요.

Amazon Data Lifecycle Manager에 전달하는 역할에는 Amazon Data Lifecycle Manager가 스냅샷 및 AMI 생성, 스냅샷 및 AMI 복사, 스냅샷 삭제, AMI 등록 취소 등 정책 작업과 관련된 작업을 수행할 수 있는 권한이 부여된 IAM 정책이 있어야 합니다. Amazon Data Lifecycle Manager 정책 유형마다 서로 다른 권한이 필요합니다. 또한 Amazon Data Lifecycle Manager가 신뢰할 수 있는 엔터티로 역할에 나열되어 있어야 Amazon Data Lifecycle Manager가 해당 역할을 맡을 수 있습니다.

주제

- Amazon Data Lifecycle Manager를 위한 기본 서비스 역할
- Amazon Data Lifecycle Manager를 위한 사용자 지정 서비스 역할

Amazon Data Lifecycle Manager를 위한 기본 서비스 역할

Amazon Data Lifecycle Manager는 다음 기본 서비스 역할을 사용합니다.

- AWSDataLifecycleManagerDefaultRole—스냅샷 관리를 위한 기본 역할입니다. 역할을 수임할 대상으로 `dlm.amazonaws.com` 서비스만 신뢰하며, Amazon Data Lifecycle Manager가 사용자를

대신하여 스냅샷 및 교차 계정 스냅샷 복사 정책에 필요한 작업을 수행하도록 합니다. 이 역할은 `AWSDataLifecycleManagerServiceRole` AWS 관리형 정책을 사용합니다.

> **Note**
>
> 역할의 ARN 형식은 콘솔을 사용하여 생성했는지 AWS CLI를 사용하여 생성했는지에 따라 다릅니다. 콘솔을 사용하여 역할을 생성한 경우 ARN 형식은 `arn:aws:iam::account_id:role/service-role/AWSDataLifecycleManagerDefaultRole`입니다. 를 사용하여 역할을 생성한 경우 AWS CLI ARN 형식은 입니다`arn:aws:iam::account_id:role/AWSDataLifecycleManagerDefaultRole`.

- AWSDataLifecycleManagerDefaultRoleForAMIManagement—AMI 관리를 위한 기본 역할입니다. 역할을 수임할 대상으로 `dlm.amazonaws.com` 서비스만 신뢰하며, Amazon Data Lifecycle Manager가 사용자를 대신하여 EBS 지원 AMI 정책에 필요한 작업을 수행하도록 합니다. 이 역할은 `AWSDataLifecycleManagerServiceRoleForAMIManagement` AWS 관리형 정책을 사용합니다.

Amazon Data Lifecycle Manager 콘솔을 사용할 경우, 스냅샷 또는 교차 계정 스냅샷 복사 정책을 처음 생성할 때 Amazon Data Lifecycle Manager가 AWSDataLifecycleManagerDefaultRole 서비스 역할을 자동으로 생성하며, EBS 지원 AMI 정책을 처음 생성할 때 AWSDataLifecycleManagerDefaultRoleForAMIManagement 서비스 역할을 자동으로 생성합니다.

콘솔을 사용하지 않을 경우 create-default-role 명령을 사용하여 서비스 역할을 수동으로 생성할 수 있습니다. `--resource-type`에 `snapshot`을 지정하여 AWSDataLifecycleManagerDefaultRole을 생성하거나 `image`를 지정하여 AWSDataLifecycleManagerDefaultRoleForAMIManagement를 생성합니다.

```
$ aws dlm create-default-role --resource-type snapshot|image
```

기본 서비스 역할을 삭제한 다음 다시 생성해야 하는 경우 동일한 프로세스를 사용하여 계정에서 역할을 다시 생성할 수 있습니다.

Amazon Data Lifecycle Manager를 위한 사용자 지정 서비스 역할

기본 서비스 역할을 사용하는 것에 대한 대안으로, 필요한 권한이 있는 사용자 지정 IAM 역할을 생성한 다음, 수명 주기 정책을 생성할 때 이를 선택해도 됩니다.

사용자 지정 IAM 역할을 생성하려면

1. 다음 권한을 가진 역할을 생성합니다.

   - 스냅샷 수명 주기 정책을 관리하는 데 필요한 권한

     JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateSnapshot",
                "ec2:CreateSnapshots",
                "ec2:DeleteSnapshot",
                "ec2:DescribeInstances",
                "ec2:DescribeVolumes",
                "ec2:DescribeSnapshots",
                "ec2:EnableFastSnapshotRestores",
                "ec2:DescribeFastSnapshotRestores",
                "ec2:DisableFastSnapshotRestores",
                "ec2:CopySnapshot",
                "ec2:ModifySnapshotAttribute",
                "ec2:DescribeSnapshotAttribute",
                "ec2:ModifySnapshotTier",
                "ec2:DescribeSnapshotTierStatus",
                "ec2:DescribeAvailabilityZones"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateTags"
            ],
            "Resource": "arn:aws:ec2:*::snapshot/*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "events:PutRule",
                "events:DeleteRule",
```

```
                "events:DescribeRule",
                "events:EnableRule",
                "events:DisableRule",
                "events:ListTargetsByRule",
                "events:PutTargets",
                "events:RemoveTargets"
            ],
            "Resource": "arn:aws:events:*:*:rule/
AwsDataLifecycleRule.managed-cwe.*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ssm:GetCommandInvocation",
                "ssm:ListCommands",
                "ssm:DescribeInstanceInformation"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ssm:SendCommand",
                "ssm:DescribeDocument",
                "ssm:GetDocument"
            ],
            "Resource": [
                "arn:aws:ssm:*:*:document/*"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/DLMScriptsAccess": "true"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ssm:SendCommand",
                "ssm:DescribeDocument",
                "ssm:GetDocument"
            ],
            "Resource": [
                "arn:aws:ssm:*::document/*"
```

```
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "ssm:SendCommand"
            ],
            "Resource": [
                "arn:aws:ec2:*:*:instance/*"
            ],
            "Condition": {
                "StringNotLike": {
                    "aws:ResourceTag/DLMScriptsAccess": "false"
                }
            }
        }
    ]
}
```

- AMI 수명 주기 정책을 관리하는 데 필요한 권한

  JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ec2:CreateTags",
            "Resource": [
                "arn:aws:ec2:*::snapshot/*",
                "arn:aws:ec2:*::image/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeImages",
                "ec2:DescribeInstances",
                "ec2:DescribeImageAttribute",
                "ec2:DescribeVolumes",
                "ec2:DescribeSnapshots"
            ],
            "Resource": "*"
```

```
        },
        {
            "Effect": "Allow",
            "Action": "ec2:DeleteSnapshot",
            "Resource": "arn:aws:ec2:*::snapshot/*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:ResetImageAttribute",
                "ec2:DeregisterImage",
                "ec2:CreateImage",
                "ec2:CopyImage",
                "ec2:ModifyImageAttribute"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:EnableImageDeprecation",
                "ec2:DisableImageDeprecation"
            ],
            "Resource": "arn:aws:ec2:*::image/*"
        }
    ]
}
```

자세한 내용은 IAM 사용 설명서의 역할 생성을 참조하세요.

2. 역할에 신뢰 관계를 추가합니다.

   a. IAM 콘솔에서 역할을 선택합니다.

   b. 생성한 역할을 선택하고 신뢰 관계(Trust relationships)를 선택합니다.

   c. 신뢰 관계 편집을 선택하고 다음 정책을 추가한 뒤 신뢰 정책 업데이트를 선택합니다.

   JSON

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

```
  "Effect": "Allow",
  "Principal": {
   "Service": "dlm.amazonaws.com"
  },
  "Action": "sts:AssumeRole"
 }]
}
```

혼동된 대리자 문제로부터 자신을 보호하기 위하여 aws:SourceAccount 및 aws:SourceArn 조건 키를 사용할 것을 권장합니다. 예를 들어 이전 신뢰 정책에 다음 조건 블록을 추가할 수 있습니다. aws:SourceAccount는 수명 주기 정책의 소유자이고, aws:SourceArn은 수명 주기 정책의 ARN입니다. 수명 주기 정책 ID를 모르는 경우 ARN의 해당 부분을 와일드카드(*)로 바꾼 다음 수명 주기 정책을 만든 후 신뢰 정책을 업데이트할 수 있습니다.

```
"Condition": {
    "StringEquals": {
        "aws:SourceAccount": "account_id"
    },
    "ArnLike": {
        "aws:SourceArn": "arn:partition:dlm:region:account_id:policy/policy_id"
    }
}
```

Amazon Data Lifecycle Manager 정책 모니터링

다음 기능을 사용하여 스냅샷 및 AMI의 수명 주기를 모니터링할 수 있습니다.

특성

- 콘솔 및 AWS CLI
- EventBridge를 사용하여 Data Lifecycle Manager 정책 모니터링
- CloudWatch를 사용하여 Data Lifecycle Manager 정책 모니터링
- 를 사용하여 Amazon Data Lifecycle Manager API 호출 로깅 AWS CloudTrail

콘솔 및 AWS CLI

Amazon EC2 콘솔 또는 AWS CLI를 사용하여 수명 주기 정책을 볼 수 있습니다. 정책에 따라 생성된 각 스냅샷 및 AMI에는 타임스탬프 및 해당 정책과 관련된 태그가 있습니다. 이 태그로 스냅샷 및 AMI를 필터링하여 백업이 의도대로 생성되고 있는지 확인할 수 있습니다.

EventBridge를 사용하여 Data Lifecycle Manager 정책 모니터링

Amazon EBS 및 Amazon Data Lifecycle Manager는 수명 주기 정책 작업과 관련된 이벤트를 발생시킵니다. AWS Lambda 및 Amazon CloudWatch Events를 사용하여 이벤트 알림을 프로그래밍 방식으로 처리할 수 있습니다. 이벤트는 최선의 작업을 기반으로 발생됩니다. 자세한 내용은 Amazon EventBridge 사용 설명서를 참조하세요.

다음과 같은 이벤트를 사용할 수 있습니다.

> **Note**
>
> AMI 수명 주기 정책 작업의 경우 이벤트가 생성되지 않습니다.

- `createSnapshot` - `CreateSnapshot` 작업이 성공하거나 실패할 때 발생하는 Amazon EBS 이벤트입니다. 자세한 내용은 Amazon EBS용 Amazon EventBridge 이벤트 단원을 참조하십시오.
- `DLM Policy State Change` - 수명 주기 정책이 오류 상태가 될 때 발생하는 Amazon Data Lifecycle Manager 이벤트입니다. 이 이벤트에는 오류의 원인에 대한 설명이 포함되어 있습니다.

  다음은 IAM 역할에서 부여한 권한이 충분하지 않을 때 나타나는 이벤트의 예입니다.

```
{
    "version": "0",
    "id": "01234567-0123-0123-0123-0123456789ab",
    "detail-type": "DLM Policy State Change",
    source": "aws.dlm",
    "account": "123456789012",
    "time": "2018-05-25T13:12:22Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:dlm:us-east-1:123456789012:policy/policy-0123456789abcdef"
    ],
    "detail": {
        "state": "ERROR",
```

```
        "cause": "Role provided does not have sufficient permissions",
        "policy_id": "arn:aws:dlm:us-east-1:123456789012:policy/
policy-0123456789abcdef"
    }
}
```

다음은 한도를 초과할 때 발생하는 이벤트의 예입니다.

```
{
    "version": "0",
    "id": "01234567-0123-0123-0123-0123456789ab",
    "detail-type": "DLM Policy State Change",
    "source": "aws.dlm",
    "account": "123456789012",
    "time": "2018-05-25T13:12:22Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:dlm:us-east-1:123456789012:policy/policy-0123456789abcdef"
    ],
    "detail":{
        "state": "ERROR",
        "cause": "Maximum allowed active snapshot limit exceeded",
        "policy_id": "arn:aws:dlm:us-east-1:123456789012:policy/
policy-0123456789abcdef"
    }
}
```

- `DLM Pre Post Script Notification` - 사전 또는 사후 스크립트가 시작되거나 성공하거나 실패할 때 발생하는 이벤트입니다.

  다음은 VSS 백업이 성공한 경우의 예시 이벤트입니다.

```
{
    "version": "0",
    "id": "12345678-1234-1234-1234-123456789012",
    "detail-type": "DLM Pre Post Script Notification",
    "source": "aws.dlm",
    "account": "123456789012",
    "time": "2023-10-27T22:04:52Z",
    "region": "us-east-1",
    "resources": ["arn:aws:dlm:us-east-1:123456789012:policy/
policy-01234567890abcdef"],
    "detail": {
```

```
        "script_stage": "",
        "result": "success",
        "cause": "",
        "policy_id": "arn:aws:dlm:us-east-1:123456789012:policy/
policy-01234567890abcdef",
        "execution_handler": "AWS_VSS_BACKUP",
        "source": "arn:aws:ec2:us-east-1:123456789012:instance/i-01234567890abcdef",
        "resource_type": "EBS_SNAPSHOT",
        "resources": [{
            "status": "pending",
            "resource_id": "arn:aws:ec2:us-east-1::snapshot/snap-01234567890abcdef",
            "source": "arn:aws:ec2:us-east-1:123456789012:volume/
vol-01234567890abcdef"
        }],
        "request_id": "a1b2c3d4-a1b2-a1b2-a1b2-a1b2c3d4e5f6",
        "start_time": "2023-10-27T22:03:29.370Z",
        "end_time": "2023-10-27T22:04:51.370Z",
        "timeout_time": ""
    }
}
```


CloudWatch를 사용하여 Data Lifecycle Manager 정책 모니터링

원시 데이터를 수집하여 실시간에 가까운 읽기 쉬운 지표로 처리하는 CloudWatch를 사용하여 Amazon Data Lifecycle Manager 수명 주기 정책을 모니터링할 수 있습니다. 이러한 지표를 사용하여 정책 및 시간 경과에 따라 생성, 삭제 및 복사되는 Amazon EBS 스냅샷 개수 및 EBS 지원 AMI 개수를 정확히 확인할 수 있습니다. 특정 임계값을 주시하다가 해당 임계값이 충족될 때 알림을 전송하거나 조치를 취하도록 경보를 설정할 수도 있습니다.

지표는 15개월 동안 보관되므로 기간별 정보에 액세스하고 수명 주기 정책이 장기간 어떻게 실행되는지 더 잘 이해할 수 있습니다.

Amazon CloudWatch에 대한 자세한 내용은 Amazon CloudWatch 사용 설명서를 참조하십시오.

주제

- 지원되는 지표
- 정책에 대한 CloudWatch 지표 보기
- 정책에 대한 그래프 지표
- 정책에 대한 CloudWatch 경보 만들기

- 사용 사례 예시
- 실패한 작업을 보고하는 정책 관리

지원되는 지표

AWS/EBS 네임스페이스에는 다음과 같은 Amazon Data Lifecycle Manager 지표가 포함됩니다. 지표는 정책 유형에 따라 다릅니다.

모든 지표는 DLMPolicyId 차원에 따라 측정됩니다. 가장 유용한 통계는 sum 및 average이고 측정 단위는 count입니다.

탭을 선택하여 해당 정책 유형에서 지원하는 지표를 봅니다.

EBS snapshot policies

지표	설명
Resources Targeted	스냅샷 또는 EBS 지원 AMI 정책에서 지정한 태그의 대상이 되는 리소스 수입니다.
Snapshots CreateStarted	스냅샷 정책을 통해 시작한 스냅샷 생성 작업의 수입니다. 재시도를 여러 번 연속으로 수행한 경우에도 각 작업은 한 번만 기록됩니다. 스냅샷 생성 작업이 실패할 경우 Amazon Data Lifecycle Manager는 SnapshotsCreateFailed 지표를 전송합니다.
Snapshots CreateCompleted	스냅샷 정책에서 생성한 스냅샷 수입니다. 여기에는 예약된 시간으로부터 60분 이내에 성공한 재시도를 포함합니다.
Snapshots CreateFailed	스냅샷 정책에서 생성할 수 없는 스냅샷 수입니다. 여기에는 예약된 시간으로부터 60분 이내에 실패한 재시도를 포함합니다.
Snapshots SharedCompleted	스냅샷 정책을 통해 계정 간에 공유되는 스냅샷 수입니다.

지표	설명
SnapshotsDeleteCompleted	스냅샷 또는 EBS 지원 AMI 정책을 통해 삭제한 스냅샷 수입니다. 이 지표는 정책을 통해 생성한 스냅샷에만 적용됩니다. 정책을 통해 생성한 교차 리전 스냅샷 복사본에는 적용되지 않습니다. 이 지표에는 EBS 지원 AMI 정책이 AMI를 등록 취소할 때 삭제되는 스냅샷을 포함합니다.
SnapshotsDeleteFailed	스냅샷 또는 EBS 지원 AMI 정책을 통해 삭제할 수 없는 스냅샷 수입니다. 이 지표는 정책을 통해 생성한 스냅샷에만 적용됩니다. 정책을 통해 생성한 교차 리전 스냅샷 복사본에는 적용되지 않습니다. 이 지표에는 EBS 지원 AMI 정책이 AMI를 등록 취소할 때 삭제되는 스냅샷을 포함합니다.
SnapshotsCopiedRegionStarted	스냅샷 정책을 통해 시작한 교차 리전 스냅샷 복사 작업의 수입니다.
SnapshotsCopiedRegionCompleted	스냅샷 정책을 통해 생성한 교차 리전 스냅샷 복사본 수입니다. 여기에는 예약된 시간으로부터 24시간 이내에 성공한 재시도를 포함합니다.
SnapshotsCopiedRegionFailed	스냅샷 정책을 통해 생성할 수 없는 교차 리전 스냅샷 복사본 수입니다. 여기에는 예약된 시간으로부터 24시간 이내에 실패한 재시도를 포함합니다.
SnapshotsCopiedRegionDeleteCompleted	보존 규칙에서 지정한 대로 스냅샷 정책을 통해 삭제한 교차 리전 스냅샷 복사본의 수입니다.
SnapshotsCopiedRegionDeleteFailed	보존 규칙에서 지정한 대로 스냅샷 정책을 통해 삭제할 수 없는 교차 리전 스냅샷 복사본의 수입니다.

지표	설명
snapshots ArchiveDeletionFailed	스냅샷 정책에 따라 아카이브 티어에서 삭제할 수 없었던 아카이빙된 스냅샷 수입니다.
snapshots ArchiveScheduled	스냅샷 정책에 따라 아카이빙이 예약된 스냅샷 수입니다.
snapshots ArchiveCompleted	스냅샷 정책에 따라 아카이빙된 스냅샷 수입니다.
snapshots ArchiveFailed	스냅샷 정책 따라 아카이빙할 수 없었던 스냅샷 수입니다.
snapshots ArchiveDeletionCompleted	스냅샷 정책에 따라 아카이브 티어에서 삭제된 아카이빙된 스냅샷 수입니다.
PreScriptStarted	사전 스크립트가 성공적으로 시작된 인스턴스 수입니다. 스크립트 재시도가 활성화된 경우 정책 실행당 이 지표가 여러 번 내보내질 수 있습니다.
PreScriptCompleted	사전 스크립트가 성공적으로 완료된 인스턴스 수입니다. 사전 스크립트가 지정된 제한 시간을 벗어나 완료되는 경우에도 지표가 내보내집니다. 스크립트 재시도가 활성화된 경우 정책 실행당 이 지표가 여러 번 내보내질 수 있습니다.

지표	설명
PreScript Failed	사전 스크립트가 성공적으로 완료되지 못한 인스턴스 수입니다. 사전 스크립트가 지정된 제한 시간을 벗어나 완료되는 경우에도 지표가 내보내집니다. 스크립트 재시도가 활성화된 경우 정책 실행당 이 지표가 여러 번 내보내질 수 있습니다.
PostScript tStarted	사후 스크립트가 성공적으로 시작된 인스턴스 수입니다. 스크립트 재시도가 활성화된 경우 정책 실행당 이 지표가 여러 번 내보내질 수 있습니다.
PostScriptCompleted	사후 스크립트가 성공적으로 완료된 인스턴스 수입니다. 사후 스크립트가 지정된 제한 시간을 벗어나 완료되는 경우에도 지표가 내보내집니다. 스크립트 재시도가 활성화된 경우 정책 실행당 이 지표가 여러 번 내보내질 수 있습니다.
PostScriptFailed	사후 스크립트가 성공적으로 완료되지 못한 인스턴스 수입니다. 사후 스크립트가 지정된 제한 시간을 벗어나 완료되는 경우에도 지표가 내보내집니다. 스크립트 재시도가 활성화된 경우 정책 실행당 이 지표가 여러 번 내보내질 수 있습니다.
VSSBackup Started	VSS 백업이 성공적으로 시작된 인스턴스 수입니다. 스크립트 재시도가 활성화된 경우 정책 실행당 이 지표가 여러 번 내보내질 수 있습니다.
VSSBackup Completed	VSS 백업이 성공적으로 완료된 인스턴스 수입니다. VSS 백업이 제한 시간을 벗어나 완료되는 경우에도 지표가 내보내집니다. 스크립트 재시도가 활성화된 경우 정책 실행당 이 지표가 여러 번 내보내질 수 있습니다.

지표	설명
VSSBackup Failed	VSS 백업이 성공적으로 완료되지 못한 인스턴스 수입니다. VSS 백업이 제한 시간을 벗어나 완료되는 경우에도 지표가 내보내집니다. 스크립트 재시도가 활성화된 경우 정책 실행당 이 지표가 여러 번 내보내질 수 있습니다.

EBS-backed AMI policies

EBS 지원 AMI 정책과 함께 사용할 수 있는 지표는 다음과 같습니다.

지표	설명
Resources Targeted	스냅샷 또는 EBS 지원 AMI 정책에서 지정한 태그의 대상이 되는 리소스 수입니다.
Snapshots DeleteCompleted	스냅샷 또는 EBS 지원 AMI 정책을 통해 삭제한 스냅샷 수입니다. 이 지표는 정책을 통해 생성한 스냅샷에만 적용됩니다. 정책을 통해 생성한 교차 리전 스냅샷 복사본에는 적용되지 않습니다. 이 지표에는 EBS 지원 AMI 정책이 AMI를 등록 취소할 때 삭제되는 스냅샷을 포함합니다.
Snapshots DeleteFailed	스냅샷 또는 EBS 지원 AMI 정책을 통해 삭제할 수 없는 스냅샷 수입니다. 이 지표는 정책을 통해 생성한 스냅샷에만 적용됩니다. 정책을 통해 생성한 교차 리전 스냅샷 복사본에는 적용되지 않습니다. 이 지표에는 EBS 지원 AMI 정책이 AMI를 등록 취소할 때 삭제되는 스냅샷을 포함합니다.
Snapshots CopiedRegionDelete Completed	보존 규칙에서 지정한 대로 스냅샷 정책을 통해 삭제한 교차 리전 스냅샷 복사본의 수입니다.

지표	설명
SnapshotsCopiedRegionDeleteFailed	보존 규칙에서 지정한 대로 스냅샷 정책을 통해 삭제할 수 없는 교차 리전 스냅샷 복사본의 수입니다.
ImagesCreateStarted	EBS 지원 AMI 정책을 통해 시작한 CreateImage 작업 수입니다.
ImagesCreateCompleted	EBS 지원 AMI 정책을 통해 생성한 AMI 수입니다.
ImagesCreateFailed	EBS 지원 AMI 정책을 통해 생성할 수 없는 AMI 수입니다.
ImagesDeregisterCompleted	EBS 지원 AMI 정책을 통해 등록 취소된 AMI 수입니다.
ImagesDeregisterFailed	EBS 지원 AMI 정책을 통해 등록 취소할 수 없는 AMI 수입니다.
ImagesCopiedRegionStarted	EBS 지원 AMI 정책을 통해 시작한 교차 리전 복사 작업의 수입니다.
ImagesCopiedRegionCompleted	EBS 지원 AMI 정책을 통해 생성한 교차 리전 AMI 복사본 수입니다.

지표	설명
ImagesCopiedRegionFailed	EBS 지원 AMI 정책을 통해 생성할 수 없는 교차 리전 AMI 복사본 수입니다.
ImagesCopiedRegionDeregisterCompleted	보존 규칙에서 지정한 대로 EBS 지원 AMI 정책을 통해 등록 취소된 교차 리전 AMI 복사본의 수입니다.
ImagesCopiedRegionDeregisteredFailed	보존 규칙에서 지정한 대로 EBS 지원 AMI 정책을 통해 등록 취소할 수 없는 교차 리전 AMI 복사본의 수입니다.
EnableImageDeprecationCompleted	EBS 지원 AMI 정책에 의해 사용 중단으로 표시된 AMI 수입니다.
EnableImageDeprecationFailed	EBS 지원 AMI 정책에 의해 사용 중단으로 표시될 수 없는 AMI 수입니다.
EnableCopiedImageDeprecationCompleted	EBS 지원 AMI 정책에 의해 사용 중단으로 표시된 교차 리전 AMI 사본의 수입니다.
EnableCopiedImageDeprecationFailed	EBS 지원 AMI 정책에 의해 사용 중단으로 표시될 수 없는 교차 리전 AMI 사본의 수입니다.

Cross-account copy event policies

다음 지표는 교차 계정 복사 이벤트 정책과 함께 사용할 수 있습니다.

지표	설명
`SnapshotsCopiedAccountStarted`	교차 계정 복사 이벤트 정책을 통해 시작한 교차 계정 스냅샷 복사 작업의 수입니다.
`SnapshotsCopiedAccountCompleted`	교차 계정 복사 이벤트 정책을 통해 다른 계정에서 복사된 스냅샷 수입니다. 여기에는 예약된 시간으로부터 24시간 이내에 성공한 재시도를 포함합니다.
`SnapshotsCopiedAccountFailed`	교차 계정 복사 이벤트 정책을 통해 다른 계정에서 복사할 수 없는 스냅샷 수입니다. 여기에는 예약된 시간으로부터 24시간 이내에 실패한 재시도를 포함합니다.
`SnapshotsCopiedAccountDeleteCompleted`	교차 계정 복사 이벤트 정책을 통해 보존 규칙에서 지정된 대로 삭제한 교차 리전 스냅샷 복사본의 수입니다.
`SnapshotsCopiedAccountDeleteFailed`	교차 계정 복사 이벤트 정책을 통해 보존 규칙에서 지정된 대로 삭제할 수 없는 교차 리전 스냅샷 복사본의 수입니다.

정책에 대한 CloudWatch 지표 보기

AWS Management Console 또는 명령줄 도구를 사용하여 Amazon Data Lifecycle Manager가 Amazon CloudWatch로 보내는 지표를 나열할 수 있습니다.

Amazon EC2 console

Amazon EC2 콘솔을 사용하여 지표를 보려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 수명 주기 관리자(Lifecycle Manager)를 선택합니다.
3. 그리드에서 정책을 선택한 다음 모니터링(Monitoring) 탭을 선택합니다.

CloudWatch console

Amazon CloudWatch 콘솔을 사용한 지표 확인

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 지표를 선택합니다.
3. EBS 네임스페이스를 선택한 다음 [Data Lifecycle Manager 지표(Data Lifecycle Manager metrics)]를 선택합니다.

AWS CLI

Amazon Data Lifecycle Manager에 대해 사용 가능한 모든 지표를 나열하려면

list-metrics 명령을 사용합니다.

```
$ C:\> aws cloudwatch list-metrics \
    --namespace AWS/EBS
```

특정 정책에 대한 모든 지표를 나열하려면

list-metrics 명령을 사용하여 `DLMPolicyId` 차원을 지정합니다.

```
$ C:\> aws cloudwatch list-metrics \
    --namespace AWS/EBS \
    --dimensions Name=DLMPolicyId,Value=policy-abcdef01234567890
```

모든 정책에서 단일 지표를 나열하려면

list-metrics 명령을 사용하여 `--metric-name` 옵션을 지정합니다.

```
$ C:\> aws cloudwatch list-metrics \
    --namespace AWS/EBS \
```

```
    --metric-name SnapshotsCreateCompleted
```

정책에 대한 그래프 지표

정책을 생성한 후 Amazon EC2 콘솔을 열고 모니터링(Monitoring) 탭에서 정책에 대한 모니터링 그래프를 볼 수 있습니다. 각 그래프는 사용 가능한 Amazon EC2 측정치 중 하나를 기반으로 합니다.

다음과 같은 그래프 지표를 사용할 수 있습니다.

- 대상 리소스(ResourcesTargeted 기준)
- 스냅샷 생성이 시작됨(SnapshotsCreateStarted 기준)
- 스냅샷 생성이 완료됨(SnapshotsCreateCompleted 기준)
- 스냅샷 생성 실패(SnapshotsCreateFailed 기준)
- 스냅샷 공유가 완료됨(SnapshotsSharedCompleted 기준)
- 스냅샷 삭제가 완료됨(SnapshotsDeleteCompleted 기준)
- 스냅샷 삭제 실패(SnapshotsDeleteFailed 기준)
- 스냅샷 교차 리전 복사가 시작됨(SnapshotsCopiedRegionStarted 기준)
- 스냅샷 교차 리전 복사가 완료됨(SnapshotsCopiedRegionCompleted 기준)
- 스냅샷 교차 리전 복사 실패(SnapshotsCopiedRegionFailed 기준)
- 스냅샷 교처 리전 사본 삭제가 완료됨(SnapshotsCopiedRegionDeleteCompleted 기준)
- 스냅샷 교차 리전 사본 삭제 실패(SnapshotsCopiedRegionDeleteFailed 기준)
- 스냅샷 교차 계정 복사가 시작됨(SnapshotsCopiedAccountStarted 기준)
- 스냅샷 교차 계정 복사가 완료됨(SnapshotsCopiedAccountCompleted 기준)
- 스냅샷 교차 계정 복사 실패(SnapshotsCopiedAccountFailed 기준)
- 스냅샷 교차 계정 사본 삭제가 완료됨(SnapshotsCopiedAccountDeleteCompleted 기준)
- 스냅샷 교차 계정 사본 삭제 실패(SnapshotsCopiedAccountDeleteFailed 기준)
- AMI 생성이 시작됨(ImagesCreateStarted 기준)
- AMI 생성이 완료됨(ImagesCreateCompleted 기준)
- AMI 생성 실패(ImagesCreateFailed 기준)
- AMI 등록 취소가 완료됨(ImagesDeregisterCompleted 기준)
- AMI 등록 취소 실패(ImagesDeregisterFailed 기준)
- AMI 교차 리전 복사가 시작됨(ImagesCopiedRegionStarted 기준)

- AMI 교차 리전 복사가 완료됨(ImagesCopiedRegionCompleted 기준)
- AMI 교차 리전 복사 실패(ImagesCopiedRegionFailed 기준)
- AMI 교차 리전 사본 등록 취소가 완료됨(ImagesCopiedRegionDeregisterCompleted 기준)
- AMI 교차 리전 사본 등록 취소 실패(ImagesCopiedRegionDeregisteredFailed 기준)
- AMI 사용 중단 활성화가 완료됨(EnableImageDeprecationCompleted 기준)
- AMI 사용 중단 활성화 실패(EnableImageDeprecationFailed 기준)
- AMI 교차 리전 사본 사용 중단 활성화가 완료됨(EnableCopiedImageDeprecationCompleted 기준)
- AMI 교차 리전 사본 사용 중단 활성화 실패(EnableCopiedImageDeprecationFailed 기준)

정책에 대한 CloudWatch 경보 만들기

정책에 대한 CloudWatch 지표를 모니터링하는 CloudWatch 경보를 생성할 수 있습니다. 지표가 지정한 임계값에 도달하면 CloudWatch에서 자동으로 알림을 보냅니다. CloudWatch 콘솔을 이용하여 CloudWatch 경보를 생성할 수 있습니다.

CloudWatch 콘솔을 사용하여 경보를 생성하는 방법에 대한 정보는 Amazon CloudWatch 사용 설명서에서 다음 주제를 참조하세요.

- 정적 임계점을 기반으로 CloudWatch 경보 생성
- 이상 탐지를 기반으로 CloudWatch 경보 생성

사용 사례 예시

다음은 사용 사례의 예입니다.

주제

- 예제 1: ResourcesTargeted 지표
- 예제 2: SnapshotDeleteFailed 지표
- 예제 3: SnapshotsCopiedRegionFailed 지표

예제 1: ResourcesTargeted 지표

ResourcesTargeted 지표를 사용하여 특정 정책이 실행될 때마다 대상이 되는 총 리소스 수를 모니터링할 수 있습니다. 이렇게 하면 대상 리소스 수가 예상 임계값보다 작거나 초과할 때 경보를 트리거할 수 있습니다.

예를 들어 일일 정책에 따라 50 볼륨 이하의 백업을 생성할 것으로 예상하는 경우, ResourcesTargeted에 대한 sum이 1 시간의 기간에 대해 50보다 클 때 이메일 알림을 보내는 경보를 생성할 수 있습니다. 이렇게 하면 잘못 태그가 지정된 볼륨에서 예기치 않게 생성된 스냅샷이 없는지 확인할 수 있습니다.

다음 명령을 사용하여 이 경보를 생성할 수 있습니다.

```
$ C:\> aws cloudwatch put-metric-alarm \
    --alarm-name resource-targeted-monitor \
    --alarm-description "Alarm when policy targets more than 50 resources" \
    --metric-name ResourcesTargeted \
    --namespace AWS/EBS \
    --statistic Sum \
    --period 3600 \
    --threshold 50 \
    --comparison-operator GreaterThanThreshold \
    --dimensions "Name=DLMPolicyId,Value=policy_id" \
    --evaluation-periods 1 \
    --alarm-actions sns_topic_arn
```

예제 2: SnapshotDeleteFailed 지표

SnapshotDeleteFailed 지표를 사용하여 정책의 스냅샷 보존 규칙에 따라 스냅샷 삭제 실패를 모니터링할 수 있습니다.

예를 들어, 스냅샷을 12시간마다 자동으로 삭제하는 정책을 작성한 경우, SnapshotDeletionFailed의 sum이 1 시간의 기간에 대해 0보다 클 때 엔지니어링 팀에 알리는 경보를 만들 수 있습니다. 이렇게 하면 부적절한 스냅샷 보존을 조사하고 불필요한 스냅샷으로 인해 스토리지 비용이 증가하지 않도록 하는 데 도움이 될 수 있습니다.

다음 명령을 사용하여 이 경보를 생성할 수 있습니다.

```
$ C:\> aws cloudwatch put-metric-alarm \
    --alarm-name snapshot-deletion-failed-monitor \
```

```
    --alarm-description "Alarm when snapshot deletions fail" \
    --metric-name SnapshotsDeleteFailed \
    --namespace AWS/EBS \
    --statistic Sum \
    --period 3600 \
    --threshold 0 \
    --comparison-operator GreaterThanThreshold \
    --dimensions "Name=DLMPolicyId,Value=policy_id" \
    --evaluation-periods 1 \
    --alarm-actions sns_topic_arn
```

예제 3: SnapshotsCopiedRegionFailed 지표

SnapshotsCopiedRegionFailed 지표를 사용하여 정책이 스냅샷을 다른 리전으로 복사하지 못하는 시기를 식별할 수 있습니다.

예를 들어 정책이 여러 리전에 걸쳐 매일 스냅샷을 복사하는 경우, SnapshotCrossRegionCopyFailed의 sum이 1 시간의 기간에 대해 0보다 클 때 엔지니어링 팀에 SMS를 보내는 경보를 만들 수 있습니다. 이 기능은 계통의 후속 스냅샷이 정책을 통해 성공적으로 복사되었는지 확인하는 데 유용할 수 있습니다.

다음 명령을 사용하여 이 경보를 생성할 수 있습니다.

```
$ C:\> aws cloudwatch put-metric-alarm \
    --alarm-name snapshot-copy-region-failed-monitor \
    --alarm-description "Alarm when snapshot copy fails" \
    --metric-name SnapshotsCopiedRegionFailed \
    --namespace AWS/EBS \
    --statistic Sum \
    --period 3600 \
    --threshold 0 \
    --comparison-operator GreaterThanThreshold \
    --dimensions "Name=DLMPolicyId,Value=policy_id" \
    --evaluation-periods 1 \
    --alarm-actions sns_topic_arn
```

실패한 작업을 보고하는 정책 관리

정책 중 하나에서 실패한 작업 지표에 대해 0이 아닌 값을 예기치 않게 보고할 경우 수행할 작업에 대한 자세한 내용은 문서 Amazon Data Lifecycle Manager가 CloudWatch 지표에서 실패한 작업을 보고할 경우 어떻게 해야 합니까?를 참조하세요.

를 사용하여 Amazon Data Lifecycle Manager API 호출 로깅 AWS CloudTrail

를 사용하면 사용자 활동 및 API 사용량을 추적하여 내부 정책 및 규제 표준 준수를 입증 AWS CloudTrail할 수 있습니다. 자세한 내용은 AWS CloudTrail 사용 설명서를 참조하십시오.

Amazon Data Lifecycle Manager(Amazon Data Lifecycle Manager)는 Amazon Data Lifecycle Manager에서 사용자, 역할 또는 AWS CloudTrail서비스가 수행한 작업에 대한 레코드를 제공하는 AWS 서비스와 통합됩니다. CloudTrail은 Amazon Data Lifecycle Manager 콘솔의 호출 및 Amazon Data Lifecycle Manager API에 대한 코드 호출을 포함하여 Amazon Data Lifecycle Manager에 대한 모든 APIs. 추적을 생성하면 Amazon Data Lifecycle Manager에 대한 이벤트를 포함하여 CloudTrail 이벤트를 지속적으로 Amazon S3 버킷에 배포할 수 있습니다. Amazon S3 추적을 구성하지 않은 경우에도 이벤트 기록에서 CloudTrail 콘솔의 최신 이벤트를 볼 수 있습니다. CloudTrail에서 수집한 정보를 사용하여 Amazon Data Lifecycle Manager에 수행된 요청, 요청이 수행된 IP 주소, 요청을 수행한 사람, 요청이 수행된 시간 및 추가 세부 정보를 확인할 수 있습니다.

CloudTrail에 대한 자세한 내용은 AWS CloudTrail 사용 설명서를 참조하세요.

CloudTrail의 Amazon Data Lifecycle Manager 정보

AWS 계정을 생성할 때 계정에서 CloudTrail이 활성화됩니다. Amazon Data Lifecycle Manager에서 활동이 발생하면 해당 활동이 이벤트 기록의 다른 AWS 서비스 이벤트와 함께 CloudTrail 이벤트에 기록됩니다. AWS 계정에서 최근 이벤트를 보고 검색하고 다운로드할 수 있습니다. 자세한 설명은 CloudTrail 이벤트 기록으로 이벤트 보기를 참조하세요.

Amazon Data Lifecycle Manager에 대한 이벤트를 포함하여 AWS 계정에 이벤트를 지속적으로 기록하려면 추적을 생성합니다. CloudTrail은 추적을 사용하여 Amazon S3 버킷으로 로그 파일을 전송할 수 있습니다. 콘솔에서 트레일을 생성하면 기본적으로 모든 리전에 트레일이 적용됩니다. 추적은 AWS 파티션의 모든 리전에서 이벤트를 로깅하고 지정한 Amazon S3 버킷으로 로그 파일을 전송합니다. 또한 CloudTrail 로그에서 수집된 이벤트 데이터를 추가로 분석하고 조치를 취하도록 다른 AWS 서비스를 구성할 수 있습니다. 자세한 내용은 다음을 참조하세요.

- 트레일 생성 개요
- CloudTrail 지원 서비스 및 통합
- CloudTrail에서 Amazon SNS 알림 구성
- 여러 리전으로부터 CloudTrail 로그 파일 받기 및 여러 계정으로부터 CloudTrail 로그 파일 받기

모든 Amazon Data Lifecycle Manager 작업은 CloudTrail에서 로깅됩니다. 예를 들어 `CreateLifecyclePolicy` 및 `DeleteLifecyclePolicy` 작업을 호출하면 CloudTrail 로그 파일에 항목이 생성됩니다. 작업의 전체 목록은 작업을 참조하세요.

모든 이벤트 또는 로그 항목에는 요청을 생성했던 사용자에 관한 정보가 포함됩니다. ID 정보를 이용하면 다음을 쉽게 판단할 수 있습니다.

- 요청한 사용자입니다.
- 역할 또는 페더레이션 사용자의 임시 자격 증명을 사용하여 요청이 생성되었는지 여부.
- 요청이 다른 AWS 서비스에 의해 이루어졌는지 여부입니다.

자세한 설명은 CloudTrail userIdentity 요소를 참조하세요.

Amazon Data Lifecycle Manager 로그 파일 항목 이해

트레일이란 지정한 S3 버킷에 이벤트를 로그 파일로 입력할 수 있게 하는 구성입니다. CloudTrail 로그 파일에는 하나 이상의 로그 항목이 포함될 수 있습니다. 이벤트는 모든 소스의 단일 요청을 나타내며 요청된 작업, 작업 날짜와 시간, 요청 파라미터 등에 대한 정보를 포함하고 있습니다. CloudTrail 로그 파일은 퍼블릭 API 직접 호출에 대한 순서 지정된 스택 추적이 아니기 때문에 특정 순서로 표시되지 않습니다.

Example예: CreateLifecyclePolicy

다음 예제는 `CreateLifecyclePolicy` 작업에 대한 CloudTrail 로그 항목을 보여 줍니다.

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "Root",
        "principalId": "123456789012",
        "arn": "arn:aws:iam::123456789012:root",
        "accountId": "123456789012",
        "accessKeyId": "AKIAJA2ELRVCPEXAMPLE",
        "userName": "user",
        "sessionContext": {
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2018-07-24T18:01:05Z"
            }
        }
    },
```

```
    "eventTime": "2018-07-24T18:20:28Z",
    "eventSource": "dlm.amazonaws.com",
    "eventName": "CreateLifecyclePolicy",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "54.240.230.179",
    "userAgent": "console.ec2.amazonaws.com",
    "requestParameters": {
        "ExecutionRoleArn": "arn:aws:iam::123456789012:role/service-role/
AWSDataLifecycleManagerServiceRole",
        "PolicyDetails": {
            "ResourceTypes": [
                "VOLUME"
            ],
            "Schedules": [
                {
                    "CreateRule": {
                        "Interval": 12,
                        "IntervalUnit": "HOURS",
                        "Times": [
                            "09:00"
                        ]
                    },
                    "Name": "Default Schedule",
                    "RetainRule": {
                        "Count": 3
                    },
                    "TagsToAdd": [
                        {
                            "Key": "Name",
                            "Value": "backup-my-volume"
                        }
                    ]
                }
            ],
            "TargetTags": [
                {
                    "Key": "Name",
                    "Value": "my-volume"
                }
            ]
        },
        "Description": "test-cloudtrail",
        "State": "DISABLED"
    },
```

```
    "responseElements": {
        "PolicyId": "policy-04ff8755fce0599eb"
    },
    "requestID": "3d714ca6-8f6e-11e8-92a4-35fd765427f0",
    "eventID": "28ab3121-6040-4a40-80c7-ae59b3adf405",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "recipientAccountId": "123456789012"
}
```

Example예: DeleteLifecyclePolicy

다음 예제는 `DeleteLifecyclePolicy` 작업에 대한 CloudTrail 로그 항목을 보여 줍니다.

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "Root",
        "principalId": "123456789012",
        "arn": "arn:aws:iam::123456789012:root",
        "accountId": "123456789012",
        "accessKeyId": "AKIAJA2ELRVCPEXAMPLE",
        "userName": "user",
        "sessionContext": {
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2018-07-24T18:01:05Z"
            }
        }
    },
    "eventTime": "2018-07-24T19:33:33Z",
    "eventSource": "dlm.amazonaws.com",
    "eventName": "DeleteLifecyclePolicy",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "54.240.230.241",
    "userAgent": "console.ec2.amazonaws.com",
    "requestParameters": {
        "policyId": "policy-04ff8755fce0599eb"
    },
    "responseElements": null,
    "requestID": "73260971-8f78-11e8-a156-598016e53fb2",
    "eventID": "3740f2fb-0d6a-4712-a7ad-eb9f17103fb2",
    "readOnly": false,
    "eventType": "AwsApiCall",
```

```
    "recipientAccountId": "123456789012"
}
```

Amazon Data Lifecycle Manager를 위한 서비스 엔드포인트

엔드포인트는 AWS 웹 서비스의 진입점 역할을 하는 URL입니다. Amazon Data Lifecycle Manager는 다음과 같은 엔드포인트 유형을 지원합니다.

- IPv4 엔드포인트
- IPv4 및 IPv6를 모두 지원하는 이중 스택 엔드포인트
- FIPS 엔드포인트

요청 시에, 사용할 엔드포인트와 리전을 지정할 수 있습니다. 엔드포인트를 지정하지 않으면 기본적으로 IPv4 엔드포인트가 사용됩니다. 다른 엔드포인트 유형을 사용하려면 요청에서 이를 지정해야 합니다. 이렇게 하는 방법의 예제는 엔드포인트 지정 섹션을 참조하세요.

Amazon Data Lifecycle Manager의 경우 Amazon Web Services 일반 참조의 Amazon Data Lifecycle Manager 엔드포인트를 참조하세요

주제

- IPv4 엔드포인트
- 이중 스택(IPv4 및 IPv6) 엔드포인트
- FIPS 엔드포인트
- 엔드포인트 지정

IPv4 엔드포인트

IPv4 엔드포인트는 IPv4 트래픽만 지원합니다. IPv4 엔드포인트는 모든 리전에 사용할 수 있습니다.

엔드포인트 이름의 일부로 리전을 지정해야 합니다. 엔드포인트 이름에는 다음 명명 규칙이 사용됩니다.

- dlm.*region*.amazonaws.com

예를 들어 미국 동부(버지니아 북부) 리전의 IPv4 엔드포인트는 `dlm.us-east-1.amazonaws.com`입니다.

이중 스택(IPv4 및 IPv6) 엔드포인트

이중 스택 엔드포인트는 IPv4 트래픽과 IPv6 트래픽을 모두 지원합니다. 이중 스택 엔드포인트는 모든 리전에 사용할 수 있습니다.

IPv6를 사용하려면 이중 스택 엔드포인트를 사용해야 합니다. 이중 스택 엔드포인트에 요청하는 경우, 엔드포인트 URL이 네트워크 및 클라이언트에서 사용하는 프로토콜에 따라 IPv6 또는 IPv4 주소로 확인됩니다.

엔드포인트 이름의 일부로 리전을 지정해야 합니다. 이중 스택 엔드포인트 이름에는 다음 명명 규칙이 사용됩니다.

- `dlm.region.api.aws`

예를 들어 미국 동부(버지니아 북부) 리전의 듀얼 스택 엔드포인트는 `dlm.us-east-1.api.aws`입니다.

FIPS 엔드포인트

Amazon Data Lifecycle Manager는 다음 리전에 대해 FIPS 검증 듀얼 스택(IPv4 및 IPv6) 엔드포인트를 제공합니다.

- `us-east-1` - 미국 동부(버지니아 북부)
- `us-east-2` - 미국 동부(오하이오)
- `us-west-1` - 미국 서부(캘리포니아 북부)
- `us-west-2` - 미국 서부(오레곤)
- `ca-central-1` - 캐나다(중부)
- `ca-west-1` - 캐나다 서부(캘거리)

FIPS 듀얼 스택 엔드포인트는 명명 규칙을 사용합니다`dlm-fips.region.api.aws`. 예를 들어 미국 동부(버지니아 북부) 리전의 FIPS 듀얼 스택 엔드포인트는 `dlm-fips.us-east-1.api.aws`입니다.

엔드포인트 지정

다음 예는 AWS CLI를 사용하여 `US East (N. Virginia)` 리전의 엔드포인트를 지정하는 방법을 보여줍니다.

- 이중 스택

```
aws dlm create-default-role \
--resource-type snapshot \
--endpoint-url https://dlm.us-east-2.api.aws
```

- IPv4

```
aws dlm create-default-role \
--resource-type snapshot \
--endpoint-url https://dlm.us-east-2.amazonaws.com
```

VPC와 Amazon EBS 간에 프라이빗 연결 생성

AWS PrivateLink에서 제공되는 인터페이스 VPC 엔드포인트를 생성하여 VPC와 Amazon EBS 간에 프라이빗 연결을 설정할 수 있습니다. 인터넷 게이트웨이, NAT 디바이스, VPN 연결 또는 AWS Direct Connect 연결을 사용하지 않고 VPC에 있는 것처럼 Amazon EBS에 액세스할 수 있습니다. VPC의 인스턴스는 Amazon EBS와 통신하는 데 퍼블릭 IP 주소를 필요로 하지 않습니다.

인터페이스 엔드포인트에 대해 사용 설정하는 각 서브넷에서 엔드포인트 네트워크 인터페이스를 생성합니다.

자세한 내용은 AWS PrivateLink 가이드의를 AWS 서비스 통한 액세스를 AWS PrivateLink 참조하세요.

> **Note**
>
> Amazon Data Lifecycle Manager는 모든 상용 및 AWS GovCloud (US) 리전에 대해 IPv4 인터페이스 VPC 엔드포인트를 지원하고 상용 리전에 대해서만 IPv6 인터페이스 VPC 엔드포인트를 지원합니다.

Amazon EBS VPC 엔드포인트에 대한 고려 사항

Amazon EBS에 대한 인터페이스 VPC 엔드포인트를 설정하려면 먼저 AWS PrivateLink 가이드의 고려사항을 검토합니다.

기본적으로 엔드포인트를 통해 Amazon EBS에 대한 전체 액세스가 허용됩니다. VPC 엔드포인트 정책을 사용하여 인터페이스 엔드포인트에 대한 액세스를 제어할 수 있습니다. Amazon EBS에 대한 액

세스를 제어하는 VPC 엔드포인트에 엔드포인트 정책을 연결할 수 있습니다. 이 정책은 다음 정보를 지정합니다.

- 작업을 수행할 수 있는 위탁자.
- 수행할 수 있는 작업.
- 작업을 수행할 수 있는 리소스.

자세한 내용은 Amazon VPC 사용 설명서의 VPC 종단점을 통해 서비스에 대한 액세스 제어를 참조하세요.

다음은 Amazon EBS에 대한 엔드포인트 정책의 예입니다. 이 정책은 엔드포인트에 연결되면 모든 사용자에게 Amazon Data Lifecycle Manager 정책에 대한 요약 정보를 가져올 수 있는 권한을 부여합니다.

```
{
  "Statement": [{
    "Action": "dlm:GetLifecyclePolicies",
    "Effect": "Allow",
    "Principal": "*",
    "Resource": "*"
  }]
}
```

Amazon EBS에 대한 인터페이스 VPC 엔드포인트를 생성합니다.

Amazon VPC 콘솔 또는 AWS Command Line Interface (AWS CLI)를 사용하여 Amazon EBS에 대한 VPC 엔드포인트를 생성할 수 있습니다. 자세한 정보는 AWS PrivateLink 가이드의 VPC 엔드포인트 생성을 참조하세요.

다음 서비스 이름을 사용하여 Amazon EBS에 대한 VPC 엔드포인트를 생성합니다.

- `com.amazonaws.region.dlm`

엔드포인트에 프라이빗 DNS를 사용하도록 설정하는 경우 리전에 대한 기본 DNS 이름(예: `dlm.us-east-1.amazonaws.com`)을 사용하여 Amazon EBS에 API 요청을 할 수 있습니다.

Amazon Data Lifecycle Manager 문제 해결

다음 설명서는 발생 가능한 문제를 해결하는 데 도움이 됩니다.

주제

- 오류: Role with name already exists

오류: `Role with name already exists`

설명

콘솔을 사용하여 정책을 생성하려고 하면 `Role with name AWSDataLifecycleManagerDefaultRole already exists` 또는 `Role with name AWSDataLifecycleManagerDefaultRoleForAMIManagement already exists` 오류가 발생합니다.

원인

기본 역할의 ARN 형식은 콘솔을 사용하여 생성했는지 AWS CLI를 사용하여 생성했는지에 따라 다릅니다. ARN은 다르지만 역할은 동일한 역할 이름을 사용하므로 콘솔과 AWS CLI간에 역할 이름 지정 충돌이 발생합니다.

Solution

이 문제를 해결하려면 다음과 같이 실행합니다.

1. (사전 및 사후 스크립트에 대해 활성화된 스냅샷 정책의 경우에만 해당) AWSDataLifecycleManagerSSMFullAccess AWS 관리형 정책을 AWSDataLifecycleManagerDefaultRole IAM 역할에 수동으로 연결합니다. 자세한 내용은 IAM ID 권한 추가를 참조하세요.
2. Amazon Data Lifecycle Manager 정책을 생성할 때 IAM 역할에서 다른 역할 선택을 선택하고 AWSDataLifecycleManagerDefaultRole(스냅샷 정책의 경우) 또는 AWSDataLifecycleManagerDefaultRoleForAMIManagement(AMI 정책의 경우)를 선택합니다.
3. 평소와 같이 정책을 계속 생성합니다.

AWS Backup Amazon Elastic Block Store용

AWS Backup 는 전체 AWS 환경에서 데이터 보호를 중앙 집중화하고 자동화하는 완전 관리형 서비스입니다. 스냅샷 일정 및 리소스당 보존을 관리하는 대신 단일 백업 정책을 정의하고 계정 및 리전에 적용합니다. 리소스에 태그를 지정하고 팀별 구성 없이 자동으로 AWS Backup 보호합니다. 이 서비스는 Amazon EBS, Amazon EC2, Amazon S3, Amazon EFS, Amazon RDS 및 Aurora를 비롯한 많은 AWS 서비스를 지원합니다. 따라서 하나의 정책은 워크로드당 별도의 백업 도구 없이 볼륨, 데이터베이스, 파일 시스템 및 객체 스토어를 함께 보호할 수 있습니다.

Amazon EBS의 경우는 Amazon EC2 인스턴스에 연결된 모든 볼륨에서 point-in-time 백업을 조정하는 충돌 일관성 다중 볼륨 스냅샷을 AWS Backup 제공합니다. 이는 여러 볼륨에 걸쳐 있는 데이터베이스 및 클러스터링된 애플리케이션에 필수적입니다. AWS Backup 검색은 항목 수준 복구를 활성화하여 한 단계 더 나아가 복구합니다. 전체 볼륨을 복원하지 않고도 Amazon EBS 스냅샷에서 개별 파일 또는 디렉터리를 찾고 복원할 수 있습니다.

AWS Backup 또한는 스냅샷에 엔터프라이즈급 보안 및 규정 준수를 추가합니다. 논리적 에어 갭 저장소는 프로덕션 환경에서 복구 시점을 격리합니다. 볼트 잠금은 불변성을 적용합니다. Amazon GuardDuty 통합은 백업에서 맬웨어를 스캔합니다. Backup Audit Manager는 백업 태세를 지속적으로 평가하고 감사 준비 규정 준수 보고서를 생성합니다.

시작하려면 다음 리소스를 참조하세요.

- AWS Backup 개발자 안내서
- 다중 볼륨, 충돌 일관성 백업 생성
- Amazon EBS 볼륨 복원
- Amazon EBS 스냅샷에서 개별 파일 검색 및 복구

EBS 다이렉트 API를 사용하여 EBS 스냅샷 콘텐츠에 액세스

Amazon Elastic Block Store(Amazon EBS) direct API를 사용하여 EBS 스냅샷을 생성하고, 스냅샷에 직접 데이터를 쓰고, 스냅샷에서 데이터를 읽고, 두 스냅샷 간의 차이점 또는 변경 사항을 파악할 수 있습니다. Amazon EBS용 백업 서비스를 제공하는 Independent Software Vendor(ISV)는 EBS 다이렉트 API를 사용하여 스냅샷을 통해 더 효율적이고 비용 효과적으로 EBS 볼륨에 대한 증분 변경 사항을 추적할 수 있습니다. 스냅샷에서 새 볼륨을 만들지 않고도 이 작업이 가능하며 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스를 사용하여 차이를 비교할 수 있습니다.

온프레미스의 데이터에서 직접 EBS 볼륨 및 클라우드로 증분 스냅샷을 생성하고 이를 사용해 신속하게 재해 복구를 수행할 수 있습니다. 스냅샷을 쓰고 읽을 수 있으므로 재해 발생 시 온프레미스 데이터를 EBS 스냅샷에 쓸 수 있습니다. 그런 다음 복구 후 스냅샷에서 AWS 또는 온프레미스로 복원할 수 있습니다. Amazon EBS로/에서 데이터를 복사하기 위해 더 이상 복잡한 메커니즘을 빌드하고 유지 관리할 필요가 없습니다.

이 사용 설명서에는 EBS 다이렉트 API를 구성하는 요소의 개요와 이 요소를 효과적으로 사용하는 방법의 예가 나와 있습니다. API의 작업, 데이터 형식, 파라미터 및 오류에 대한 자세한 내용은 EBS 다이렉트 API 참조를 참조하세요. EBS 다이렉트 APIs에 대해 지원되는 AWS 리전, 엔드포인트 및 서비스 할당량에 대한 자세한 내용은의 Amazon EBS 엔드포인트 및 할당량을 참조하세요AWS 일반 참조.

주제

- EBS 다이렉트 API 요금
- EBS 다이렉트 API 개념
- IAM을 사용하여 EBS 다이렉트 API에 대한 액세스 제어
- EBS 다이렉트 API를 사용하여 Amazon EBS 스냅샷 읽기
- EBS 다이렉트 API를 사용하여 Amazon EBS 스냅샷 쓰기
- EBS 다이렉트 API의 암호화 결과
- EBS 다이렉트 API 체크섬을 사용하여 스냅샷 데이터 검증
- StartSnapshot API 요청에서 멱등성 보장
- EBS 다이렉트 API의 오류 재시도
- EBS 다이렉트 API의 성능 최적화
- EBS 다이렉트 API용 서비스 엔드포인트
- AWS EBS 다이렉APIs용 SDK 코드 예제
- VPC와 EBS 다이렉트 API 간에 프라이빗 연결 생성

- APIs 호출 로깅 AWS CloudTrail
- EBS 다이렉트 API FAQ

EBS 다이렉트 API 요금

API 요금

EBS 다이렉트 API를 사용하기 위해 지불하는 요금은 요청하는 내용에 따라 다릅니다. 자세한 내용은 Amazon EBS 요금을 참조하세요.

- ListChangedBlocks 및 ListSnapshotBlocks API는 요청당 요금이 청구됩니다. 예를 들어 1,000개의 요청당 0.0006 USD를 청구하는 리전에서 ListSnapshotBlocks API 요청을 100,000회 수행하면 0.06 USD(요청 1,000개당 0.0006 USD x 100)가 청구됩니다.
- GetSnapshotBlock은 반환된 블록당 요금이 부과됩니다. 예를 들어 리전에 반환된 블록 1,000개당 0.003 USD를 청구하는 리전에서 GetSnapshotBlock API 요청을 100,000회 수행하면 0.30 USD(반환된 블록 1,000개당 0.003 USD x 100)가 청구됩니다.
- PutSnapshotBlock은 작성된 블록당 요금이 부과됩니다. 예를 들어 작성된 1,000개의 블록당 0.006 USD를 청구하는 리전에서 10만 건의 PutSnapshotBlock API 요청을 수행하면 0.60 USD(작성된 블록 1,000개당 0.006 USD x 100)가 청구됩니다.

네트워킹 비용

데이터 전송 비용

비 FIPS 엔드포인트를 사용하는 경우 EBS 다이APIs와 동일한 AWS 리전의 Amazon EC2 인스턴스 간에 직접 전송되는 데이터는 무료입니다. 자세한 내용은 AWS 서비스 엔드포인트를 참조하세요. 다른 AWS 서비스가 데이터 전송 경로에 있는 경우 관련 데이터 처리 비용이 청구됩니다. 이러한 서비스에는 PrivateLink 엔드포인트, NAT 게이트웨이 및 Transit Gateway가 포함되지만 이에 국한되지는 않습니다.

VPC 인터페이스 엔드포인트

프라이빗 서브넷의 Amazon EC2 인스턴스 또는 AWS Lambda 함수에서 EBS 다이렉트 APIs를 사용하는 경우 NAT 게이트웨이를 사용하는 대신 VPC 인터페이스 엔드포인트를 사용하여 네트워크 데이터 전송 비용을 줄일 수 있습니다. 자세한 내용은 VPC와 EBS 다이렉트 API 간에 프라이빗 연결 생성 단원을 참조하십시오.

EBS 다이렉트 API 개념

EBS 다이렉트 API를 시작하려면 먼저 다음과 같은 핵심 개념을 이해해야 합니다.

스냅샷

스냅샷은 EBS 볼륨에서 데이터를 백업하는 기본 방법입니다. EBS 다이렉트 API를 사용하면 온프레미스 디스크의 데이터를 스냅샷으로 백업할 수도 있습니다. 스토리지 비용을 절약하기 위해 이전 스냅샷 이후로 변경된 볼륨 데이터만 연속 스냅샷에 증분식으로 포함시킵니다. 자세한 내용은 Amazon EBS 스냅샷 단원을 참조하십시오.

> **Note**
>
> EBS 다이렉APIs는 퍼블릭 스냅샷 및 로컬 스냅샷을 지원하지 않습니다 AWS Outposts.

블록

블록은 스냅샷에 있는 데이터의 조각입니다. 각 스냅샷에 수천 개의 블록을 포함할 수 있습니다. 스냅샷에 있는 모든 블록은 크기가 고정되어 있습니다.

블록 인덱스

블록 인덱스는 512KiB 블록 단위의 논리적 인덱스입니다. 블록 인덱스를 식별하려면 논리적 볼륨에 있는 데이터의 논리적 오프셋을 블록 크기(데이터의 논리적 오프셋/524288)로 나눕니다. 데이터의 논리적 오프셋은 512KiB로 정렬되어야 합니다.

블록 토큰

블록 토큰은 스냅샷에 있는 블록의 식별 해시이며 블록 데이터를 찾는 데 사용됩니다. EBS 다이렉트 API에서 반환된 블록 토큰은 임시로, 지정된 만료 타임스탬프에 변경되거나 동일한 스냅샷에 대해 다른 ListSnapshotBlocks 또는 ListChangedBlocks 요청을 실행하면 변경됩니다.

체크섬

체크섬은 전송 또는 저장 중에 발생한 오류를 탐지하기 위해 데이터 블록에서 제공되는 작은 크기의 데이터입니다. EBS 다이렉트 API는 데이터 무결성을 검증하기 위해 체크섬을 사용합니다. EBS 스냅샷에서 데이터를 읽을 때는 이 서비스가 전송된 각 데이터 블록에 대해 Base64로 인코딩된 SHA256 체크

섬을 제공하며, 이 체크섬을 검증에 사용할 수 있습니다. EBS 스냅샷에 데이터를 쓸 때는 사용자가 전송된 각 데이터 블록에 대해 Base64로 인코딩된 SHA256 체크섬을 제공해야 합니다. 그러면 이 서비스가 제공된 체크섬을 사용하여 수신된 데이터를 검증합니다. 자세한 내용은 이 설명서 후반부의 EBS 다이렉트 API 체크섬을 사용하여 스냅샷 데이터 검증 섹션을 참조하세요.

암호화(Encryption)

암호화는 읽을 수 없는 코드로 데이터를 변환하고 암호화에 사용되는 KMS 키에 액세스할 수 있는 사용자만 코드를 해독할 수 있도록 하여 데이터를 보호합니다. EBS 다이렉트 API를 사용하여 암호화된 스냅샷을 읽고 쓸 수 있지만 몇 가지 제한 사항이 있습니다. 자세한 내용은 이 설명서 후반부의 EBS 다이렉트 API의 암호화 결과 섹션을 참조하세요.

API 작업

EBS 다이렉트 API는 6가지 작업으로 구성됩니다. 3가지 읽기 작업과 3가지 쓰기 작업이 있습니다. 읽기 작업은 다음과 같습니다.

- ListSnapshotBlocks - 지정된 스냅샷에 있는 블록의 블록 인덱스와 블록 토큰을 반환합니다.
- ListChangedBlocks - 볼륨/스냅샷 계보가 같은 2개의 지정된 스냅샷에서 차이가 있는 블록의 블록 인덱스와 블록 토큰을 반환합니다.
- GetSnapshotBlock - 지정된 스냅샷 ID, 블록 인덱스 및 블록 토큰의 블록 내 데이터를 반환합니다.

쓰기 작업은 다음과 같습니다.

- StartSnapshot - 기존 스냅샷의 증분 스냅샷 또는 새 스냅샷으로 스냅샷을 시작합니다. 시작된 스냅샷은 CompleteSnapshot 작업 사용을 마칠 때까지 대기 중 상태로 유지됩니다.
- PutSnapshotBlock - 개별 블록 형태로 시작된 스냅샷에 데이터를 추가합니다. 전송되는 데이터의 블록에 대해 Base64 인코딩 SHA256 체크섬을 지정해야 합니다. 이 서비스는 전송이 완료되고 나면 체크섬을 검증합니다. 서비스에서 계산된 체크섬이 지정한 체크섬과 일치하지 않으면 요청이 실패합니다.
- CompleteSnapshot - 대기 중 상태인 시작된 스냅샷을 완료합니다. 그러면 스냅샷이 완료됨 상태로 바뀝니다.

서명 버전 4 서명

서명 버전 4는 HTTP에서 보낸 AWS 요청에 인증 정보를 추가하는 프로세스입니다. 보안을 위해에 대한 대부분의 요청은 액세스 키 ID와 보안 액세스 키로 구성된 액세스 키로 서명해야 AWS 합니다. 이 두

키는 일반적으로 보안 자격 증명이라고 합니다. 계정의 자격 증명을 얻는 방법에 대한 자세한 내용은 AWS 보안 자격 증명을 참조하세요.

HTTP 요청을 수동으로 생성하려는 경우 서명하는 방법을 알아야 합니다. AWS Command Line Interface (AWS CLI) 또는 AWS SDKs 중 하나를 사용하여 요청을 할 때 AWS이러한 도구는 도구를 구성할 때 지정한 액세스 키로 요청에 자동으로 서명합니다. 따라서 이러한 도구를 사용할 경우 요청에 서명하는 방법을 알 필요가 없습니다.

자세한 내용은 IAM 사용 설명서의 AWS API 요청 서명을 참조하세요.

IAM을 사용하여 EBS 다이렉트 API에 대한 액세스 제어

사용자는 다음 정책이 있어야 EBS 다이렉트 API를 사용할 수 있습니다. 자세한 내용은 IAM 사용자의 권한 변경을 참조하세요.

IAM 권한 정책에 사용할 수 있는 EBS 다이렉트 API 리소스, 작업 및 조건 컨텍스트 키에 대한 자세한 내용은 서비스 승인 참조에서 Amazon Elastic Block Store에 사용되는 작업, 리소스 및 조건 키를 참조하세요.

> **Important**
>
> 사용자에게 다음 정책을 할당할 때는 주의해야 합니다. 이 정책을 할당하면 CopySnapshot 또는 CreateVolume 작업과 같은 Amazon EC2 API를 통해 동일한 리소스에 액세스하는 것이 거부된 사용자에게 액세스 권한을 부여할 수 있습니다.

스냅샷 읽기 권한

다음 정책은 특정 AWS 리전의 모든 스냅샷에서 읽기 EBS 다이렉트 APIs를 사용하도록 허용합니다. 정책에서 *<Region>*을 스냅샷의 리전으로 바꾸십시오.

다음 정책은 특정 키-값 태그가 있는 스냅샷에 읽기 EBS 다이렉트 API를 사용할 수 있도록 허용합니다. 정책에서 *<Key>*를 태그의 키 값으로 바꾸고 *<Value>*를 태그 값으로 바꾸십시오.

JSON

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

```
        {
            "Effect": "Allow",
            "Action": [
                "ebs:ListSnapshotBlocks",
                "ebs:ListChangedBlocks",
                "ebs:GetSnapshotBlock"
            ],
            "Resource": "arn:aws:ec2:*::snapshot/*",
            "Condition": {
                "StringEqualsIgnoreCase": {
                    "aws:ResourceTag/<Key>": "<Value>"
                }
            }
        }
    ]
}
```

다음 정책은 특정 시간 범위 내에서만 계정의 모든 스냅샷에 읽기 EBS 다이렉트 API를 사용할 수 있도록 허용합니다. 이 정책은 `aws:CurrentTime` 전역 조건 키를 기반으로 EBS 다이렉트 API를 사용할 수 있는 권한을 부여합니다. 정책에서 표시된 날짜 및 시간 범위를 정책의 날짜 및 시간 범위로 바꿔야 합니다.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ebs:ListSnapshotBlocks",
                "ebs:ListChangedBlocks",
                "ebs:GetSnapshotBlock"
            ],
            "Resource": "arn:aws:ec2:*::snapshot/*",
            "Condition": {
                "DateGreaterThan": {
                    "aws:CurrentTime": "2018-05-29T00:00:00Z"
                },
                "DateLessThan": {
                    "aws:CurrentTime": "2020-05-29T23:59:59Z"
```

```
                }
            }
        }
    ]
}
```

자세한 내용은 IAM 사용 설명서의 사용자의 권한 변경을 참조하세요.

스냅샷 쓰기 권한

다음 정책은 특정 AWS 리전의 모든 스냅샷에서 쓰기 EBS 다이렉트 APIs를 사용하도록 허용합니다. 정책에서 *<Region>*을 스냅샷의 리전으로 바꾸십시오.

다음 정책은 특정 키-값 태그가 있는 스냅샷에 쓰기 EBS 다이렉트 API를 사용할 수 있도록 허용합니다. 정책에서 *<Key>*를 태그의 키 값으로 바꾸고 *<Value>*를 태그 값으로 바꾸십시오.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ebs:StartSnapshot",
                "ebs:PutSnapshotBlock",
                "ebs:CompleteSnapshot"
            ],
            "Resource": "arn:aws:ec2:*::snapshot/*",
            "Condition": {
                "StringEqualsIgnoreCase": {
                    "aws:ResourceTag/<Key>": "<Value>"
                }
            }
        }
    ]
}
```

다음 정책은 모든 EBS 다이렉트 API를 사용할 수 있도록 허용합니다. 또한 상위 스냅샷 ID가 지정된 경우에 한해 StartSnapshot 작업을 허용합니다. 즉, 이 정책은 상위 스냅샷을 사용하지 않고는 새 스냅샷을 시작하지 못하도록 차단합니다.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ebs:*",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "ebs:ParentSnapshot": "arn:aws:ec2:*::snapshot/*"
                }
            }
        }
    ]
}
```

다음 정책은 모든 EBS 다이렉트 API를 사용할 수 있도록 허용합니다. 또한 새 스냅샷에 대해 user 태그 키만 생성할 수 있도록 합니다. 또한 이 정책은 사용자가 태그를 생성할 수 있는 액세스 권한을 갖도록 합니다. StartSnapshot 작업은 태그를 지정할 수 있는 유일한 작업입니다.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ebs:*",
            "Resource": "*",
            "Condition": {
                "ForAllValues:StringEquals": {
                    "aws:TagKeys": "user"
                }
            }
```

```
        },
        {
            "Effect": "Allow",
            "Action": "ec2:CreateTags",
            "Resource": "*"
        }
    ]
}
```

다음 정책은 특정 시간 범위 내에서만 계정의 모든 스냅샷에 쓰기 EBS 다이렉트 API를 사용할 수 있도록 허용합니다. 이 정책은 `aws:CurrentTime` 전역 조건 키를 기반으로 EBS 다이렉트 API를 사용할 수 있는 권한을 부여합니다. 정책에서 표시된 날짜 및 시간 범위를 정책의 날짜 및 시간 범위로 바꿔야 합니다.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ebs:StartSnapshot",
                "ebs:PutSnapshotBlock",
                "ebs:CompleteSnapshot"
            ],
            "Resource": "arn:aws:ec2:*::snapshot/*",
            "Condition": {
                "DateGreaterThan": {
                    "aws:CurrentTime": "2018-05-29T00:00:00Z"
                },
                "DateLessThan": {
                    "aws:CurrentTime": "2020-05-29T23:59:59Z"
                }
            }
        }
    ]
}
```

자세한 내용은 IAM 사용 설명서의 사용자의 권한 변경을 참조하세요.

사용 권한 AWS KMS keys

다음 정책은 특정 KMS 키를 사용하여 암호화된 스냅샷을 해독할 수 있는 권한을 부여합니다. 또한 EBS 암호화를 위한 기본 KMS 키를 사용하여 새 스냅샷을 암호화할 수 있는 권한을 부여합니다. 정책에서 *<Region>*을 KMS 키의 리전으로 바꾸고, *<AccountId>*를 KMS 키의 AWS 계정 ID로 바꾸고, *<KeyId>*를 KMS 키의 ID로 바꿉니다.

> **Note**
>
> 기본적으로 계정의 모든 보안 주체는 Amazon EBS 암호화를 위한 기본 AWS 관리형 KMS 키에 액세스할 수 있으며 EBS 암호화 및 복호화 작업에 사용할 수 있습니다. 고객 관리형 키를 사용하는 경우 고객 관리형 키에 대한 보안 주체 액세스 권한을 부여하려면 고객 관리형 키에 대한 새 키 정책을 생성하거나 기존 키 정책을 수정해야 합니다. 자세한 내용은AWS Key Management Service 개발자 안내서의 AWS KMS의 키 정책을 참조하세요.

> **Tip**
>
> 최소 권한의 원칙을 따르려면 `kms:CreateGrant`에 대한 전체 액세스 권한을 허용하지 마세요. 대신 다음 예제와 AWS 같이 서비스에 의해 사용자를 대신하여 권한 부여가 생성되는 경우에만 `kms:GrantIsForAWSResource` 조건 키를 사용하여 사용자가 KMS 키에 대한 권한 부여를 생성하도록 허용합니다.

자세한 내용은 IAM 사용 설명서의 사용자의 권한 변경을 참조하세요.

EBS 다이렉트 API를 사용하여 Amazon EBS 스냅샷 읽기

다음 단계에서는 EBS 다이렉트 API를 사용하여 스냅샷을 읽는 방법을 설명합니다.

1. ListSnapshotBlocks 작업을 사용하여 스냅샷에 있는 블록의 모든 블록 인덱스와 블록 토큰을 표시합니다. 또는 ListChangedBlocks 작업을 사용하여 볼륨/스냅샷 계보가 같은 2개의 지정된 스냅샷에서 차이가 있는 블록의 블록 인덱스와 블록 토큰만 표시합니다. 이러한 작업을 통해 데이터를 가져올 블록의 블록 토큰 및 블록 인덱스를 확인할 수 있습니다.
2. GetSnapshotBlock 작업을 사용하고 데이터를 가져올 블록의 블록 인덱스 및 블록 토큰을 지정합니다.

> **Note**
>
> 아카이브된 스냅샷에는 EBS 다이렉트 API를 사용할 수 없습니다.

다음 예에서는 EBS 다이렉트 API를 사용하여 스냅샷을 읽는 방법을 보여 줍니다.

주제

- 스냅샷 블록 나열
- 두 스냅샷에서 차이가 있는 블록 나열
- 스냅샷에서 블록 데이터 가져오기

스냅샷 블록 나열

AWS CLI

다음 list-snapshot-blocks 예제 명령은 스냅샷 `snap-0987654321`에 있는 블록의 블록 인덱스 및 블록 토큰을 반환합니다. `--starting-block-index` 파라미터는 결과를 `1000`보다 큰 블록 인덱스로 제한하고, `--max-results` 파라미터는 결과를 첫 번째 `100` 블록으로 제한합니다.

```
aws ebs list-snapshot-blocks --snapshot-id snap-0987654321 --starting-block-
index 1000 --max-results 100
```

이전 명령에 대한 다음 예제 응답은 스냅샷의 블록 인덱스 및 블록 토큰을 나열합니다. `get-snapshot-block` 명령을 사용하고 데이터를 가져올 블록의 블록 인덱스 및 블록 토큰을 지정합니다. 블록 토큰은 기재된 만료 시간까지 유효합니다.

```
{
      "Blocks": [
          {
              "BlockIndex": 1001,
              "BlockToken": "AAABAV3/
PNhXOynVdMYHUpPsetaSvjLB1dtIGfbJv5OJ0sX855EzGTWos4a4"
          },
          {
              "BlockIndex": 1002,
              "BlockToken": "AAABATGQIgwr0WwIuqIMjCA/Sy7e/
YoQFZsHejzGNvjKauzNgzeI13YHBfQB"
```

```
            },
            {
                "BlockIndex": 1007,
                "BlockToken": "AAABAZ9CTuQtUvp/
dXqRWw4d07eOgTZ3jvn6hiW30W9duM8MiMw6yQayzF2c"
            },
            {
                "BlockIndex": 1012,
                "BlockToken": "AAABAQdzxhw0rVV6PNmsfo/
YRIxo9JPR85XxPf1BLjg0Hec6pygYr6laE1p0"
            },
            {
                "BlockIndex": 1030,
                "BlockToken": "AAABAaYvPax6mv+iGWLdTUjQtFWouQ7Dqz6nSD9L
+CbXnvpkswA6iDID523d"
            },
            {
                "BlockIndex": 1031,
                "BlockToken": "AAABATgWZC0XcFwUKvTJbUXMiSPg59KVxJGL
+BWBClkw6spzCxJVqDVaTskJ"
            },
            ...
        ],
        "ExpiryTime": 1576287332.806,
        "VolumeSize": 32212254720,
        "BlockSize": 524288
    }
```

AWS API

다음 ListSnapshotBlocks 예제 요청은 스냅샷 `snap-0acEXAMPLEcf41648`에 있는 블록의 블록 인덱스 및 블록 토큰을 반환합니다. `startingBlockIndex` 파라미터는 결과를 `1000`보다 큰 블록 인덱스로 제한하고, `maxResults` 파라미터는 결과를 첫 번째 `100` 블록으로 제한합니다.

```
GET /snapshots/snap-0acEXAMPLEcf41648/blocks?maxResults=100&startingBlockIndex=1000
 HTTP/1.1
  Host: ebs.us-east-2.amazonaws.com
  Accept-Encoding: identity
  User-Agent: <User agent parameter>
  X-Amz-Date: 20200617T231953Z
  Authorization: <Authentication parameter>
```

이전 요청에 대한 다음 예제 응답은 스냅샷의 블록 인덱스 및 블록 토큰을 나열합니다. GetSnapshotBlock 작업을 사용하고 데이터를 가져올 블록의 블록 인덱스 및 블록 토큰을 지정합니다. 블록 토큰은 기재된 만료 시간까지 유효합니다.

```
HTTP/1.1 200 OK
  x-amzn-RequestId: d6e5017c-70a8-4539-8830-57f5557f3f27
  Content-Type: application/json
  Content-Length: 2472
  Date: Wed, 17 Jun 2020 23:19:56 GMT
  Connection: keep-alive

  {
      "BlockSize": 524288,
      "Blocks": [
          {
              "BlockIndex": 0,
              "BlockToken": "AAUBAcuWqOCnDNuKle11s7IIX6jp6FYcC/q8oT93913HhvLvA
+3JRrSybp/0"
          },
          {
              "BlockIndex": 1536,
              "BlockToken":
 "AAUBAWudwfmofcrQhGVlLwuRKm2b8ZXPiyrgoykTRC6IU1NbxKWDY1pPjvnV"
          },
          {
              "BlockIndex": 3072,
              "BlockToken":
 "AAUBAV7p6pC5fKAC7TokoNCtAnZhqq27u6YEXZ3MwRevBkDjmMx6iuA6tsBt"
          },
          {
              "BlockIndex": 3073,
              "BlockToken":
 "AAUBAbqt9zpqBUEvtO2HINAfFaWToOwlPjbIsQOlx6JUN/0+iMQl0NtNbnX4"
          },
          ...
      ],
      "ExpiryTime": 1.59298379649E9,
      "VolumeSize": 3
  }
```

두 스냅샷에서 차이가 있는 블록 나열

두 스냅샷 간에 변경된 블록을 나열하기 위해 페이지 매김 요청을 할 때 다음 사항에 유의하세요.

- 응답에는 하나 이상의 빈 ChangedBlocks 배열이 포함될 수 있습니다. 예:
  - 스냅샷 1: 블록 인덱스가 0~999이고 1,000개의 블록이 포함된 전체 스냅샷.
  - 스냅샷 2: 블록 인덱스가 999이고 변경된 블록이 하나만 있는 증분 스냅샷.

  이러한 스냅샷에 대해 변경된 블록을 StartingBlockIndex = 0 및 MaxResults = 100으로 나열하면 ChangedBlocks의 빈 배열이 반환됩니다. 블록 인덱스가 900~999인 블록을 포함하는 10번째 결과 집합에 변경된 블록이 반환될 때까지 nextToken을 사용하여 나머지 결과를 요청해야 합니다.
- 응답은 스냅샷에서 기록되지 않은 블록을 건너뛸 수 있습니다. 예:
  - 스냅샷 1: 블록 인덱스가 2000~2999이고 1,000개의 블록이 포함된 전체 스냅샷.
  - 스냅샷 2: 블록 인덱스가 2000이고 변경된 블록이 하나만 포함된 증분 스냅샷.

  이러한 스냅샷에 대해 변경된 블록을 StartingBlockIndex = 0 및 MaxResults = 100으로 나열하면 응답에서 블록 인덱스 0~1999를 건너뛰고 블록 인덱스 2000을 포함합니다. 응답에는 빈 ChangedBlocks 배열이 포함되지 않습니다.

AWS CLI

다음 list-changed-blocks 예제 요청은 스냅샷 snap-1234567890과 snap-0987654321에서 차이가 있는 블록의 블록 인덱스 및 블록 토큰을 반환합니다. --starting-block-index 파라미터는 결과를 0보다 큰 블록 인덱스로 제한하고, --max-results 파라미터는 결과를 첫 번째 500 블록으로 제한합니다.

```
aws ebs list-changed-blocks --first-snapshot-id snap-1234567890 --second-snapshot-
id snap-0987654321 --starting-block-index 0 --max-results 500
```

이전 명령에 대한 다음 예제 응답은 블록 인덱스 0, 6000, 6001, 6002, 6003이 두 스냅샷에서 차이가 있다는 것을 보여 줍니다. 또한 응답에 기재된 두 번째 블록 토큰이 없으므로 지정된 첫 번째 스냅샷 ID에만 블록 인덱스 6001, 6002, 6003이 있고 두 번째 스냅샷 ID에는 없습니다.

get-snapshot-block 명령을 사용하고 데이터를 가져올 블록의 블록 인덱스 및 블록 토큰을 지정합니다. 블록 토큰은 기재된 만료 시간까지 유효합니다.

```
{
```

```
      "ChangedBlocks": [
          {
              "BlockIndex": 0,
              "FirstBlockToken": "AAABAVahm9SO60Dyi0ORySzn2ZjGjW/
KN3uygGlS0QOYWesbzBbDnX2dGpmC",
              "SecondBlockToken":
 "AAABAf8o0o6UFi1rDbSZGIRaCEdDyBu9TlvtCQxxoKV8qrUPQP7vcM6iWGSr"
          },
          {
              "BlockIndex": 6000,
              "FirstBlockToken": "AAABAbYSiZvJ0/
R9tz8suI8dSzecLjN4kkazK8inFXVintPkdaVFLfCMQsKe",
              "SecondBlockToken":
 "AAABAZnqTdzFmKRpsaMAsDxviVqEI/3jJzI2crq2eFDCgHmyNf777elD9oVR"
          },
          {
              "BlockIndex": 6001,
              "FirstBlockToken": "AAABASBpSJ2UAD3PLxJnCt6zun4/
T4sU25Bnb8jB5Q6FRXHFqAIAqE04hJoR"
          },
          {
              "BlockIndex": 6002,
              "FirstBlockToken": "AAABASqX4/
NWjvNceoyMUljcRd0DnwbSwNnes1UkoP62CrQXvn47BY5435aw"
          },
          {
              "BlockIndex": 6003,
              "FirstBlockToken":
 "AAABASmJ0O5JxAOce25rF4P1sdRtyIDsX12tFEDunnePYUKOf4PBROuICb2A"
          },
          ...
      ],
      "ExpiryTime": 1576308931.973,
      "VolumeSize": 32212254720,
      "BlockSize": 524288,
      "NextToken": "AAADARqElNng/sV98CYk/bJDCXeLJmLJHnNSkHvLzVaO0zsPH/QM3Bi3zF//
O6Mdi/BbJarBnp8h"
  }
```

AWS API

다음 ListChangedBlocks 예제 요청은 스냅샷 snap-0acEXAMPLEcf41648과 snap-0c9EXAMPLE1b30e2f에서 차이가 있는 블록의 블록 인덱스 및 블록 토큰을 반환합니다.

startingBlockIndex 파라미터는 결과를 0보다 큰 블록 인덱스로 제한하고, maxResults 파라미터는 결과를 첫 번째 500 블록으로 제한합니다.

```
GET /snapshots/snap-0c9EXAMPLE1b30e2f/changedblocks?
firstSnapshotId=snap-0acEXAMPLEcf41648&maxResults=500&startingBlockIndex=0 HTTP/1.1
  Host: ebs.us-east-2.amazonaws.com
  Accept-Encoding: identity
  User-Agent: <User agent parameter>
  X-Amz-Date: 20200617T232546Z
  Authorization: <Authentication parameter>
```

이전 요청에 대한 다음 예제 응답은 블록 인덱스 0, 3072, 6002, 6003이 두 스냅샷에서 차이가 있다는 것을 보여 줍니다. 또한 응답에 기재된 두 번째 블록 토큰이 없으므로 지정된 첫 번째 스냅샷 ID에만 블록 인덱스 6002 및 6003이 있고 두 번째 스냅샷 ID에는 없습니다.

GetSnapshotBlock 작업을 사용하고 데이터를 가져올 블록의 블록 인덱스 및 블록 토큰을 지정합니다. 블록 토큰은 기재된 만료 시간까지 유효합니다.

```
HTTP/1.1 200 OK
  x-amzn-RequestId: fb0f6743-6d81-4be8-afbe-db11a5bb8a1f
  Content-Type: application/json
  Content-Length: 1456
  Date: Wed, 17 Jun 2020 23:25:47 GMT
  Connection: keep-alive

  {
      "BlockSize": 524288,
      "ChangedBlocks": [
          {
              "BlockIndex": 0,
              "FirstBlockToken": "AAUBAVaWqOCnDNuKle11s7IIX6jp6FYcC/
tJuVT1GgP23AuLntwiMdJ+OJkL",
              "SecondBlockToken": "AAUBASxzy0Y0b33JVRLoYm3NOresCxn5RO+HVFzXW3Y/
RwfFaPX2Edx8QHCh"
          },
          {
              "BlockIndex": 3072,
              "FirstBlockToken":
 "AAUBAcHp6pC5fKAC7TokoNCtAnZhqq27u6fxRfZOLEmeXLmHBf2R/Yb24MaS",
              "SecondBlockToken":
 "AAUBARGCaufCqBRZC8tEkPYGGkSv3vqvOjJ2xKDi3ljDFiytUxBLXYgTmkid"
          },
```

```
            {
                "BlockIndex": 6002,
                "FirstBlockToken": "AAABASqX4/
NWjvNceoyMUljcRd0DnwbSwNnes1UkoP62CrQXvn47BY5435aw"
            },
            {
                "BlockIndex": 6003,
                "FirstBlockToken":
 "AAABASmJ0O5JxAOce25rF4P1sdRtyIDsX12tFEDunnePYUKOf4PBROuICb2A"
            },
            ...
        ],
        "ExpiryTime": 1.592976647009E9,
        "VolumeSize": 3
    }
```

스냅샷에서 블록 데이터 가져오기

AWS CLI

다음 get-snapshot-block 예제 요청은 스냅샷 6001에서 블록 토큰이 AAABASBpSJ2UAD3PLxJnCt6zun4/T4sU25Bnb8jB5Q6FRXHFqAIAqE04hJoR인 블록 인덱스 snap-1234567890의 데이터를 반환합니다. Windows 컴퓨터에서 data 디렉터리의 C:\Temp 파일로 이진 데이터가 출력됩니다. Linux 또는 Unix 컴퓨터에서 명령을 실행할 때는 출력 경로를 /tmp/data로 바꾸어 data 디렉터리의 /tmp 파일로 데이터를 출력하십시오.

```
aws ebs get-snapshot-block --snapshot-id snap-1234567890 --block-index 6001 --block-
token AAABASBpSJ2UAD3PLxJnCt6zun4/T4sU25Bnb8jB5Q6FRXHFqAIAqE04hJoR C:/Temp/data
```

이전 명령에 대한 다음 예제 응답은 반환된 데이터의 크기, 데이터를 검증하기 위한 체크섬 및 체크섬 알고리즘을 보여줍니다. 요청 명령에서 지정한 디렉터리와 파일에 이진 데이터가 자동으로 저장됩니다.

```
{
      "DataLength": "524288",
      "Checksum": "cf0Y6/Fn0oFa4VyjQPOa/iD0zhTflPTKzxGv2OKowXc=",
      "ChecksumAlgorithm": "SHA256"
  }
```

AWS API

다음 GetSnapshotBlock 예제 요청은 스냅샷 3072에서 블록 토큰이 AAUBARGCaufCqBRZC8tEkPYGGkSv3vqvOjJ2xKDi3ljDFiytUxBLXYgTmkid인 블록 인덱스 snap-0c9EXAMPLE1b30e2f의 데이터를 반환합니다.

```
GET /snapshots/snap-0c9EXAMPLE1b30e2f/blocks/3072?
blockToken=AAUBARGCaufCqBRZC8tEkPYGGkSv3vqvOjJ2xKDi3ljDFiytUxBLXYgTmkid HTTP/1.1
  Host: ebs.us-east-2.amazonaws.com
  Accept-Encoding: identity
  User-Agent: <User agent parameter>
  X-Amz-Date: 20200617T232838Z
  Authorization: <Authentication parameter>
```

이전 요청에 대한 다음 예제 응답은 반환된 데이터의 크기, 데이터를 검증하기 위한 체크섬 및 체크섬 생성에 사용되는 알고리즘을 보여줍니다. 이진 데이터는 응답 본문에 포함돼 전송되며 다음 예에서는 *BlockData*로 표시되어 있습니다.

```
HTTP/1.1 200 OK
  x-amzn-RequestId: 2d0db2fb-bd88-474d-a137-81c4e57d7b9f
  x-amz-Data-Length: 524288
  x-amz-Checksum: Vc0yY2j3qg8bUL9I6GQuI2orTudrQRBDMIhcy7bdEsw=
  x-amz-Checksum-Algorithm: SHA256
  Content-Type: application/octet-stream
  Content-Length: 524288
  Date: Wed, 17 Jun 2020 23:28:38 GMT
  Connection: keep-alive

  BlockData
```

EBS 다이렉트 API를 사용하여 Amazon EBS 스냅샷 쓰기

다음 단계에서는 EBS 다이렉트 API를 사용하여 증분 스냅샷을 쓰는 방법을 설명합니다.

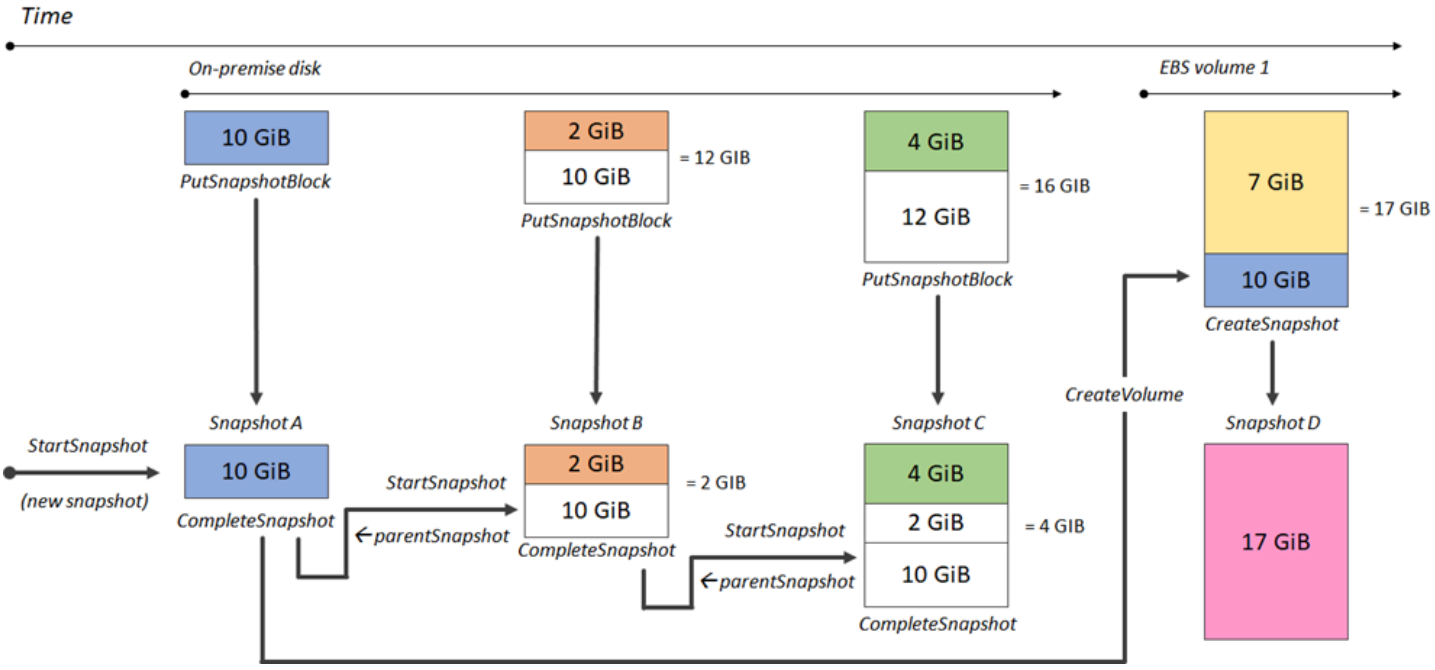
1. StartSnapshot 작업을 사용하고 상위 스냅샷 ID를 지정하여 기존 스냅샷의 증분 스냅샷으로 스냅샷을 시작하거나 상위 스냅샷 ID를 생략하여 새 스냅샷을 시작합니다. 이 작업은 대기 중 상태의 새 스냅샷 ID를 반환합니다.
2. PutSnapshotBlock 작업을 사용하고 대기 중인 스냅샷의 ID를 지정하여 개별 블록 형식으로 데이터를 추가합니다. 전송되는 데이터의 블록에 대해 Base64 인코딩 SHA256 체크섬을 지정해야 합니다.

이 서비스는 수신된 데이터의 체크섬을 계산하고 지정된 체크섬과 대조하여 검증합니다. 체크섬이 일치하지 않으면 작업이 실패합니다.

3. 대기 중인 스냅샷에 데이터를 추가했으면 CompleteSnapshot 작업을 사용하여 스냅샷을 봉인하고 완료됨 상태로 전환하는 비동기 워크플로우를 시작합니다.

이 단계를 반복하여 이전에 생성한 스냅샷을 상위 스냅샷으로 사용해 새 증분 스냅샷을 생성합니다.

예를 들어 다음 다이어그램에서 스냅샷 A는 처음 시작된 새 스냅샷입니다. 스냅샷 A는 스냅샷 B를 시작하는 데 상위 스냅샷으로 사용됩니다. 스냅샷 B는 스냅샷 C를 시작하고 생성하는 데 상위 스냅샷으로 사용됩니다. 스냅샷 A, B, C는 증분 스냅샷입니다. 스냅샷 A는 EBS 볼륨 1을 생성하는 데 사용됩니다. 스냅샷 D는 EBS 볼륨 1에서 생성됩니다. 스냅샷 D는 A의 증분 스냅샷이며 B 또는 C의 증분 스냅샷이 아닙니다.



다음 예에서는 EBS 다이렉트 API를 사용하여 스냅샷을 쓰는 방법을 보여 줍니다.

주제

- 스냅샷 시작
- 스냅샷에 데이터 추가
- 스냅샷 완료

스냅샷 시작

AWS CLI

다음 start-snapshot 예제 요청은 스냅샷 8을 상위 스냅샷으로 사용하여 snap-123EXAMPLE1234567GiB 스냅샷을 시작합니다. 새 스냅샷은 상위 스냅샷의 증분 스냅샷이 됩니다. 지정된 60분의 제한 시간 내에 스냅샷에 대한 추가 또는 완료 요청이 없으면 스냅샷이 오류 상태로 전환됩니다. 550e8400-e29b-41d4-a716-446655440000 클라이언트 토큰은 요청에 대한 멱등성을 보장합니다. 클라이언트 토큰을 생략하면 AWS SDK가 자동으로 토큰을 생성합니다. 멱등성에 대한 자세한 내용은 StartSnapshot API 요청에서 멱등성 보장 섹션을 참조하세요.

```
aws ebs start-snapshot --volume-size 8 --parent-snapshot snap-123EXAMPLE1234567 --
timeout 60 --client-token 550e8400-e29b-41d4-a716-446655440000
```

이전 명령에 대한 다음 예제 응답은 스냅샷 ID, AWS 계정 ID, 상태, 볼륨 크기(GiB) 및 스냅샷의 블록 크기를 보여줍니다. 스냅샷이 pending 상태로 시작됩니다. 후속 put-snapshot-block 명령에서 스냅샷 ID를 지정하여 스냅샷에 데이터를 쓴 다음 complete-snapshot 명령을 사용하여 스냅샷을 완료하고 상태를 completed로 변경합니다.

```
{
    "SnapshotId": "snap-0aaEXAMPLEe306d62",
    "OwnerId": "111122223333",
    "Status": "pending",
    "VolumeSize": 8,
    "BlockSize": 524288
}
```

AWS API

다음 StartSnapshot 예제 요청은 스냅샷 8을 상위 스냅샷으로 사용하여 snap-123EXAMPLE1234567GiB 스냅샷을 시작합니다. 새 스냅샷은 상위 스냅샷의 증분 스냅샷이 됩니다. 지정된 60분의 제한 시간 내에 스냅샷에 대한 추가 또는 완료 요청이 없으면 스냅샷이 오류 상태로 전환됩니다. 550e8400-e29b-41d4-a716-446655440000 클라이언트 토큰은 요청에 대한 멱등성을 보장합니다. 클라이언트 토큰을 생략하면 AWS SDK가 자동으로 토큰을 생성합니다. 멱등성에 대한 자세한 내용은 StartSnapshot API 요청에서 멱등성 보장 섹션을 참조하세요.

```
POST /snapshots HTTP/1.1
Host: ebs.us-east-2.amazonaws.com
Accept-Encoding: identity
User-Agent: <User agent parameter>
```

```
X-Amz-Date: 20200618T040724Z
Authorization: <Authentication parameter>

{
    "VolumeSize": 8,
    "ParentSnapshot": snap-123EXAMPLE1234567,
    "ClientToken": "550e8400-e29b-41d4-a716-446655440000",
    "Timeout": 60
}
```

이전 요청에 대한 다음 예제 응답은 스냅샷 ID, AWS 계정 ID, 상태, 볼륨 크기(GiB) 및 스냅샷의 블록 크기를 보여줍니다. 스냅샷이 대기 중 상태로 시작됩니다. 후속 PutSnapshotBlocks 요청에서 스냅샷 ID를 지정하여 스냅샷에 데이터를 씁니다.

```
HTTP/1.1 201 Created
x-amzn-RequestId: 929e6eb9-7183-405a-9502-5b7da37c1b18
Content-Type: application/json
Content-Length: 181
Date: Thu, 18 Jun 2020 04:07:29 GMT
Connection: keep-alive

{
    "BlockSize": 524288,
    "Description": null,
    "OwnerId": "138695307491",
    "Progress": null,
    "SnapshotId": "snap-052EXAMPLEc85d8dd",
    "StartTime": null,
    "Status": "pending",
    "Tags": null,
    "VolumeSize": 8
}
```

스냅샷에 데이터 추가

AWS CLI

다음 put-snapshot-block 예제 명령은 스냅샷 snap-0aaEXAMPLEe306d62의 블록 인덱스 1000에 524288바이트의 데이터를 씁니다. Base64로 인코딩된 QOD3gmEQOXATfJx2Aa34W4FU2nZGyXfqtsUuktOw8DM= 체크섬은 SHA256 알고리즘을 사용하여 생성되었습니다. 전송되는 데이터는 /tmp/data 파일에 있습니다.

```
aws ebs put-snapshot-block --snapshot-id snap-0aaEXAMPLEe306d62
 --block-index 1000 --data-length 524288 --block-data /tmp/data --
checksum QOD3gmEQOXATfJx2Aa34W4FU2nZGyXfqtsUuktOw8DM= --checksum-algorithm SHA256
```

이전 명령에 대한 다음 예제 응답은 서비스에서 수신한 데이터의 데이터 길이, 체크섬 및 체크섬 알고리즘을 확인합니다.

```
{
    "DataLength": "524288",
    "Checksum": "QOD3gmEQOXATfJx2Aa34W4FU2nZGyXfqtsUuktOw8DM=",
    "ChecksumAlgorithm": "SHA256"
}
```

AWS API

다음 PutSnapshot 예제 요청은 스냅샷 524288의 블록 인덱스 1000에 snap-052EXAMPLEc85d8dd바이트의 데이터를 씁니다. Base64로 인코딩된 QOD3gmEQOXATfJx2Aa34W4FU2nZGyXfqtsUuktOw8DM= 체크섬은 SHA256 알고리즘을 사용하여 생성되었습니다. 데이터는 요청 본문에 포함돼 전송되며 다음 예에서는 *BlockData*로 표시되어 있습니다.

```
PUT /snapshots/snap-052EXAMPLEc85d8dd/blocks/1000 HTTP/1.1
Host: ebs.us-east-2.amazonaws.com
Accept-Encoding: identity
x-amz-Data-Length: 524288
x-amz-Checksum: QOD3gmEQOXATfJx2Aa34W4FU2nZGyXfqtsUuktOw8DM=
x-amz-Checksum-Algorithm: SHA256
User-Agent: <User agent parameter>
X-Amz-Date: 20200618T042215Z
X-Amz-Content-SHA256: UNSIGNED-PAYLOAD
Authorization: <Authentication parameter>

          BlockData
```

이전 요청에 대한 다음 예제 응답은 서비스에서 수신한 데이터의 데이터 길이, 체크섬 및 체크섬 알고리즘을 확인합니다.

```
HTTP/1.1 201 Created
x-amzn-RequestId: 643ac797-7e0c-4ad0-8417-97b77b43c57b
x-amz-Checksum: QOD3gmEQOXATfJx2Aa34W4FU2nZGyXfqtsUuktOw8DM=
x-amz-Checksum-Algorithm: SHA256
```

```
Content-Type: application/json
Content-Length: 2
Date: Thu, 18 Jun 2020 04:22:12 GMT
Connection: keep-alive

{}
```

스냅샷 완료

AWS CLI

다음 complete-snapshot 예제 명령은 스냅샷 snap-0aaEXAMPLEe306d62를 완료합니다. 이 명령은 5 블록을 스냅샷에 쓰도록 지정합니다. 6D3nmwi5f2F0wlh7xX8QprrJBFzDX8aacdOcA3KCM3c= 체크섬은 스냅샷에 쓴 전체 데이터 세트의 체크섬을 나타냅니다. 체크섬에 대한 자세한 내용은 이 설명서 앞부분에 있는 EBS 다이렉트 API 체크섬을 사용하여 스냅샷 데이터 검증 섹션을 참조하세요.

```
aws ebs complete-snapshot --snapshot-id snap-0aaEXAMPLEe306d62 --changed-blocks-
count 5 --checksum 6D3nmwi5f2F0wlh7xX8QprrJBFzDX8aacdOcA3KCM3c= --checksum-
algorithm SHA256 --checksum-aggregation-method LINEAR
```

다음은 이전 명령에 대한 응답의 예입니다.

```
{
    "Status": "pending"
}
```

AWS API

다음 CompleteSnapshot 예제 요청은 스냅샷 snap-052EXAMPLEc85d8dd를 완료합니다. 이 명령은 5 블록을 스냅샷에 쓰도록 지정합니다. 6D3nmwi5f2F0wlh7xX8QprrJBFzDX8aacdOcA3KCM3c= 체크섬은 스냅샷에 쓴 전체 데이터 세트의 체크섬을 나타냅니다.

```
POST /snapshots/completion/snap-052EXAMPLEc85d8dd HTTP/1.1
Host: ebs.us-east-2.amazonaws.com
Accept-Encoding: identity
x-amz-ChangedBlocksCount: 5
x-amz-Checksum: 6D3nmwi5f2F0wlh7xX8QprrJBFzDX8aacdOcA3KCM3c=
x-amz-Checksum-Algorithm: SHA256
```

```
x-amz-Checksum-Aggregation-Method: LINEAR
User-Agent: <User agent parameter>
X-Amz-Date: 20200618T043158Z
Authorization: <Authentication parameter>
```

다음은 이전 요청에 대한 응답의 예입니다.

```
HTTP/1.1 202 Accepted
x-amzn-RequestId: 06cba5b5-b731-49de-af40-80333ac3a117
Content-Type: application/json
Content-Length: 20
Date: Thu, 18 Jun 2020 04:31:50 GMT
Connection: keep-alive

{"Status":"pending"}
```

EBS 다이렉트 API의 암호화 결과

StartSnapshot을 사용하여 새 스냅샷을 시작할 때 암호화 상태는 암호화됨, KmsKeyArn 및 ParentSnapshotId에 대해 지정한 값과 AWS 계정이 기본적으로 암호화가 활성화되어 있는지 여부에 따라 달라집니다.

> **Note**
>
> - 암호화와 함께 EBS 다이렉트 API를 사용하려면 추가 IAM 권한이 필요할 수 있습니다. 자세한 내용은 사용 권한 AWS KMS keys 섹션을 참조하세요.
> - AWS 계정에서 Amazon EBS 암호화가 기본적으로 활성화된 경우 암호화되지 않은 스냅샷을 생성할 수 없습니다.
> - AWS 계정에서 Amazon EBS 암호화가 기본적으로 활성화된 경우 암호화되지 않은 상위 스냅샷을 사용하여 새 스냅샷을 시작할 수 없습니다. 먼저 상위 스냅샷을 복사하여 암호화해야 합니다. 자세한 내용은 Amazon EBS 스냅샷 복사 단원을 참조하십시오.

주제

- 암호화 결과: 암호화되지 않은 상위 스냅샷
- 암호화 결과: 암호화된 상위 스냅샷
- 암호화 결과: 상위 스냅샷 없음

암호화 결과: 암호화되지 않은 상위 스냅샷

다음 표에서는 암호화되지 않은 상위 스냅샷을 지정할 때 가능한 각 설정 조합에 대한 암호화 결과를 설명합니다.

ParentSnapshotId	암호화됨	KmsKeyArn	암호화 기본 제공	결과
암호화되지 않음	생략됨	생략됨	활성화됨	`ValidationException` 과 함께 요청이 실패합니다.
			비활성화됨	스냅샷이 암호화되지 않았습니다.
		지정됨	활성화됨	
			비활성화됨	
암호화되지 않음	True	생략됨	활성화됨	`ValidationException` 과 함께 요청이 실패합니다.
			비활성화됨	
		지정됨	활성화됨	
			비활성화됨	
암호화되지 않음	False	생략됨	활성화됨	`ValidationException` 과 함께 요청이 실패합니다.
			비활성화됨	
		지정됨	활성화됨	
			비활성화됨	

암호화 결과: 암호화된 상위 스냅샷

다음 표에서는 암호화된 상위 스냅샷을 지정할 때 가능한 각 설정 조합에 대한 암호화 결과를 설명합니다.

ParentSnapshotId	암호화됨	KmsKeyArn	암호화 기본 제공	결과
암호화됨	생략됨	생략됨	활성화됨	스냅샷은 상위 스냅샷과 동일한 KMS 키를 사용하여 암호화됩니다.
			비활성화됨	
		지정됨	활성화됨	`ValidationException` 과 함께 요청이 실패합니다.
			비활성화됨	
암호화됨	True	생략됨	활성화됨	`ValidationException` 과 함께 요청이 실패합니다.
			비활성화됨	
		지정됨	활성화됨	
			비활성화됨	
암호화됨	False	생략됨	활성화됨	`ValidationException` 과 함께 요청이 실패합니다.
			비활성화됨	
		지정됨	활성화됨	
			비활성화됨	

암호화 결과: 상위 스냅샷 없음

다음 표에서는 상위 스냅샷을 사용하지 않을 때 가능한 각 설정 조합에 대한 암호화 결과를 설명합니다.

ParentSnapshotId	암호화됨	KmsKeyArn	암호화 기본 제공	결과
생략됨	True	생략됨	활성화됨	스냅샷은 계정의 기본 KMS 키를 사용하여 암호화됩니다. *
			비활성화됨	

ParentSnapshotId	암호화됨	KmsKeyArn	암호화 기본 제공	결과
		지정됨	활성화됨	스냅샷은 KmsKeyArn에 대해 지정된 KMS 키를 사용하여 암호화됩니다.
			비활성화됨	
생략됨	False	생략됨	활성화됨	`ValidationException` 과 함께 요청이 실패합니다.
			비활성화됨	스냅샷이 암호화되지 않았습니다.
		지정됨	활성화됨	`ValidationException` 과 함께 요청이 실패합니다.
			비활성화됨	
생략됨	생략됨	생략됨	활성화됨	스냅샷은 계정의 기본 KMS 키를 사용하여 암호화됩니다. *
			비활성화됨	스냅샷이 암호화되지 않았습니다.
		지정됨	활성화됨	스냅샷은 KmsKeyArn에 대해 지정된 KMS 키를 사용하여 암호화됩니다.
			비활성화됨	

*이 기본 KMS 키는 고객 관리형 키이거나 Amazon EBS 암호화를 위한 기본 AWS 관리형 KMS 키일 수 있습니다.

EBS 다이렉트 API 체크섬을 사용하여 스냅샷 데이터 검증

GetSnapshotBlock 작업은 스냅샷 블록에 있는 데이터를 반환하고, PutSnapshotBlock 작업은 스냅샷의 블록에 데이터를 추가합니다. 전송되는 블록 데이터는 Signature 버전 4 서명 프로세스의 일부로 서명되지 않습니다. 따라서 다음과 같이 체크섬을 사용하여 데이터의 무결성이 검증됩니다.

- GetSnapshotBlock 작업을 사용하는 경우 응답은 x-amz-Checksum 헤더를 사용하여 블록 데이터에 대한 Base64로 인코딩된 SHA256 체크섬을 제공하고 x-amz-Checksum-Algorithm 헤더를 사용하여

체크섬 알고리즘을 제공합니다. 반환된 체크섬을 사용하여 데이터의 무결성을 검증하세요. 생성한 체크섬이 Amazon EBS에서 제공된 체크섬과 일치하지 않는 경우 데이터가 유효하지 않은 것으로 간주하고 요청을 다시 시도해야 합니다.

- PutSnapshotBlock 작업을 사용하는 경우 요청에서 x-amz-Checksum 헤더를 사용하여 블록 데이터에 대한 Base64로 인코딩된 SHA256 체크섬을 제공하고 x-amz-Checksum-Algorithm 헤더를 사용하여 체크섬 알고리즘을 제공해야 합니다. 제공하는 체크섬은 Amazon EBS에서 생성된 체크섬과 대조되어 데이터의 무결성이 검증됩니다. 체크섬이 일치하지 않으면 요청이 실패합니다.
- CompleteSnapshot 작업을 사용하는 경우 요청에서 스냅샷에 추가된 전체 데이터 세트에 대해 Base64로 인코딩된 SHA256 체크섬을 선택적으로 제공할 수 있습니다. x-amz-Checksum 헤더를 사용하여 체크섬을 제공하고, x-amz-Checksum-Algorithm 헤더를 사용하여 체크섬 알고리즘을 제공하고, x-amz-Checksum-Aggregation-Method 헤더를 사용하여 체크섬 집계 방법을 제공합니다. 선형 집계 방법을 사용하여 집계된 체크섬을 생성하려면 작성된 각 블록의 체크섬을 블록 인덱스의 오름차순으로 정렬하고 이를 연결하여 단일 문자열을 형성한 다음 SHA256 알고리즘을 사용하여 전체 문자열에 대한 체크섬을 생성합니다.

이러한 작업의 체크섬은 Signature 버전 4 서명 프로세스의 일부입니다.

StartSnapshot API 요청에서 멱등성 보장

멱등성은 API 요청이 한 번만 완료되도록 합니다. 멱등성 요청에서는 원래 요청이 성공적으로 완료된 경우 후속 재시도에서 원래 성공한 요청의 결과를 반환하며 추가 영향이 없습니다.

StartSnapshot API는 클라이언트 토큰을 사용하여 멱등성을 지원합니다. 클라이언트 토큰은 API 요청을 할 때 지정하는 고유 문자열입니다. 요청이 성공적으로 완료된 후 동일한 클라이언트 토큰과 동일한 요청 파라미터를 사용하여 API 요청을 다시 시도하면 원래 요청의 결과가 반환됩니다. 동일한 클라이언트 토큰으로 요청을 다시 시도하지만 요청 파라미터를 하나 이상 변경하면 `ConflictException` 오류가 반환됩니다.

자체 클라이언트 토큰을 지정하지 않으면 AWS SDKs가 요청에 대한 클라이언트 토큰을 자동으로 생성하여 멱등성을 보장합니다.

클라이언트 토큰은 최대 64자의 ASCII 문자를 포함하는 모든 문자열이 될 수 있습니다. 다른 요청에 동일한 클라이언트 토큰을 재사용해서는 안 됩니다.

API를 사용하여 고유 클라이언트 토큰으로 멱등성 StartSnapshot 요청을 하려면

`ClientToken` 요청 파라미터를 지정합니다.

```
POST /snapshots HTTP/1.1
Host: ebs.us-east-2.amazonaws.com
Accept-Encoding: identity
User-Agent: <User agent parameter>
X-Amz-Date: 20200618T040724Z
Authorization: <Authentication parameter>

{
    "VolumeSize": 8,
    "ParentSnapshot": snap-123EXAMPLE1234567,
    "ClientToken": "550e8400-e29b-41d4-a716-446655440000",
    "Timeout": 60
}
```

를 사용하여 자체 클라이언트 토큰으로 멱등성 StartSnapshot 요청을 수행하려면 AWS CLI

`client-token` 요청 파라미터를 지정합니다.

```
$ C:\> aws ebs start-snapshot --region us-east-2 --volume-size 8 --parent-
snapshot snap-123EXAMPLE1234567 --timeout 60 --client-token 550e8400-e29b-41d4-
a716-446655440000
```

EBS 다이렉트 API의 오류 재시도

AWS SDK는 오류 응답을 반환하는 요청에 대해 자동 재시도 로직을 구현합니다. AWS SDK에 대한 재시도 설정을 구성할 수 있습니다. 자세한 내용은 해당 SDK의 설명서를 참조하세요.

실패한 일부 요청을 자동으로 다시 시도하도록 AWS CLI를 구성할 수 있습니다. 에 대한 재시도 구성에 대한 자세한 내용은 AWS Command Line Interface 사용 설명서의 AWS CLI 재시도를 AWS CLI참조하세요.

AWS Query API는 실패한 요청에 대한 재시도 로직을 지원하지 않습니다. HTTP 또는 HTTPS 요청을 사용하는 경우 클라이언트 애플리케이션에서 재시도 로직을 구현해야 합니다.

다음 표에는 가능한 API 오류 응답이 나와 있습니다. 일부 API 오류는 재시도할 수 있습니다. 클라이언트 애플리케이션에서 항상 재시도 가능한 오류를 수신하는 실패한 요청을 재시도해야 합니다.

오류	응답 코드	설명	발생 원인	재시도 가능 여부
InternalServerException	500	네트워크 또는 AWS 서버 측 문제로 인해 요청이 실패했습니다.	모든 API	예
ThrottlingException	400	API 요청 수가 계정에 대해 허용되는 최대 API 요청 제한 제한을 초과했습니다.	모든 API	예
RequestThrottleException	400	API 요청 수가 스냅샷에 대해 허용되는 최대 API 요청 제한 제한을 초과했습니다.	GetSnapshotBlock | PutSnapshotBlock	예
'Failed to read block data' 메시지와 함께 ValidationException 발생	400	제공된 데이터 블록을 읽을 수 없습니다.	PutSnapshotBlock	예
다른 메시지와 함께 ValidationException 발생	400	요청 구문의 형식이 잘못되었거나 입력이 AWS 서비스에서 지정한 제약 조건을 충족하지 않습니다.	모든 API	아니요
ResourceNotFoundException	404	지정된 스냅샷 ID가 존재하지 않습니다.	모든 API	아니요

오류	응답 코드	설명	발생 원인	재시도 가능 여부
ConflictException	409	지정된 클라이언트 토큰이 이전에 요청 파라미터가 다른 유사한 요청에 사용되었습니다. 자세한 내용은 StartSnapshot API 요청에서 멱등성 보장 단원을 참조하십시오.	StartSnapshot	아니요
AccessDeniedException	403	요청한 작업을 수행할 수 있는 권한이 없습니다.	모든 API	아니요
ServiceQuotaExceededException	402	요청을 이행하면 계정에 대한 하나 이상의 종속 서비스 할당량이 초과되기 때문에 요청에 실패했습니다.	모든 API	아니요
InvalidSignatureException	403	요청 승인 서명이 만료되었습니다. 승인 서명을 새로 고친 후에만 요청을 재시도할 수 있습니다.	모든 API	아니요

EBS 다이렉트 API의 성능 최적화

API 요청을 동시에 실행할 수 있습니다. PutSnapshotBlock 지연 시간이 100ms라고 가정하면 한 스레드는 1초에 10개의 요청을 처리할 수 있습니다. 또한 클라이언트 애플리케이션이 여러 스레드 및 연결(예: 100개의 연결)을 생성한다고 가정하면 초당 총 1000(10 * 100)개의 요청을 할 수 있습니다. 이는 초당 약 500MB의 처리량에 해당합니다.

다음은 애플리케이션에서 고려할 몇 가지 사항입니다.

- 각 스레드가 별도의 연결을 사용합니까? 애플리케이션에서 연결이 제한되어 있으면 여러 스레드가 연결을 사용할 수 있을 때까지 대기하여 처리량이 낮아집니다.
- 두 추가 요청 사이에 애플리케이션에 대기 시간이 있습니까? 이 경우 스레드의 유효 처리량이 줄어듭니다.
- 인스턴스의 대역폭 제한 - 인스턴스의 대역폭을 다른 애플리케이션에서 공유하는 경우 PutSnapshotBlock 요청에 사용할 수 있는 처리량이 제한될 수 있습니다.

병목 현상을 방지하려면 계정에서 실행될 수 있는 다른 워크로드를 고려해야 합니다. 또한 조절, 시간 초과 및 서비스 사용 불가를 처리하려면 EBS 다이렉트 API 워크플로우에 재시도 메커니즘을 빌드해야 합니다.

EBS 다이렉트 API 서비스 할당량을 검토하여 초당 실행할 수 있는 최대 API 요청을 확인합니다. 자세한 내용은AWS 일반 참조에서 Amazon Elastic Block Store 엔드포인트 및 할당량을 참조하세요.

EBS 다이렉트 API용 서비스 엔드포인트

엔드포인트는 AWS 웹 서비스의 진입점 역할을 하는 URL입니다. EBS 다이렉트 API는 다음과 같은 엔드포인트 유형을 지원합니다.

- IPv4 엔드포인트
- IPv4 및 IPv6를 모두 지원하는 이중 스택 엔드포인트
- FIPS 엔드포인트

요청 시에, 사용할 엔드포인트와 리전을 지정할 수 있습니다. 엔드포인트를 지정하지 않으면 기본적으로 IPv4 엔드포인트가 사용됩니다. 다른 엔드포인트 유형을 사용하려면 요청에서 이를 지정해야 합니다. 이렇게 하는 방법의 예제는 엔드포인트 지정 섹션을 참조하세요.

리전에 대한 자세한 내용은 Amazon EC2 사용 설명서의 리전 및 가용 영역을 참조하세요. EBS 다이렉트 API의 엔드포인트 목록은 Amazon Web Services 일반 참조의 EBS 다이렉트 API용 엔드포인트를 참조하세요.

주제

- IPv4 엔드포인트
- 이중 스택(IPv4 및 IPv6) 엔드포인트
- FIPS 엔드포인트

- 엔드포인트 지정

IPv4 엔드포인트

IPv4 엔드포인트는 IPv4 트래픽만 지원합니다. IPv4 엔드포인트는 모든 리전에 사용할 수 있습니다.

EBS 다이렉트 API는 리전별 IPv4 엔드포인트만 요청에 사용할 수 있도록 지원합니다. 엔드포인트 이름의 일부로 리전을 지정해야 합니다. 엔드포인트 이름에는 다음 명명 규칙이 사용됩니다.

- `ebs.region.amazonaws.com`

예를 들어 요청을 `us-east-2` IPv4 엔드포인트로 전달하려면 `ebs.us-east-2.amazonaws.com`을 엔드포인트로 지정해야 합니다. EBS 다이렉트 API의 엔드포인트 목록은 Amazon Web Services 일반 참조의 EBS 다이렉트 API용 엔드포인트를 참조하세요.

가격 책정

동일한 리전에서 IPv4 엔드포인트를 사용하여 EBS 다이렉트 API와 Amazon EC2 인스턴스 간에 직접 전송되는 데이터에 대해서는 요금이 부과되지 않습니다. 그러나 AWS PrivateLink 엔드포인트, NAT 게이트웨이 또는 Amazon VPC Transit Gateway와 같은 중간 서비스가 있는 경우 관련 비용이 청구됩니다.

이중 스택(IPv4 및 IPv6) 엔드포인트

이중 스택 엔드포인트는 IPv4 트래픽과 IPv6 트래픽을 모두 지원합니다. 이중 스택 엔드포인트는 모든 리전에 사용할 수 있습니다.

IPv6를 사용하려면 이중 스택 엔드포인트를 사용해야 합니다. 이중 스택 엔드포인트에 요청하는 경우, 엔드포인트 URL이 네트워크 및 클라이언트에서 사용하는 프로토콜에 따라 IPv6 또는 IPv4 주소로 확인됩니다.

EBS 다이렉트 API는 리전별 이중 스택 엔드포인트만 지원합니다. 이는 엔드포인트 이름의 일부로 리전을 지정해야 함을 뜻합니다. 이중 스택 엔드포인트 이름에는 다음 명명 규칙이 사용됩니다.

- `ebs.region.api.aws`

예를 들어 `eu-west-1` 리전의 이중 스택 엔드포인트 이름은 `ebs.eu-west-1.api.aws`입니다. EBS 다이렉트 API의 엔드포인트 목록은 Amazon Web Services 일반 참조의 EBS 다이렉트 API용 엔드포인트를 참조하세요.

가격 책정

동일한 리전에서 이중 스택 엔드포인트를 사용하여 EBS 다이렉트 API와 Amazon EC2 인스턴스 간에 직접 전송되는 데이터에 대해서는 요금이 부과되지 않습니다. 그러나 AWS PrivateLink 엔드포인트, NAT 게이트웨이 또는 Amazon VPC Transit Gateway와 같은 중간 서비스가 있는 경우 관련 비용이 청구됩니다.

FIPS 엔드포인트

EBS 다이렉트 API는 다음 리전에 대해 FIPS 검증을 거친 IPv4 및 이중 스택(IPv4 및 IPv6) 엔드포인트를 제공합니다.

- `us-east-1` - 미국 동부(버지니아 북부)
- `us-east-2` - 미국 동부(오하이오)
- `us-west-1` - 미국 서부(캘리포니아 북부)
- `us-west-2` - 미국 서부(오레곤)
- `ca-central-1` - 캐나다(중부)
- `ca-west-1` - 캐나다 서부(캘거리)

FIPS IPv4 엔드포인트에는 `ebs-fips.region.amazonaws.com`이라는 명명 규칙이 사용됩니다. 예를 들어 `us-east-1`의 FIPS IPv4 엔드포인트는 `ebs-fips.us-east-1.amazonaws.com`입니다.

FIPS 이중 스택 엔드포인트에는 `ebs-fips.region.api.aws`라는 명명 규칙이 사용됩니다. 예를 들어 `us-east-1`의 FIPS 이중 스택 엔드포인트는 `ebs-fips.us-east-1.api.aws`입니다.

FIPS 엔드포인트에 대한 자세한 내용은 Amazon Web Services 일반 참조의 FIPS 엔드포인트를 참조하세요.

엔드포인트 지정

이 섹션에서는 요청 시에 엔드포인트를 지정하는 방법을 몇 가지 예로 보여줍니다.

AWS CLI

다음 예는 AWS CLI를 사용하여 `us-east-2` 리전의 엔드포인트를 지정하는 방법을 보여줍니다.

- 이중 스택

```
aws ebs list-snapshot-blocks --snapshot-id snap-0987654321 --starting-block-index
 1000 --endpoint-url https://ebs.us-east-2.api.aws
```

- IPv4

```
aws ebs list-snapshot-blocks --snapshot-id snap-0987654321 --starting-block-index
 1000 --endpoint-url https://ebs.us-east-2.amazonaws.com
```

AWS SDK for Java 2.x

다음 예는 AWS SDK for Java 2.x를 사용하여 us-east-2 리전의 엔드포인트를 지정하는 방법을 보여줍니다.

- 이중 스택

```
AwsClientBuilder.EndpointConfiguration config = new
 AwsClientBuilder.EndpointConfiguration("https://ebs.us-east-2.api.aws", "us-
east-2");
AmazonEBS ebs = AmazonEBSClientBuilder.standard()
    .withEndpointConfiguration(config)
    .build();
```

- IPv4

```
AwsClientBuilder.EndpointConfiguration config = new
 AwsClientBuilder.EndpointConfiguration("https://ebs.us-east-2.amazonaws.com",
 "us-east-2");
AmazonEBS ebs = AmazonEBSClientBuilder.standard()
    .withEndpointConfiguration(config)
    .build();
```

AWS SDK for Go

다음 예는 AWS SDK for Go를 사용하여 us-east-2 리전의 엔드포인트를 지정하는 방법을 보여줍니다.

- 이중 스택

```
sess := session.Must(session.NewSession())
```

```
svc := ebs.New(sess, &aws.Config{
    Region: aws.String(endpoints.UsEast2RegionID),
    Endpoint: aws.String("https://ebs.us-east-2.api.aws")
})
```

- IPv4

```
sess := session.Must(session.NewSession())
svc := ebs.New(sess, &aws.Config{
    Region: aws.String(endpoints.UsEast2RegionID),
    Endpoint: aws.String("https://ebs.us-east-2.amazonaws.com")
})
```

AWS EBS 다이렉APIs용 SDK 코드 예제

다음 코드 예제에서는 AWS 소프트웨어 개발 키트(SDK)와 함께 EBS 다이렉APIs를 사용하는 방법을 보여줍니다.

작업

- AWS SDK 또는 CLI와 StartSnapshot 함께 사용
- AWS SDK 또는 CLI와 PutSnapshotBlock 함께 사용
- AWS SDK 또는 CLI와 CompleteSnapshot 함께 사용

AWS SDK 또는 CLI와 **StartSnapshot** 함께 사용

다음 코드 예시는 StartSnapshot의 사용 방법을 보여 줍니다.

Rust

SDK for Rust

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
async fn start(client: &Client, description: &str) -> Result<String, Error> {
```

```
    let snapshot = client
        .start_snapshot()
        .description(description)
        .encrypted(false)
        .volume_size(1)
        .send()
        .await?;

    Ok(snapshot.snapshot_id.unwrap())
}
```

- API 세부 정보는 AWS SDK for Rust API 참조의 StartSnapshot을 참조하세요.

AWS SDK 또는 CLI와 **PutSnapshotBlock** 함께 사용

다음 코드 예시는 PutSnapshotBlock의 사용 방법을 보여 줍니다.

Rust

SDK for Rust

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
async fn add_block(
    client: &Client,
    id: &str,
    idx: usize,
    block: Vec<u8>,
    checksum: &str,
) -> Result<(), Error> {
    client
        .put_snapshot_block()
        .snapshot_id(id)
        .block_index(idx as i32)
        .block_data(ByteStream::from(block))
        .checksum(checksum)
```

```
        .checksum_algorithm(ChecksumAlgorithm::ChecksumAlgorithmSha256)
        .data_length(EBS_BLOCK_SIZE as i32)
        .send()
        .await?;

    Ok(())
}
```

- API 세부 정보는 AWS SDK for Rust API 참조의 PutSnapshotBlock을 참조하세요.

AWS SDK 또는 CLI와 **CompleteSnapshot** 함께 사용

다음 코드 예시는 CompleteSnapshot의 사용 방법을 보여 줍니다.

Rust

SDK for Rust

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
async fn finish(client: &Client, id: &str) -> Result<(), Error> {
    client
        .complete_snapshot()
        .changed_blocks_count(2)
        .snapshot_id(id)
        .send()
        .await?;

    println!("Snapshot ID {}", id);
    println!("The state is 'completed' when all of the modified blocks have been
 transferred to Amazon S3.");
    println!("Use the get-snapshot-state code example to get the state of the
 snapshot.");

    Ok(())
}
```

- API 세부 정보는 AWS SDK for Rust API 참조의 CompleteSnapshot을 참조하세요.

VPC와 EBS 다이렉트 API 간에 프라이빗 연결 생성

AWS PrivateLink에서 제공되는 인터페이스 VPC 엔드포인트를 생성하여 VPC와 EBS 다이렉트 API 간에 프라이빗 연결을 설정할 수 있습니다. 인터넷 게이트웨이, NAT 디바이스, VPN 연결 또는 AWS Direct Connect 연결을 사용하지 않고 VPC에 있는 것처럼 EBS 다이렉트 API에 액세스할 수 있습니다. VPC의 인스턴스는 EBS 다이렉트 API와 통신하는 데 퍼블릭 IP 주소를 필요로 하지 않습니다.

인터페이스 엔드포인트에 대해 사용 설정하는 각 서브넷에서 엔드포인트 네트워크 인터페이스를 생성합니다.

자세한 내용은 AWS PrivateLink 가이드의를 AWS 서비스 통한 액세스를 AWS PrivateLink 참조하세요.

EBS 다이렉트 API VPC 엔드포인트에 대한 고려 사항

EBS 다이렉트 API에 대한 인터페이스 VPC 엔드포인트를 설정하려면 먼저 AWS PrivateLink 가이드의 Considerations(고려 사항)를 검토합니다.

기본적으로 엔드포인트를 통해 EBS 다이렉트 API에 대한 전체 액세스가 허용됩니다. VPC 엔드포인트 정책을 사용하여 인터페이스 엔드포인트에 대한 액세스를 제어할 수 있습니다. EBS 다이렉트 API에 대한 액세스를 제어하는 엔드포인트 정책을 VPC 엔드포인트에 연결할 수 있습니다. 이 정책은 다음 정보를 지정합니다.

- 작업을 수행할 수 있는 위탁자.
- 수행할 수 있는 작업.
- 작업을 수행할 수 있는 리소스.

자세한 내용은 Amazon VPC 사용 설명서의 VPC 종단점을 통해 서비스에 대한 액세스 제어를 참조하세요.

다음은 EBS 다이렉트 API에 대한 엔드포인트 정책의 예입니다. 이 정책은 엔드포인트에 연결되면 모든 리소스에서 키 `Environment` 및 값 `Test`으로 태그가 지정된 스냅샷을 제외한 모든 리소스에서 모든 EBS 다이렉트 API 작업에 대한 액세스 권한을 부여합니다.

```
{
    "Statement": [
        {
            "Effect": "Deny",
            "Action": "ebs:*",
            "Principal": "*",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/Environment": "Test"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "ebs:*",
            "Principal": "*",
            "Resource": "*"
        }
    ]
}
```

EBS 다이렉트 API에 대한 인터페이스 VPC 엔드포인트 생성

Amazon VPC 콘솔 또는 AWS Command Line Interface (AWS CLI)을 사용하여 EBS 다이렉트 API에 대한 VPC 엔드포인트를 생성할 수 있습니다. 자세한 정보는 AWS PrivateLink 가이드의 VPC 엔드포인트 생성을 참조하세요.

다음 서비스 이름 중 하나를 사용하여 EBS 다이렉트 API용 VPC 엔드포인트를 생성합니다.

- `com.amazonaws.region.ebs`
- `com.amazonaws.region.ebs-fips` - 미국 연방 정보 처리 표준(FIPS) Publication 140-2 US 정부 표준을 준수하는 인터페이스 VPC 엔드포인트를 생성합니다.

> **Note**
>
> FIPS 준수 인터페이스 VPC 엔드포인트는 `us-east-1` | `us-east-2` | `us-west-1` | `us-west-2` | `ca-central-1` | `ca-west-1`.리전에 대해 생성될 수 있습니다. FIPS 준수 인터페이스 VPC 엔드포인트는 IPv4와 IPv6 트래픽을 모두 지원합니다.

엔드포인트에 프라이빗 DNS를 사용하도록 설정하는 경우, 리전에 대한 기본 DNS 이름(예: `ebs.us-east-1.amazonaws.com`)을 사용하여 EBS 다이렉트 API에 API 요청을 할 수 있습니다.

APIs 호출 로깅 AWS CloudTrail

EBS 다이렉APIs는 사용자 AWS CloudTrail, 역할 또는 서비스가 수행한 작업에 대한 레코드를 제공하는 AWS 서비스와 통합됩니다. CloudTrail은 EBS 다이렉트 API에 대한 호출을 이벤트로 캡처합니다. 캡처되는 호출에는의 호출 AWS Management Console 과 EBS 다이렉트 APIs에 대한 코드 호출이 포함됩니다. CloudTrail에서 수집한 정보를 사용하여 EBS 다이렉트 API에 전송된 요청, 요청이 이루어진 IP 주소, 요청이 이루어진 시간, 추가 세부 정보를 확인할 수 있습니다.

모든 이벤트 또는 로그 항목에는 요청을 생성했던 사용자에 관한 정보가 포함됩니다. 자격 증명을 이용하면 다음을 쉽게 판단할 수 있습니다.

- 요청을 루트 사용자로 했는지 사용자 보안 인증으로 했는지 여부.
- IAM Identity Center 사용자를 대신하여 요청이 이루어졌는지 여부입니다.
- 역할 또는 페더레이션 사용자의 임시 자격 증명을 사용하여 요청이 생성되었는지 여부.
- 다른 AWS 서비스에서 요청했는지 여부

CloudTrail은 계정을 생성할 AWS 계정 때에서 활성화되며 CloudTrail 이벤트 기록에 자동으로 액세스할 수 있습니다. CloudTrail 이벤트 기록은 지난 90일 간 AWS 리전의 관리 이벤트에 대해 보기, 검색 및 다운로드가 가능하고, 수정이 불가능한 레코드를 제공합니다. 자세한 설명은 AWS CloudTrail 사용 설명서의 CloudTrail 이벤트 기록 작업을 참조하세요. 이벤트 기록 보기는 CloudTrail 요금이 부과되지 않습니다.

AWS 계정 지난 90일 동안의 이벤트를 지속적으로 기록하려면 추적 또는 CloudTrail Lake 이벤트 데이터 스토어를 생성합니다.

CloudTrail 추적

> CloudTrail은 추적을 사용하여 Amazon S3 버킷으로 로그 파일을 전송할 수 있습니다. 를 사용하여 생성된 모든 추적 AWS Management Console 은 다중 리전입니다. AWS CLI를 사용하여 단일 리전 또는 다중 리전 추적을 생성할 수 있습니다. 계정의 모든 AWS 리전 에서 활동을 캡처하므로 다중 리전 추적을 생성하는 것이 좋습니다. 단일 리전 추적을 생성하는 경우 추적의 AWS 리전에 로깅된 이벤트만 볼 수 있습니다. 추적에 대한 자세한 내용은 AWS CloudTrail 사용 설명서의 AWS 계정에 대한 추적 생성 및 조직에 대한 추적 생성을 참조하세요.

CloudTrail에서 추적을 생성하여 진행 중인 관리 이벤트의 사본 하나를 Amazon S3 버킷으로 무료로 전송할 수는 있지만, Amazon S3 스토리지 요금이 부과됩니다. CloudTrail 요금에 관한 자세한 내용은 AWS CloudTrail 요금을 참조하세요. Amazon S3 요금에 관한 자세한 내용은 Amazon S3 요금을 참조하세요.

CloudTrail Lake 이벤트 데이터 스토어

CloudTrail Lake를 사용하면 이벤트에 대해 SQL 기반 쿼리를 실행할 수 있습니다. CloudTrail Lake는 행 기반 JSON 형식의 기존 이벤트를 Apache ORC 형식으로 변환합니다. ORC는 빠른 데이터 검색에 최적화된 열 기반 스토리지 형식입니다. 이벤트는 이벤트 데이터 스토어로 집계되며, 이벤트 데이터 스토어는 고급 이벤트 선택기를 적용하여 선택한 기준을 기반으로 하는 변경 불가능한 이벤트 컬렉션입니다. 이벤트 데이터 스토어에 적용하는 선택기는 어떤 이벤트가 지속되고 쿼리에 사용 가능한지를 제어합니다. CloudTrail Lake에 대한 자세한 내용은 AWS CloudTrail 사용 설명서의 AWS CloudTrail Lake 작업을 참조하세요.

CloudTrail Lake 이벤트 데이터 스토어 및 쿼리에는 비용이 발생합니다. 이벤트 데이터 스토어를 생성할 때 이벤트 데이터 스토어에 사용할 요금 옵션을 선택합니다. 요금 옵션에 따라 이벤트 모으기 및 저장 비용과 이벤트 데이터 스토어의 기본 및 최대 보존 기간이 결정됩니다. CloudTrail 요금에 관한 자세한 내용은 AWS CloudTrail 요금을 참조하세요.

CloudTrail의 EBS 다이렉트 API 데이터 이벤트

데이터 이벤트는 리소스 상에서, 또는 리소스 내에서 수행되는 리소스 작업에 대한 정보를 제공합니다. 이를 데이터 플레인 작업이라고도 합니다. 데이터 이벤트는 흔히 대량 활동입니다. 기본적으로 CloudTrail은 데이터 이벤트를 로깅하지 않습니다. CloudTrail 이벤트 기록은 데이터 이벤트를 기록하지 않습니다.

데이터 이벤트에는 추가 요금이 적용됩니다. CloudTrail 요금에 관한 자세한 내용은 AWS CloudTrail 요금을 참조하세요.

CloudTrail 콘솔 AWS CLI또는 CloudTrail APIs 작업을 사용하여 EBS 다이렉트 API 리소스 유형에 대한 데이터 이벤트를 로깅할 수 있습니다. 데이터 이벤트를 로깅하는 방법에 관한 자세한 내용은 AWS CloudTrail 사용 설명서의 AWS Management Console을 사용한 데이터 이벤트 로깅 및 AWS Command Line Interface를 사용한 이벤트 데이터 로깅을 참조하세요.

다음 EBS 다이렉트 API 작업을 데이터 이벤트로 로깅할 수 있습니다.

- ListSnapshotBlocks
- ListChangedBlocks

- GetSnapshotBlock
- PutSnapshotBlock

> **Note**
>
> 공유된 스냅샷에 대해 작업을 수행하면 데이터 이벤트가 스냅샷을 소유한 AWS 계정으로 전송되지 않습니다.

CloudTrail의 EBS 다이렉트 API 관리 이벤트

관리 이벤트는의 리소스에서 수행되는 관리 작업에 대한 정보를 제공합니다 AWS 계정. 이를 컨트롤 플레인 작업이라고도 합니다. 기본적으로 CloudTrail은 관리 이벤트를 로깅합니다.

EBS 다이렉트 API 서비스는 다음과 같은 컨트롤 플레인 작업을 관리 이벤트로 CloudTrail에 로깅합니다.

- StartSnapshot
- CompleteSnapshot

EBS 다이렉트 API 이벤트 예제

이벤트는 모든 소스로부터의 단일 요청을 나타내며 요청된 API 작업, 작업 날짜와 시간, 요청 파라미터 등에 관한 정보가 들어 있습니다. CloudTrail 로그 파일은 퍼블릭 API 직접 호출의 주문 스택 추적이 아니므로 이벤트가 특정 순서로 표시되지 않습니다.

다음은 EBS 다이렉트 API에 대한 CloudTrail 이벤트의 예입니다.

StartSnapshot

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "123456789012",
        "arn": "arn:aws:iam::123456789012:root",
        "accountId": "123456789012",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "userName": "user"
```

```
    },
    "eventTime": "2020-07-03T23:27:26Z",
    "eventSource": "ebs.amazonaws.com",
    "eventName": "StartSnapshot",
    "awsRegion": "eu-west-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "PostmanRuntime/7.25.0",
    "requestParameters": {
        "volumeSize": 8,
        "clientToken": "token",
        "encrypted": true
    },
    "responseElements": {
        "snapshotId": "snap-123456789012",
        "ownerId": "123456789012",
        "status": "pending",
        "startTime": "Jul 3, 2020 11:27:26 PM",
        "volumeSize": 8,
        "blockSize": 524288,
        "kmsKeyArn": "HIDDEN_DUE_TO_SECURITY_REASONS"
    },
    "requestID": "be112233-1ba5-4ae0-8e2b-1c302EXAMPLE",
    "eventID": "6e12345-2a4e-417c-aa78-7594fEXAMPLE",
    "eventType": "AwsApiCall",
    "recipientAccountId": "123456789012"
}
```

CompleteSnapshot

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "123456789012",
        "arn": "arn:aws:iam::123456789012:root",
        "accountId": "123456789012",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "userName": "user"
    },
    "eventTime": "2020-07-03T23:28:24Z",
    "eventSource": "ebs.amazonaws.com",
    "eventName": "CompleteSnapshot",
    "awsRegion": "eu-west-1",
```

```
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "PostmanRuntime/7.25.0",
    "requestParameters": {
        "snapshotId": "snap-123456789012",
        "changedBlocksCount": 5
    },
    "responseElements": {
        "status": "completed"
    },
    "requestID": "be112233-1ba5-4ae0-8e2b-1c302EXAMPLE",
    "eventID": "6e12345-2a4e-417c-aa78-7594fEXAMPLE",
    "eventType": "AwsApiCall",
    "recipientAccountId": "123456789012"
}
```

ListSnapshotBlocks

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "AIDAT4HPB2AO3JEXAMPLE",
        "arn": "arn:aws:iam::123456789012:user/user",
        "accountId": "123456789012",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "userName": "user"
    },
    "eventTime": "2021-06-03T00:32:46Z",
    "eventSource": "ebs.amazonaws.com",
    "eventName": "ListSnapshotBlocks",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "111.111.111.111",
    "userAgent": "PostmanRuntime/7.28.0",
    "requestParameters": {
        "snapshotId": "snap-abcdef01234567890",
        "maxResults": 100,
        "startingBlockIndex": 0
    },
    "responseElements": null,
    "requestID": "example6-0e12-4aa9-b923-1555eexample",
    "eventID": "example4-218b-4f69-a9e0-2357dexample",
    "readOnly": true,
    "resources": [
```

```
        {
            "accountId": "123456789012",
            "type": "AWS::EC2::Snapshot",
            "ARN": "arn:aws:ec2:us-west-2::snapshot/snap-abcdef01234567890"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": false,
    "recipientAccountId": "123456789012",
    "eventCategory": "Data",
    "tlsDetails": {
        "tlsVersion": "TLSv1.2",
        "cipherSuite": "ECDHE-RSA-AES128-SHA",
        "clientProvidedHostHeader": "ebs.us-west-2.amazonaws.com"
    }
}
```

ListChangedBlocks

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "AIDAT4HPB2AO3JEXAMPLE",
        "arn": "arn:aws:iam::123456789012:user/user",
        "accountId": "123456789012",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "userName": "user"
    },
    "eventTime": "2021-06-02T21:11:46Z",
    "eventSource": "ebs.amazonaws.com",
    "eventName": "ListChangedBlocks",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "111.111.111.111",
    "userAgent": "PostmanRuntime/7.28.0",
    "requestParameters": {
        "firstSnapshotId": "snap-abcdef01234567890",
        "secondSnapshotId": "snap-9876543210abcdef0",
        "maxResults": 100,
        "startingBlockIndex": 0
    },
    "responseElements": null,
    "requestID": "example0-f4cb-4d64-8d84-72e1bexample",
```

```
    "eventID": "example3-fac4-4a78-8ebb-3e9d3example",
    "readOnly": true,
    "resources": [
        {
            "accountId": "123456789012",
            "type": "AWS::EC2::Snapshot",
            "ARN": "arn:aws:ec2:us-west-2::snapshot/snap-abcdef01234567890"
        },
       {
            "accountId": "123456789012",
            "type": "AWS::EC2::Snapshot",
            "ARN": "arn:aws:ec2:us-west-2::snapshot/snap-9876543210abcdef0"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": false,
    "recipientAccountId": "123456789012",
    "eventCategory": "Data",
    "tlsDetails": {
        "tlsVersion": "TLSv1.2",
        "cipherSuite": "ECDHE-RSA-AES128-SHA",
        "clientProvidedHostHeader": "ebs.us-west-2.amazonaws.com"
    }
}
```

GetSnapshotBlock

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "AIDAT4HPB2AO3JEXAMPLE",
        "arn": "arn:aws:iam::123456789012:user/user",
        "accountId": "123456789012",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "userName": "user"
    },
    "eventTime": "2021-06-02T20:43:05Z",
    "eventSource": "ebs.amazonaws.com",
    "eventName": "GetSnapshotBlock",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "111.111.111.111",
    "userAgent": "PostmanRuntime/7.28.0",
```

```
    "requestParameters": {
        "snapshotId": "snap-abcdef01234567890",
        "blockIndex": 1,
        "blockToken": "EXAMPLEiL5E3pMPFpaDWjExM2/mnSKh1mQfcbjwe2mM7EwhrgCdPAEXAMPLE"
    },
    "responseElements": null,
    "requestID": "examplea-6eca-4964-abfd-fd9f0example",
    "eventID": "example6-4048-4365-a275-42e94example",
    "readOnly": true,
    "resources": [
        {
          "accountId": "123456789012",
          "type": "AWS::EC2::Snapshot",
          "ARN": "arn:aws:ec2:us-west-2::snapshot/snap-abcdef01234567890"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": false,
    "recipientAccountId": "123456789012",
    "eventCategory": "Data",
    "tlsDetails": {
        "tlsVersion": "TLSv1.2",
        "cipherSuite": "ECDHE-RSA-AES128-SHA",
        "clientProvidedHostHeader": "ebs.us-west-2.amazonaws.com"
    }
}
```


PutSnapshotBlock


```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "AIDAT4HPB2AO3JEXAMPLE",
        "arn": "arn:aws:iam::123456789012:user/user",
        "accountId": "123456789012",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "userName": "user"
    },
    "eventTime": "2021-06-02T21:09:17Z",
    "eventSource": "ebs.amazonaws.com",
    "eventName": "PutSnapshotBlock",
    "awsRegion": "us-east-1",
```

```
    "sourceIPAddress": "111.111.111.111",
    "userAgent": "PostmanRuntime/7.28.0",
    "requestParameters": {
        "snapshotId": "snap-abcdef01234567890",
        "blockIndex": 1,
        "dataLength": 524288,
        "checksum": "exampleodSGvFSb1e3kxWUgbOQ4TbzPurnsfVexample",
        "checksumAlgorithm": "SHA256"
    },
    "responseElements": {
        "checksum": "exampleodSGvFSb1e3kxWUgbOQ4TbzPurnsfVexample",
        "checksumAlgorithm": "SHA256"
    },
    "requestID": "example3-d5e0-4167-8ee8-50845example",
    "eventID": "example8-4d9a-4aad-b71d-bb31fexample",
    "readOnly": false,
    "resources": [
        {
            "accountId": "123456789012",
            "type": "AWS::EC2::Snapshot",
            "ARN": "arn:aws:ec2:us-west-2::snapshot/snap-abcdef01234567890"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": false,
    "recipientAccountId": "123456789012",
    "eventCategory": "Data",
    "tlsDetails": {
        "tlsVersion": "TLSv1.2",
        "cipherSuite": "ECDHE-RSA-AES128-SHA",
        "clientProvidedHostHeader": "ebs.us-west-2.amazonaws.com"
    }
}
```


CloudTrail 레코드 콘텐츠에 관한 자세한 내용은 AWS CloudTrail 사용 설명서의 CloudTrail 레코드 콘텐츠를 참조하세요.

EBS 다이렉트 API FAQ

대기 중 상태일 때 EBS 다이렉트 API를 사용하여 스냅샷에 액세스할 수 있습니까?

아니요, 완료됨 상태인 경우에만 스냅샷에 액세스할 수 있습니다.

EBS 다이렉트 API에서 번호순으로 블록 인덱스를 반환합니까?

예. 반환되는 블록 인덱스는 고유하며 번호순입니다.

MaxResults 파라미터 값이 100 미만인 요청을 제출할 수 있습니까?

아니요, 사용할 수 있는 최소 MaxResult 파라미터 값은 100입니다. MaxResult 파라미터 값이 100 미만인 요청을 제출할 경우 스냅샷에 있는 블록이 100개를 초과하면 API가 최소한 100개의 결과를 반환합니다.

API 요청을 동시에 실행할 수 있습니까?

API 요청을 동시에 실행할 수 있습니다. 병목 현상을 방지하려면 계정에서 실행될 수 있는 다른 워크로드를 고려해야 합니다. 또한 조절, 시간 초과 및 서비스 사용 불가를 처리하려면 EBS 다이렉트 API 워크플로우에 재시도 메커니즘을 빌드해야 합니다. 자세한 내용은 EBS 다이렉트 API의 성능 최적화 섹션을 참조하세요.

EBS 다이렉트 API 서비스 할당량을 검토하여 초당 실행할 수 있는 API 요청을 확인합니다. 자세한 내용은 AWS 일반 참조에서 Amazon Elastic Block Store 엔드포인트 및 할당량을 참조하세요.

ListChangedBlocks 작업을 실행할 때 스냅샷에 블록이 있어도 빈 응답을 받을 수 있습니까?

예. 스냅샷에 변경된 블록이 거의 없으면 응답이 비어 있을 수 있지만 API가 다음 페이지 토큰 값을 반환합니다. 다음 페이지 토큰 값을 사용하여 결과의 다음 페이지로 계속 진행하세요. API가 다음 페이지 토큰 값 null을 반환하면 마지막 결과 페이지에 도달했음을 확인할 수 있습니다.

NextToken 파라미터가 StartingBlockIndex 파라미터와 함께 지정되면 둘 중 어느 것이 사용됩니까?

NextToken이 사용되며 StartingBlockIndex는 무시됩니다.

블록 토큰과 다음 토큰은 얼마 동안 유효합니까?

블록 토큰은 7일간 유효하고 다음 토큰은 60분간 유효합니다.

암호화된 스냅샷이 지원됩니까?

예. EBS 다이렉트 API를 사용하여 암호화된 스냅샷에 액세스할 수 있습니다.

암호화된 스냅샷에 액세스하려면 스냅샷을 암호화하는 데 사용되는 KMS 키와 AWS KMS 복호화 작업에 액세스할 수 있어야 합니다. 사용자에게 할당할 AWS KMS 정책은이 가이드 앞부분의 IAM을 사용하여 EBS 다이렉트 API에 대한 액세스 제어 섹션을 참조하세요.

퍼블릭 스냅샷이 지원됩니까?

퍼블릭 스냅샷이 지원되지 않습니다.

의 Amazon EBS 로컬 스냅샷이 AWS Outposts 지원되나요?

의 Amazon EBS 로컬 스냅샷 AWS Outposts 은 지원되지 않습니다.

목록 스냅샷 블록은 스냅샷의 모든 블록 인덱스와 블록 토큰을 반환합니까? 아니면 데이터가 쓰여진 블록 인덱스와 블록 토큰만 반환합니까?

데이터가 쓰여진 블록 인덱스와 토큰만 반환합니다.

보안 분석 및 운영 문제 해결 목적으로 내 계정에서 이루어진 모든 EBS 다이렉트 API API 호출 기록을 얻을 수 있습니까?

예. 계정에서 수행된 EBS 다이렉트 API 호출 기록을 수신하려면 AWS Management Console에서 AWS CloudTrail 을 활성화합니다. 자세한 내용은 APIs 호출 로깅 AWS CloudTrail 단원을 참조하십시오.

휴지통을 사용하여 삭제된 EBS 볼륨, EBS 스냅샷 및 EBS 지원 AMIs 복구

Recycle Bin은 실수로 삭제된 EBS 볼륨, EBS 스냅샷 및 EBS 기반 AMI를 복원할 수 있는 데이터 복구 기능입니다. 휴지통을 사용할 때 리소스가 삭제되면 영구적으로 삭제되기 전에 지정한 기간 동안 휴지통에 보관됩니다.

보존 기간이 만료되기 전에 언제든지 휴지통에서 리소스를 복원할 수 있습니다. 휴지통에서 리소스를 복원하면 해당 리소스가 휴지통에서 제거되며 계정에서 해당 유형의 다른 리소스를 사용하는 것과 동일한 방식으로 리소스를 사용할 수 있습니다. 보존 기간이 만료되고 리소스가 복원되지 않으면 휴지통에서 리소스가 영구적으로 삭제되고 더 이상 복원할 수 없습니다.

휴지통을 사용하면 우발적인 삭제로부터 비즈니스 크리티컬 데이터를 보호하여 비즈니스 연속성을 보장하는 데 도움이 됩니다.

휴지통은 Amazon Elastic Block Store(Amazon EBS)의 서비스 기능으로 평가됩니다. Amazon EBS를 나열하는 규정 준수 프로그램(FedRAMP, HIPAA BAA, SOC 등)의 범위 내 모든AWS 서비스도 휴지통에 적용됩니다.

주제

- 지원되는 리소스
- 휴지통은 어떻게 작동합니까?
- 휴지통 고려 사항
- 할당량
- 관련 서비스
- 가격 책정
- IAM을 사용하여 휴지통에 대한 액세스 제어
- 휴지통 보존 규칙 생성
- 기존 휴지통 보존 규칙 업데이트
- 휴지통 보존 규칙을 잠가 업데이트 또는 삭제를 방지합니다.
- 휴지통 보존 규칙을 업데이트 또는 삭제할 수 있도록 잠금 해제
- 휴지통 보존 규칙에 태그 지정
- 리소스를 보존하지 않도록 휴지통 보존 규칙 삭제

- 휴지통에서 삭제된 스냅샷 복구
- 휴지통에서 삭제된 볼륨 복구
- 삭제된 AMI를 휴지통에서 복구
- Amazon EventBridge를 사용하여 휴지통 모니터링
- 를 사용하여 휴지통 모니터링 AWS CloudTrail
- 휴지통의 서비스 엔드포인트
- VPC와 휴지통 간에 프라이빗 연결 생성

지원되는 리소스

휴지통은 다음과 같은 리소스 유형을 지원합니다.

- Amazon EBS 볼륨
- Amazon EBS snapshots

> **Important**
>
> 휴지통 보존 규칙은 아카이브 스토리지 티어의 아카이빙된 스냅샷에도 적용됩니다. 보존 규칙과 일치하는 아카이빙된 스냅샷을 삭제하면 보존 규칙에 정의된 기간 동안 해당 스냅샷이 휴지통에 보존됩니다. 아카이빙된 스냅샷은 휴지통에 있는 동안 아카이빙된 스냅샷에 대한 요금으로 청구됩니다.

- Amazon EBS 지원 Amazon Machine Image(AMI)

> **Note**
>
> 보존 규칙은 비활성화된 AMI에도 적용됩니다.

휴지통은 어떻게 작동합니까?

휴지통을 활성화하고 사용하려면 리소스를 보호하려는 AWS 리전에서 보존 규칙을 생성해야 합니다. 보존 규칙은 다음을 지정합니다.

- 보호하려는 리소스 유형(볼륨, 스냅샷 또는 AMIs).
- 보존 규칙의 유형:

  - 태그 수준 보존 규칙 - 이러한 보존 규칙은 리소스 태그를 사용하여 보호할 리소스를 식별합니다. 각 보존 규칙에 대해 하나 이상의 태그 키 및 값 페어를 지정합니다. 이러한 태그 키 및 값 페어 중 하나 이상이 있는 리소스(지정된 유형)의 삭제 시 휴지통에 자동으로 유지됩니다. 이 유형의 보존 규칙을 사용하여 태그를 기준으로 계정의 특정 리소스를 보호합니다.
  - 리전 수준 보존 규칙 - 리소스에 태그가 지정되지 않은 경우에도 기본적으로 이러한 보존 규칙은 리전의 모든 리소스(지정된 유형)에 적용됩니다. 하지만 제외 태그를 지정하여 특정 태그가 있는 리소스를 제외할 수 있습니다. 리전에서 특정 유형의 모든 리소스를 보호하려면 이 유형의 보존 규칙을 사용합니다.
- 리소스가 삭제된 후 보존할 보존 기간. 이 기간이 만료되면 리소스가 휴지통에서 영구적으로 삭제됩니다. 지원되는 보존 기간은 다음과 같습니다.
  - EBS 볼륨: 1~7일
  - EBS 스냅샷 및 EBS 지원 AMIs: 1~365일

리소스가 휴지통에 있는 동안에는 언제든지 사용을 위해 복원할 수 있습니다. 리소스는 다음 중 하나가 발생할 때까지 휴지통에 남아 있습니다.

- 사용을 위해 수동으로 복원합니다. 휴지통에서 리소스를 복원하면 리소스가 휴지통에서 제거되고 즉시 사용할 수 있습니다. 계정에서 해당 유형의 다른 리소스와 동일한 방식으로 복원된 리소스를 사용할 수 있습니다.
- 보존 기간이 만료됩니다. 보존 기간이 만료되고 리소스가 휴지통에서 복원되지 않은 경우 리소스는 휴지통에서 영구적으로 삭제되고 더 이상 보거나 복원할 수 없습니다.

휴지통 고려 사항

휴지통 및 보존 규칙 작업 시 다음 고려 사항이 적용됩니다.

일반적인 고려 사항

- 삭제된 리소스는 기존 보존 규칙과 일치하는 경우에만 휴지통으로 이동됩니다. 보존 규칙과 일치하지 않는 리소스를 삭제하거나 해당 시점에 보존 규칙이 없는 경우 해당 리소스는 휴지통으로 이동되지 않고 영구적으로 삭제됩니다.
-

> **Important**
>
> 보존 규칙은 계정의 리전별로 리소스 유형별로 생성된 첫 번째 보존 규칙에 대한 최종 일관성 모델을 따릅니다. 리전의 리소스 유형에 대한 첫 번째 보존 규칙을 생성하면 해당 규칙이 활성화되지 않고 리소스 보존이 즉시 시작될 수 있습니다. 그러나 동일한 리전에서 동일한

리소스 유형에 대해 생성한 후속 보존 규칙은 활성화되고 거의 즉시 리소스를 유지하기 시작합니다.

- 리소스가 삭제 시 하나 이상의 보존 규칙과 일치하는 경우 보존 기간이 가장 긴 보존 규칙이 우선합니다.
- 휴지통에서 수동으로 리소스를 삭제할 수 없습니다. 보존 기간이 만료되면 리소스가 자동으로 삭제됩니다.
- 리소스가 휴지통에 있는 동안에는 리소스를 보거나 복원하거나 해당 태그를 수정할 수만 있습니다. 리소스를 다른 방식으로 사용하려면 먼저 리소스를 복원해야 합니다.
- AWS Backup 또는 Amazon Data Lifecycle Manager AWS 서비스와 같이 보존 규칙과 일치하는 리소스가 있는 경우 해당 리소스는 휴지통에 의해 자동으로 보존됩니다. 필요한 경우 이러한 리소스에 태그를 지정한 다음 이러한 태그를 보존 규칙에 제외 태그로 추가하여 삭제 시 해당 리소스가 휴지통에 들어가지 않도록 할 수 있습니다.
- 리소스를 휴지통으로 보내면 다음과 같은 시스템 생성 태그가 리소스에 할당됩니다.
  - 태그 키 - `aws:recycle-bin:resource-in-bin`
  - 태그 값 - `true`

  이 태그는 수동으로 편집하거나 삭제할 수 없습니다. 리소스가 휴지통에서 복원되면 태그가 자동으로 제거됩니다.

볼륨에 대한 고려 사항

- 인스턴스 종료 또는 루트 볼륨 교체로 인해 삭제된 볼륨은 휴지통으로 보호됩니다.
- 생성하지 못한 볼륨은 삭제 시 휴지통으로 보호되지 않습니다.
- 실패한 인스턴스 시작 볼륨은 삭제 시 휴지통으로 보호되지 않습니다.
- 관리형 인스턴스의 볼륨은 삭제 시 휴지통으로 보호되지 않습니다.
- 볼륨이 휴지통에 들어가면 진행 중인 볼륨 생성 또는 수정이 일시 중지되지 않습니다. 즉, 볼륨이 볼륨 초기화에 대한 Amazon EBS 프로비저닝 요율로 생성된 경우에도 그에 따라 요금이 청구됩니다.
- 휴지통의 볼륨은 일반 볼륨과 동일한 방식으로 할당량에 포함됩니다.
- 휴지통 종료 시간이 경과한 후에는 휴지통의 볼륨에 요금이 청구되지 않습니다. 이러한 볼륨은 복원할 수 없지만 아직 삭제되지 않은 경우 검색할 수 있습니다.
- `deleteVolume` 이벤트는 볼륨이 휴지통에서 삭제된 후에만 전송됩니다. 이 이벤트는 볼륨이 휴지통에 들어갈 때 내보내지지 않습니다.

스냅샷에 대한 고려 사항

- > **Important**
  >
  > AMI 및 연결된 스냅샷에 대한 보존 규칙이 있는 경우 스냅샷의 보존 기간을 AMI의 보존 기간과 같거나 더 길게 설정합니다. 이렇게 하면 AMI를 복구할 수 없게 되므로 AMI 자체를 삭제하기 전에 휴지통에서 AMI와 연결된 스냅샷을 삭제하지 않습니다.
- 스냅샷이 삭제될 때 빠른 스냅샷 복원을 사용하도록 설정하면 스냅샷이 휴지통으로 이동된 직후 빠른 스냅샷 복원이 자동으로 비활성화됩니다.
  - 스냅샷에 대해 빠른 스냅샷 복원이 비활성화되기 전에 스냅샷을 복원하면 빠른 스냅샷 복원은 활성화된 상태로 유지됩니다.
  - 빠른 스냅샷 복원이 비활성화된 후 스냅샷을 복원하면 빠른 스냅샷 복원은 비활성화된 상태로 유지됩니다. 필요한 경우 빠른 스냅샷 복원을 수동으로 다시 활성화해야 합니다.
- 공유된 스냅샷을 삭제하면 스냅샷이 휴지통으로 이동될 때 자동으로 공유 해제됩니다. 스냅샷을 복원하면 이전 공유 권한이 모두 자동으로 복원됩니다.
- 와 같이 다른 AWS 서비스에서 생성한 스냅샷 AWS Backup 이 휴지통으로 전송되고 나중에 휴지통에서 해당 스냅샷을 복원하는 경우 해당 스냅샷은 더 이상 생성한 AWS 서비스에서 관리되지 않습니다. 더 이상 필요하지 않은 경우 스냅샷을 수동으로 삭제해야 합니다.

AMI에 대한 고려 사항

- Amazon EBS 지원 AMI만 지원됩니다.
- > **Important**
  >
  > AMI 및 연결된 스냅샷에 대한 보존 규칙이 있는 경우 스냅샷의 보존 기간을 AMI의 보존 기간과 같거나 더 길게 설정합니다. 이렇게 하면 AMI를 복구할 수 없게 되므로 AMI 자체를 삭제하기 전에 휴지통에서 AMI와 연결된 스냅샷을 삭제하지 않습니다.
- 공유된 AMI를 삭제하면 AMI가 휴지통으로 이동될 때 자동으로 공유 해제됩니다. AMI를 복원하면 이전 공유 권한이 모두 자동으로 복원됩니다.
- 휴지통에서 AMI를 복원하려면 먼저 휴지통에서 연결된 모든 스냅샷을 복원하고 해당 스냅샷이 `available` 상태인지 확인해야 합니다.
- AMI와 연결된 스냅샷이 휴지통에서 삭제되면 AMI는 더 이상 복구할 수 없습니다. 보존 기간이 만료되면 AMI가 삭제됩니다.

- AWS 백업과 같은 다른 AWS 서비스에 의해 생성된 AMI가 휴지통으로 전송되고 나중에 휴지통에서 해당 AMI를 복원하는 경우 해당 AMI는 더 이상 해당 AMI를 생성한 AWS 서비스에서 관리되지 않습니다. 더 이상 필요하지 않은 경우 마지막 AMI를 수동으로 삭제해야 합니다.

Amazon Data Lifecycle Manager 스냅샷 정책에 대한 고려 사항

- 보존 규칙과 일치하는 스냅샷을 Amazon Data Lifecycle Manager에서 삭제하면 해당 스냅샷은 휴지통에 의해 자동으로 유지됩니다.
- 정책의 보존 임계값에 도달하면 Amazon Data Lifecycle Manager가 스냅샷을 삭제하고 휴지통으로 보내고 사용자가 휴지통에서 스냅샷을 수동으로 복원하는 경우 더 이상 필요하지 않을 때 해당 스냅샷을 수동으로 삭제해야 합니다. Amazon Data Lifecycle Manager는 더 이상 스냅샷을 관리하지 않습니다.
- 정책에 의해 생성된 스냅샷을 수동으로 삭제하고 정책의 보존 임계값에 도달했을 때 해당 스냅샷이 휴지통에 있는 경우 Amazon Data Lifecycle Manager는 스냅샷을 삭제하지 않습니다. Amazon Data Lifecycle Manager는 휴지통에 저장된 스냅샷을 관리하지 않습니다.

  정책의 보존 임계값에 도달하기 전에 스냅샷이 휴지통에서 복원되는 경우 정책의 보존 임계값에 도달하면 Amazon Data Lifecycle Manager가 스냅샷을 삭제합니다.

  정책의 보존 임계값에 도달한 후 스냅샷이 휴지통에서 복원되면 Amazon Data Lifecycle Manager가 더 이상 스냅샷을 삭제하지 않습니다. 더 이상 필요하지 않은 스냅샷은 수동으로 삭제해야 합니다.

AWS 백업 고려 사항

- AWS Backup이 보존 규칙과 일치하는 스냅샷을 삭제하면 해당 스냅샷은 휴지통에 의해 자동으로 보존됩니다.

아카이빙된 스냅샷 고려 사항

- 휴지통 보존 규칙은 아카이브 스토리지 티어의 아카이빙된 스냅샷에도 적용됩니다. 보존 규칙과 일치하는 아카이빙된 스냅샷을 삭제하면 보존 규칙에 정의된 기간 동안 해당 스냅샷이 휴지통에 보존됩니다.

  아카이빙된 스냅샷은 휴지통에 있는 동안 아카이빙된 스냅샷에 대한 요금으로 청구됩니다.

  즉, 최소 아카이브 기간인 90일 전에 아카이빙된 스냅샷을 보존 규칙에 따라 휴지통에서 삭제하면 남은 일수에 대한 요금이 청구됩니다. 자세한 내용은 아카이브된 스냅샷 요금 및 결제를 참조하세요.

휴지통에 있는 아카이빙된 스냅샷을 사용하려면 먼저 휴지통에서 스냅샷을 복구한 다음에 아카이브 티어에서 표준 티어로 복원해야 합니다.

할당량

다음 할당량이 휴지통에 적용됩니다.

할당량	기본 할당량			
리전별 보존 규칙	250			
보존 규칙별 태그 키 및 값 페어	50			

관련 서비스

휴지통은 다음과 같은 서비스와 함께 작동합니다.

- AWS CloudTrail - 휴지통에서 발생하는 이벤트를 기록할 수 있습니다. 자세한 내용은 를 사용하여 휴지통 모니터링 AWS CloudTrail 단원을 참조하십시오.

가격 책정

휴지통 및 보존 규칙 사용에 따른 추가 요금은 없습니다. 자세한 내용은 Amazon EBS 요금을 참조하세요.

- Amazon EBS 볼륨 - 휴지통의 볼륨은 계정의 일반 볼륨과 동일한 요금으로 청구됩니다.
- Amazon EBS 스냅샷 - 휴지통의 스냅샷에는 계정의 일반 스냅샷과 동일한 요금이 청구됩니다.
- EBS 지원 AMI - 휴지통의 AMI에는 추가 요금이 발생하지 않습니다.

> **Note**
>
> 일부 리소스는 보존 기간이 만료되고 영구적으로 삭제된 후 잠시 동안 휴지통 콘솔 또는 AWS CLI 및 API 출력에 계속 표시될 수 있습니다. 이러한 리소스에 대해서는 요금이 청구되지 않습니다. 보존 기간이 만료되는 즉시 청구가 중지됩니다.

를 사용할 때 비용 추적 및 할당 목적으로 다음과 같이 AWS 생성된 비용 할당 태그를 사용할 수 있습니다 AWS 결제 및 비용 관리.

- 키: `aws:recycle-bin:resource-in-bin`
- 값: `true`

자세한 내용은 AWS 결제 및 비용 관리 사용 설명서의 AWS에서 생성되는 비용 할당 태그를 참조하세요.

IAM을 사용하여 휴지통에 대한 액세스 제어

기본적으로 사용자는 휴지통, 보존 규칙 또는 휴지통에 있는 리소스로 작업할 수 있는 권한이 없습니다. 사용자가 이러한 리소스로 작업하도록 허용하려면 특정 리소스 및 API 작업을 사용할 권한을 부여하는 IAM 정책을 생성해야 합니다. 정책이 생성된 후에는 사용자, 그룹 또는 역할에 권한을 추가해야 합니다.

주제

- 휴지통 및 보존 규칙 작업을 위한 권한
- 휴지통의 리소스 작업을 위한 권한
- 휴지통에 사용되는 조건 키

휴지통 및 보존 규칙 작업을 위한 권한

휴지통과 보존 규칙을 사용하려면 사용자에게 다음 권한이 필요합니다.

- `rbin:CreateRule`
- `rbin:UpdateRule`
- `rbin:GetRule`
- `rbin:ListRules`

- rbin:DeleteRule
- rbin:TagResource
- rbin:UntagResource
- rbin:ListTagsForResource
- rbin:LockRule
- rbin:UnlockRule

휴지통 콘솔을 사용하려면 사용자에게 tag:GetResources 권한이 필요합니다.

다음은 콘솔 사용자의 tag:GetResources 권한을 포함하는 IAM 정책의 예입니다. 일부 권한이 필요하지 않은 경우 정책에서 권한을 제거할 수 있습니다.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "rbin:CreateRule",
                "rbin:UpdateRule",
                "rbin:GetRule",
                "rbin:ListRules",
                "rbin:DeleteRule",
                "rbin:TagResource",
                "rbin:UntagResource",
                "rbin:ListTagsForResource",
                "rbin:LockRule",
                "rbin:UnlockRule",
                "tag:GetResources"
            ],
            "Resource": "*"
        }
    ]
}
```

액세스 권한을 제공하려면 사용자, 그룹 또는 역할에 권한을 추가하세요.

- 의 사용자 및 그룹 AWS IAM Identity Center:

  권한 세트를 생성합니다. AWS IAM Identity Center 사용자 안내서에서 권한 세트 생성의 지침을 따릅니다.
- ID 제공업체를 통해 IAM에서 관리되는 사용자:

  ID 페더레이션을 위한 역할을 생성합니다. IAM 사용자 설명서의 Create a role for a third-party identity provider (federation)의 지침을 따릅니다.
- IAM 사용자:
  - 사용자가 맡을 수 있는 역할을 생성합니다. IAM 사용자 설명서에서 Create a role for an IAM user의 지침을 따릅니다.
  - (권장되지 않음) 정책을 사용자에게 직접 연결하거나 사용자를 사용자 그룹에 추가합니다. IAM 사용 설명서에서 사용자(콘솔)에 권한 추가의 지침을 따릅니다.

휴지통의 리소스 작업을 위한 권한

휴지통의 리소스 작업에 필요한 IAM 권한에 대한 자세한 내용은 다음을 참조하세요.

- 휴지통의 볼륨 작업에 대한 권한
- 휴지통의 스냅샷 작업을 위한 권한
- 휴지통의 AMI 작업을 위한 권한

휴지통에 사용되는 조건 키

휴지통은 IAM 정책의 `Condition` 요소에서 정책 설명이 적용되는 조건을 제어하는 데 사용할 수 있는 다음의 조건 키를 정의합니다. 자세한 내용은 IAM 사용 설명서의 IAM JSON 정책 요소: 조건을 참조하세요.

주제

- rbin:Request/ResourceType 조건 키
- rbin:Attribute/ResourceType 조건 키

`rbin:Request/ResourceType` 조건 키

이 `rbin:Request/ResourceType` 조건 키는 `ResourceType` 요청 파라미터에 지정된 값을 기반으로 CreateRule 및 ListRules 요청에 대한 액세스를 필터링하는 데 사용될 수 있습니다.

예제 1 - CreateRule

다음 샘플 IAM 정책은 ResourceType 요청 파라미터에 지정된 값이 EBS_SNAPSHOT 또는 EC2_IMAGE일 때만 IAM 보안 주체가 CreateRule 요청을 하도록 허용합니다. 이렇게 하면 보안 주체가 스냅샷 및 AMI에 대해서만 새 보존 규칙을 생성할 수 있습니다.

JSON

```
{
    "Version":"2012-10-17",
    "Statement" : [
        {
            "Effect" : "Allow",
            "Action" :[
                "rbin:CreateRule"
            ],
            "Resource" : "*",
            "Condition" : {
                "StringEquals" : {
                    "rbin:Request/ResourceType" : ["EBS_SNAPSHOT", "EC2_IMAGE"]
                }
            }
        }
    ]
}
```

예제 2 - ListRules

다음 샘플 IAM 정책은 ResourceType 요청 파라미터에 지정된 값이 EBS_SNAPSHOT일 때만 IAM 보안 주체가 ListRules 요청을 하도록 허용합니다. 이렇게 하면 보안 주체가 스냅샷에 대해서만 보존 규칙을 나열할 수 있으며 다른 리소스 유형에 대한 보존 규칙을 나열할 수 없습니다.

JSON

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

```
                "rbin:ListRules"
            ],
            "Resource" : "*",
            "Condition" : {
                "StringEquals" : {
                    "rbin:Request/ResourceType" : "EBS_SNAPSHOT"
                }
            }
        }
    ]
}
```

`rbin:Attribute/ResourceType` 조건 키

이 `rbin:Attribute/ResourceType` 조건 키는 보존 규칙의 `ResourceType` 속성 값을 기준으로 DeleteRule, GetRule, UpdateRule, LockRule, UnlockRule, TagResource, UntagResource 및 ListTagsForResource 요청에 대한 액세스를 필터링하는 데 사용될 수 있습니다.

예제 1 - UpdateRule

다음 샘플 IAM 정책은 요청된 보존 규칙의 `ResourceType` 속성이 `EBS_SNAPSHOT` 또는 `EC2_IMAGE`일 때만 IAM 보안 주체가 UpdateRule 요청을 하도록 허용합니다. 이렇게 하면 보안 주체가 스냅샷 및 AMI에 대해서만 보존 규칙을 업데이트할 수 있습니다.

JSON

```
{
    "Version":"2012-10-17",
    "Statement" : [
        {
            "Effect" : "Allow",
            "Action" :[
                "rbin:UpdateRule"
            ],
            "Resource" : "*",
            "Condition" : {
                "StringEquals" : {
                    "rbin:Attribute/ResourceType" : ["EBS_SNAPSHOT", "EC2_IMAGE"]
                }
            }
        }
```

```
    ]
}
```

예제 2 - DeleteRule

다음 샘플 IAM 정책은 요청된 보존 규칙의 ResourceType 속성이 EBS_SNAPSHOT일 때만 IAM 보안 주체가 DeleteRule 요청을 하도록 허용합니다. 이렇게 하면 보안 주체가 스냅샷에 대해서만 보존 규칙을 삭제할 수 있습니다.

JSON

```
{
    "Version":"2012-10-17",
    "Statement" : [
        {
            "Effect" : "Allow",
            "Action" :[
                "rbin:DeleteRule"
            ],
            "Resource" : "*",
            "Condition" : {
                "StringEquals" : {
                    "rbin:Attribute/ResourceType" : "EBS_SNAPSHOT"
                }
            }
        }
    ]
}
```

휴지통 보존 규칙 생성

보존 규칙을 생성할 때 다음 필수 파라미터를 지정해야 합니다.

- 보호할 리소스 유형(볼륨, 스냅샷 또는 AMIs).
- 보존 규칙의 유형(태그 수준 또는 리전 수준). 태그 수준 규칙은 특정 태그가 있는 리소스만 보호합니다. 리전 수준 규칙은 리전의 모든 리소스를 보호하지만 특정 태그가 있는 리소스를 제외할 수 있습니다.

- 리소스가 삭제된 후 보존할 보존 기간. 이 기간이 만료되면 리소스가 휴지통에서 영구적으로 삭제됩니다. 지원되는 보존 기간은 다음과 같습니다.
  - EBS 볼륨: 1~7일
  - EBS 스냅샷 및 EBS 지원 AMIs: 1~365일

또한 규칙 이름과 설명을 각각 최대 255자까지 지정할 수 있으며, 규칙을 식별하고 구성하는 데 도움이 되는 태그를 지정할 수 있습니다. 이름, 설명 또는 태그에는 개인 식별 정보, 기밀 정보 또는 민감한 정보를 포함하지 않는 것이 좋습니다.

생성 시에 리전 수준 보존 규칙을 잠글 수도 있습니다. 보존 규칙을 생성할 때 잠금을 설정할 경우, 잠금 해제 지연 기간(7일~30일)도 지정해야 합니다. 보존 규칙은 명시적으로 잠그지 않는 한 기본적으로 잠금 해제된 상태로 유지됩니다.

> **Note**
>
> 보존 규칙은 해당 규칙이 생성된 리전에서만 작동합니다. 다른 리전에서 휴지통을 사용하려면 해당 리전에서 추가 보존 규칙을 생성해야 합니다.

다음 방법 중 하나를 사용하여 휴지통 보존 규칙을 생성할 수 있습니다.

Recycle Bin console

태그 수준 보존 규칙을 생성하려면

1. https://console.aws.amazon.com/rbin/home/에서 휴지통 콘솔을 엽니다.
2. 탐색 창에서 Retention rules(보존 규칙)를 선택한 후 Create retention rule(보존 규칙 생성)을 선택합니다.
3. (선택 사항) 보존 규칙 이름(Retention rule name)에 보존 규칙에 대한 설명이 포함된 이름을 입력합니다.
4. (선택 사항) 보존 규칙 설명(Retention rule description)에서 보존 규칙에 대한 간략한 설명을 입력합니다.
5. 리소스 유형(Resource type)에서 보호할 보존 규칙에 대한 리소스 유형을 선택합니다. 보존 규칙은 이 유형의 리소스만 휴지통에 보관합니다.
6. 보존할 리소스 선택에서 특정 태그가 있는 리소스 보존을 선택합니다.
7. 리소스 태그에 휴지통에 보관할 리소스를 식별하는 데 사용할 태그 키 및 값 페어를 입력합니다. 지정된 태그 중 하나 이상이 있는 지정된 유형의 리소스만 보존 규칙에 의해 보존됩니다.

8. 보존 기간에 리소스를 휴지통에 보존할 기간(일)을 입력합니다.
9. 보존 규칙 생성(Create retention rule)을 선택합니다.

리전 수준 보존 규칙을 생성하려면

1. https://console.aws.amazon.com/rbin/home/에서 휴지통 콘솔을 엽니다.
2. 탐색 창에서 Retention rules(보존 규칙)를 선택한 후 Create retention rule(보존 규칙 생성)을 선택합니다.
3. (선택 사항) 보존 규칙 이름(Retention rule name)에 보존 규칙에 대한 설명이 포함된 이름을 입력합니다.
4. (선택 사항) 보존 규칙 설명(Retention rule description)에서 보존 규칙에 대한 간략한 설명을 입력합니다.
5. 리소스 유형(Resource type)에서 보호할 보존 규칙에 대한 리소스 유형을 선택합니다. 보존 규칙은 이 유형의 리소스만 휴지통에 보관합니다.
6. 보존할 리소스 선택에서 모든 리소스 보존을 선택합니다.
7. (선택 사항) 특정 태그가 있는 리소스를 제외하려면 제외 태그에 제외할 리소스를 식별하는 데 사용할 태그 키 및 값 페어를 최대 5개까지 입력합니다. 이러한 태그가 있는 리소스는 보존 규칙에 의해 무시됩니다.
8. 보존 기간에 리소스를 휴지통에 보존할 기간(일)을 입력합니다.
9. (선택 사항) 보존 규칙을 잠그려면 Rule lock settings(규칙 잠금 설정)에서 Lock(잠금)을 선택한 다음 Unlock delay period(잠금 해제 지연 기간)에 잠금 해제 지연 기간을 일 단위로 지정합니다. 잠긴 보존 규칙은 수정하거나 삭제할 수 없습니다. 규칙을 수정하거나 삭제하려면 먼저 규칙을 잠금 해제한 다음, 잠금 해제 지연 기간이 만료될 때까지 기다려야 합니다. 자세한 내용은 휴지통 보존 규칙을 잠가 업데이트 또는 삭제를 방지합니다. 섹션을 참조하세요.

   보존 규칙을 잠금 해제된 상태로 두려면 Rule lock settings(규칙 잠금 설정)에서 Unlock(잠금 해제)를 선택한 상태로 둡니다. 잠금 해제된 보존 규칙은 언제든지 수정하거나 삭제할 수 있습니다.

   > **Note**
   >
   > 제외 태그가 있는 리전 수준 보존 규칙은 잠글 수 없습니다.

10. 보존 규칙 생성(Create retention rule)을 선택합니다.

AWS CLI

보존 규칙 생성

create-route AWS CLI 명령을 사용합니다. --retention-period에 대해 삭제된 스냅샷을 휴지통에 보관할 기간(일)을 지정합니다. 의 경우 볼륨EBS_VOLUME, EBS_SNAPSHOT 스냅샷 또는 AMIsEC2_IMAGE에 대해를 --resource-type지정합니다. 태그 수준 보존 규칙을 생성하려면 에 보존할 리소스를 식별하는 데 사용할 태그를 --resource-tags지정합니다. 리전 수준 보존 규칙을 생성하려면 --resource-tags를 생략하고 선택적으로 --exclude-resource-tags를 지정하여 특정 태그가 있는 리소스를 제외합니다. 리전 수준 보존 규칙을 잠그려면 --lock-configuration을 포함하고 잠금 해제 지연 기간을 일 단위로 지정합니다.

```
aws rbin create-rule \
--retention-period RetentionPeriodValue=number_of_days,RetentionPeriodUnit=DAYS \
--resource-type EBS_VOLUME|EBS_SNAPSHOT|EC2_IMAGE \
--description "rule_description" \
--lock-configuration
 'UnlockDelay={UnlockDelayUnit=DAYS,UnlockDelayValue=unlock_delay_in_days}' \
--resource-tags ResourceTagKey=tag_key,ResourceTagValue=tag_value \
--exclude-resource-tags ResourceTagKey=tag_key,ResourceTagValue=tag_value
```

예제 1.

다음 예제 명령은 삭제된 모든 스냅샷을 7일 동안 보존하는 잠금 해제된 리전 수준 보존 규칙을 생성합니다.

```
aws rbin create-rule \
--retention-period RetentionPeriodValue=7,RetentionPeriodUnit=DAYS \
--resource-type EBS_SNAPSHOT \
--description "Match all snapshots"
```

예제 2.

다음 예제 명령은 purpose=production으로 태그가 지정된 삭제된 스냅샷을 7일 동안 보존하는 태그 수준 규칙을 생성합니다.

```
aws rbin create-rule \
--retention-period RetentionPeriodValue=7,RetentionPeriodUnit=DAYS \
--resource-type EBS_SNAPSHOT \
--description "Match snapshots with a specific tag" \
--resource-tags ResourceTagKey=purpose,ResourceTagValue=production
```

예제 3

다음 예제 명령은 삭제된 모든 스냅샷을 7일 동안 보존하는 잠긴 리전 수준 보존 규칙을 생성합니다. 보존 규칙은 7일의 잠금 해제 지연 기간으로 잠깁니다.

```
aws rbin create-rule \
--retention-period RetentionPeriodValue=7,RetentionPeriodUnit=DAYS \
--resource-type EBS_SNAPSHOT \
--description "Match all snapshots" \
--lock-configuration 'UnlockDelay={UnlockDelayUnit=DAYS,UnlockDelayValue=7}'
```

예 4

다음 예제 명령은 purpose:testing 태그가 지정된 스냅샷을 제외하고 모든 삭제된 스냅샷을 7일 동안 보존하는 잠금 설정된 리전 수준 보존 규칙을 생성합니다.

```
aws rbin create-rule \
--retention-period RetentionPeriodValue=7,RetentionPeriodUnit=DAYS \
--resource-type EBS_SNAPSHOT \
--description "Match only production snapshots" \
--exclude-resource-tags ResourceTagKey=purpose,ResourceTagValue=testing
```

기존 휴지통 보존 규칙 업데이트

잠금 해제된 보존 규칙의 설명, 리소스 태그 및 보존 기간은 생성 후 언제든지 업데이트할 수 있습니다. 보존 규칙이 잠금 해제된 경우에도 보존 규칙의 리소스 유형이나 잠금 해제 지연 기간은 업데이트할 수 없습니다.

잠긴 보존 규칙은 어떤 식으로도 업데이트할 수 없습니다. 잠긴 보존 규칙을 수정해야 하는 경우 먼저 잠금을 해제하고 잠금 해제 지연 기간이 만료될 때까지 기다려야 합니다.

잠긴 보존 규칙의 잠금 해제 지연 기간을 수정해야 하는 경우 보존 규칙을 잠금 해제하고 현재 잠금 해제 지연 기간이 만료될 때까지 기다려야 합니다. 잠금 해제 지연 기간이 만료되면 보존 규칙을 다시 잠그고 새 잠금 해제 지연 기간을 지정해야 합니다.

> **Note**
>
> 보존 규칙 설명에 개인 식별 정보, 기밀 정보 또는 민감한 정보를 포함하지 않는 것이 좋습니다.

보존 규칙을 업데이트한 후에는 보존되는 새 리소스에만 변경 사항이 적용됩니다. 변경 사항은 이전에 휴지통으로 이동된 리소스에는 영향을 주지 않습니다. 예를 들어 보존 규칙의 보존 기간을 업데이트하면 그 후에 삭제되는 스냅샷만 새 보존 기간 동안 보존됩니다. 업데이트 전에 휴지통으로 이동된 스냅샷은 이전 보존 기간 동안 계속 보존됩니다.

다음 방법 중 하나를 사용하여 보존 규칙을 업데이트할 수 있습니다.

Recycle Bin console

보존 규칙 업데이트

1. https://console.aws.amazon.com/rbin/home/에서 휴지통 콘솔을 엽니다.
2. 탐색 창에서 Retention rules(보존 규칙)를 선택합니다.
3. 그리드에서 업데이트할 보존 규칙을 선택하고 작업(Actions), 보존 규칙 편집(Edit retention rule)을 선택합니다.
4. 규칙 세부 정보(Rule details) 섹션에서 필요에 따라 보존 규칙 이름(Retention rule name)과 보존 규칙 설명(Retention rule description)을 업데이트합니다.
5. 규칙 설정 섹션에서 필요에 따라 리소스 유형, 일치시킬 리소스 태그 및 보존 기간을 업데이트합니다.
6. 태그(Tags) 섹션에서 필요에 따라 보존 규칙 태그를 추가하거나 제거합니다.
7. 보존 규칙 저장(Save retention rule)을 선택합니다.

AWS CLI

보존 규칙 업데이트

update-rule AWS CLI 명령을 사용합니다. 의 경우 업데이트할 보존 규칙의 ID를 `--identifier`지정합니다.의 경우 `EBS_VOLUME` 볼륨`--resource-types`, `EBS_SNAPSHOT` 스냅샷 또는 AMIs`EC2_IMAGE`에 대해를 지정합니다.

```
aws rbin update-rule \
--identifier rule_ID \
--retention-period RetentionPeriodValue=number_of_days,RetentionPeriodUnit=DAYS \
--resource-type EBS_VOLUME|EBS_SNAPSHOT|EC2_IMAGE \
--description "rule_description"
```

예제

다음 예제 명령은 보존 규칙 6lsJ2Fa9nh9를 업데이트하여 모든 스냅샷을 7일 동안 보존하고 관련 설명을 업데이트합니다.

```
aws rbin update-rule \
--identifier 6lsJ2Fa9nh9 \
--retention-period RetentionPeriodValue=7,RetentionPeriodUnit=DAYS \
--resource-type EBS_SNAPSHOT \
--description "Retain for three weeks"
```

휴지통 보존 규칙을 잠가 업데이트 또는 삭제를 방지합니다.

휴지통을 사용하면 언제든지 리전 수준의 보존 규칙을 잠글 수 있습니다.

잠긴 보존 규칙은 필요한 IAM 권한이 있는 사용자라도 수정하거나 삭제할 수 없습니다. 보존 규칙을 잠그면 실수로 인한 또는 악의적인 수정과 삭제를 방지할 수 있습니다.

보존 규칙을 잠글 때 잠금 해제 지연 기간을 지정해야 합니다. 이 기간은 보존 규칙을 잠금 해제한 후 수정 또는 삭제할 수 있게 되기까지 기다려야 하는 기간입니다. 잠금 해제 지연 기간 동안에는 보존 규칙을 수정하거나 삭제할 수 없습니다. 잠금 해제 지연 기간이 만료된 후에만 보존 규칙을 수정하거나 삭제할 수 있습니다.

보존 규칙을 잠근 후에는 잠금 해제 지연 기간을 변경할 수 없습니다. 계정 권한이 침해된 경우 잠금 해제 지연 기간을 통해 보안 위협을 탐지하고 대응할 추가 시간을 확보할 수 있습니다. 이 기간은 보안 침해를 식별하고 대응하는 데 걸리는 시간보다 길어야 합니다. 적절한 기간을 설정하려면 이전 보안 인시던트와 계정 침해 인시던트를 식별하고 해결하는 데 필요한 시간을 검토하면 됩니다.

Amazon EventBridge 규칙을 사용하여 보존 규칙 잠금 상태 변경을 알리는 것이 좋습니다. 자세한 내용은 Amazon EventBridge를 사용하여 휴지통 모니터링 단원을 참조하십시오.

고려 사항

- 태그 수준 보존 규칙 또는 제외 태그가 있는 리전 수준 보존 규칙은 잠글 수 없습니다.
- 잠금이 해제된 보존 규칙은 언제든지 잠글 수 있습니다.
- 잠금 해제 지연 기간은 7~30일이어야 합니다.
- 잠금 해제 지연 기간 동안 보존 규칙을 다시 잠글 수 있습니다. 보존 규칙을 다시 잠그면 잠금 해제 지연 기간이 재설정됩니다.

다음 방법 중 하나를 사용하여 리전 수준 보존 규칙을 잠글 수 있습니다.

Recycle Bin console

보존 규칙을 잠그려면

1. https://console.aws.amazon.com/rbin/home/에서 휴지통 콘솔을 엽니다.
2. 탐색 패널에서 보존 규칙(Retention rules)을 선택합니다.
3. 그리드에서 잠글 잠금 해제된 보존 규칙을 선택하고 Actions(작업), Edit retention rule lock(보존 규칙 잠금 편집)을 선택합니다.
4. 보존 규칙 잠금 편집 화면에서 Lock(잠금)을 선택한 다음 Unlock delay period(잠금 해제 지연 기간)에 잠금 해제 지연 기간을 일 단위로 지정합니다.
5. I acknowledge that locking the retention rule will prevent it from being modified or deleted(보존 규칙을 잠그면 수정 또는 삭제할 수 없음을 알고 있습니다) 확인란을 선택한 다음 Save(저장)를 선택합니다.

AWS CLI

잠금 해제된 보존 규칙을 잠그려면

lock-rule AWS CLI 명령을 사용합니다. `--identifier`에 잠글 보존 규칙의 ID를 지정합니다. `--lock-configuration`에 잠금 해제 지연 기간을 일 단위로 지정합니다.

```
aws rbin lock-rule \
--identifier rule_ID \
--lock-configuration
 'UnlockDelay={UnlockDelayUnit=DAYS,UnlockDelayValue=number_of_days}'
```

예제

다음 예제 명령은 보존 규칙 `6lsJ2Fa9nh9`를 잠그고 잠금 해제 지연 기간을 15일로 설정합니다.

```
aws rbin lock-rule \
--identifier 6lsJ2Fa9nh9 \
--lock-configuration 'UnlockDelay={UnlockDelayUnit=DAYS,UnlockDelayValue=15}'
```

휴지통 보존 규칙을 업데이트 또는 삭제할 수 있도록 잠금 해제

잠긴 보존 규칙은 수정하거나 삭제할 수 없습니다. 잠긴 보존 규칙을 수정해야 하는 경우 먼저 잠금을 해제해야 합니다. 보존 규칙을 잠금 해제한 후에는 잠금 해제 지연 기간이 만료될 때까지 기다렸다가

수정 또는 삭제할 수 있습니다. 잠금 해제 지연 기간 중에는 보존 규칙을 수정하거나 삭제할 수 없습니다.

잠금 해제된 보존 규칙은 필요한 IAM 권한이 있는 사용자가 언제든지 수정 및 삭제할 수 있습니다. 보존 규칙을 잠금 해제하면 실수로 인한 또는 악의적인 수정 및 삭제에 노출될 수 있습니다.

고려 사항

- 잠금 해제 지연 기간 동안 보존 규칙을 다시 잠글 수 있습니다.
- 잠금 해제 지연 기간이 만료된 후에는 보존 규칙을 다시 잠글 수 있습니다.
- 잠금 해제 지연 기간은 우회할 수 없습니다.
- 초기에 잠근 후에는 잠금 해제 지연 기간을 변경할 수 없습니다.

Amazon EventBridge 규칙을 사용하여 보존 규칙 잠금 상태 변경을 알리는 것이 좋습니다. 자세한 내용은 Amazon EventBridge를 사용하여 휴지통 모니터링 단원을 참조하십시오.

다음 방법 중 하나를 사용하여 잠긴 리전 수준 보존 규칙을 잠금 해제할 수 있습니다.

Recycle Bin console

보존 규칙을 잠금 해제하려면

1. https://console.aws.amazon.com/rbin/home/에서 휴지통 콘솔을 엽니다.
2. 탐색 패널에서 보존 규칙(Retention rules)을 선택합니다.
3. 그리드에서 잠금 해제할 잠긴 보존 규칙을 선택하고 Actions(작업), Edit retention rule lock(보존 규칙 잠금 편집)을 선택합니다.
4. 보존 규칙 잠금 편집 화면에서 Unlock(잠금 해제)를 선택한 다음 Save(저장)을 선택합니다.

AWS CLI

잠긴 보존 규칙을 잠금 해제하려면

unlock-rule AWS CLI 명령을 사용합니다. `--identifier`에 잠금 해제할 보존 규칙의 ID를 지정합니다.

```
aws rbin unlock-rule \
--identifier rule_ID
```

예제

다음 예제 명령은 보존 규칙 6lsJ2Fa9nh9를 잠금 해제합니다.

```
aws rbin unlock-rule \
--identifier 6lsJ2Fa9nh9
```

휴지통 보존 규칙에 태그 지정

보관 규칙에 사용자 정의 태그를 할당하여 용도, 소유자 또는 환경과 같은 다양한 방식으로 분류할 수 있습니다. 그러면 할당한 사용자 정의 태그를 기반으로 특정 보존 규칙을 효율적으로 찾을 수 있습니다.

다음 방법 중 하나로 보존 규칙에 태그를 할당할 수 있습니다.

Recycle Bin console

보존 규칙에 태그 지정

1. https://console.aws.amazon.com/rbin/home/에서 휴지통 콘솔을 엽니다.
2. 탐색 창에서 Retention rules(보존 규칙)를 선택합니다.
3. 태그를 지정할 보존 규칙을 선택하고 태그(Tags) 탭을 선택한 다음 태그 관리(Manage tags)를 선택합니다.
4. 태그 추가를 선택합니다. 키(Key)에 태그 키를 입력합니다. 값(Value)에 태그 값을 입력합니다.
5. 저장(Save)을 선택합니다.

AWS CLI

보존 규칙에 태그 지정

tag-resource AWS CLI 명령을 사용합니다. `--resource-arn`에 대해 태그를 지정할 보존 규칙의 Amazon 리소스 이름(ARN)을 지정하고, `--tags`에 대해 태그 키 및 값 페어를 지정합니다.

```
aws rbin tag-resource \
--resource-arn retention_rule_arn \
--tags key=tag_key,value=tag_value
```

예제

다음 예제 명령은 보관 규칙 arn:aws:rbin:us-east-1:123456789012:rule/nOoSBBtItF3에 태그 purpose=production을 지정합니다.

```
aws rbin tag-resource \
--resource-arn arn:aws:rbin:us-east-1:123456789012:rule/nOoSBBtItF3 \
--tags key=purpose,value=production
```

보존 규칙 태그 보기

다음 방법 중 하나를 사용하여 보존 규칙에 할당된 태그를 볼 수 있습니다.

Recycle Bin console

보존 규칙에 대한 태그 보기

1. https://console.aws.amazon.com/rbin/home/에서 휴지통 콘솔을 엽니다.
2. 탐색 창에서 Retention rules(보존 규칙)를 선택합니다.
3. 태그를 볼 보존 규칙을 선택하고 태그(Tags) 탭을 선택합니다.

AWS CLI

보존 규칙에 할당된 태그 보기

list-tags-for-resource AWS CLI 명령을 사용합니다. --resource-arn에 대해 보존 규칙의 ARN을 지정합니다.

```
aws rbin list-tags-for-resource \
--resource-arn retention_rule_arn
```

예제

다음 예제 명령은 보존 규칙 arn:aws:rbin:us-east-1:123456789012:rule/nOoSBBtItF3에 대한 태그를 나열합니다.

```
aws rbin list-tags-for-resource \
--resource-arn arn:aws:rbin:us-east-1:123456789012:rule/nOoSBBtItF3
```

보존 규칙에서 태그 제거

다음 방법 중 하나를 사용하여 보존 규칙에서 태그를 제거할 수 있습니다.

Recycle Bin console

보존 규칙에서 태그 제거

1. https://console.aws.amazon.com/rbin/home/에서 휴지통 콘솔을 엽니다.
2. 탐색 창에서 Retention rules(보존 규칙)를 선택합니다.
3. 태그를 제거할 보존 규칙을 선택하고 태그(Tags) 탭을 선택한 다음 태그 관리(Manage tags)를 선택합니다.
4. 제거할 태그 옆에 있는 제거(Remove)를 선택합니다.
5. 저장(Save)을 선택합니다.

AWS CLI

보존 규칙에서 태그 제거

untag-resource AWS CLI 명령을 사용합니다. `--resource-arn`에 대해 보존 규칙의 ARN을 지정합니다. `--tagkeys`에 대해 제거할 태그의 태그 키를 지정합니다.

```
aws rbin untag-resource \
--resource-arn retention_rule_arn \
--tagkeys tag_key
```

예제

다음 예제 명령은 보관 규칙 `arn:aws:rbin:us-east-1:123456789012:rule/nOoSBBtItF3`에서 태그 키가 `purpose`인 태그를 제거합니다.

```
aws rbin untag-resource \
--resource-arn arn:aws:rbin:us-east-1:123456789012:rule/nOoSBBtItF3 \
--tagkeys purpose
```

리소스를 보존하지 않도록 휴지통 보존 규칙 삭제

언제든지 보존 규칙을 삭제할 수 있습니다. 보존 규칙을 삭제하면 새 리소스가 삭제된 후 더 이상 휴지통에 유지되지 않습니다. 보존 규칙이 삭제되기 전에 휴지통으로 이동된 리소스는 보존 규칙에 정의된 보존 기간에 따라 휴지통에 계속 보관됩니다. 기간이 만료되면 리소스가 휴지통에서 영구적으로 삭제됩니다.

다음 방법 중 하나를 사용하여 보존 규칙을 삭제할 수 있습니다.

Recycle Bin console

보존 규칙 삭제

1. https://console.aws.amazon.com/rbin/home/에서 휴지통 콘솔을 엽니다.
2. 탐색 창에서 Retention rules(보존 규칙)를 선택합니다.
3. 그리드에서 삭제할 보존 규칙을 선택하고 작업(Actions), 보존 규칙 삭제(Delete retention rule)를 선택합니다.
4. 메시지가 표시되면 확인 메시지를 입력하고 보존 규칙 삭제(Delete retention rule)를 선택합니다.

AWS CLI

보존 규칙 삭제

delete-rule AWS CLI 명령을 사용합니다. `--identifier`에 대해 삭제할 보존 규칙의 ID를 지정합니다.

```
aws rbin delete-rule --identifier rule_ID
```

예제

다음 예제 명령은 보존 규칙 `6lsJ2Fa9nh9`를 삭제합니다.

```
aws rbin delete-rule --identifier 6lsJ2Fa9nh9
```

휴지통에서 삭제된 스냅샷 복구

이 주제에서는 휴지통에서 Amazon EBS 스냅샷을 복구하는 방법을 설명합니다.

주제

- 휴지통의 스냅샷 작업을 위한 권한
- 휴지통의 스냅샷 보기
- 휴지통에서 스냅샷 복원

휴지통의 스냅샷 작업을 위한 권한

기본적으로 사용자는 휴지통에 있는 스냅샷으로 작업할 권한이 없습니다. 사용자가 이러한 리소스로 작업하도록 허용하려면 특정 리소스 및 API 작업을 사용할 권한을 부여하는 IAM 정책을 생성해야 합니다. 정책이 생성된 후에는 사용자, 그룹 또는 역할에 권한을 추가해야 합니다.

휴지통에 있는 스냅샷을 보고 복구하려면 사용자에게 다음과 같은 권한이 있어야 합니다.

- `ec2:ListSnapshotsInRecycleBin`
- `ec2:RestoreSnapshotFromRecycleBin`

휴지통의 스냅샷에 대한 태그를 관리하려면 사용자에게 다음과 같은 추가 권한이 필요합니다.

- `ec2:CreateTags`
- `ec2:DeleteTags`

휴지통 콘솔을 사용하려면 사용자에게 `ec2:DescribeTags` 권한이 필요합니다.

다음은 예시 IAM 정책입니다. 여기에는 콘솔 사용자에 대한 `ec2:DescribeTags` 권한이 포함되며 태그 관리를 위한 `ec2:CreateTags` 및 `ec2:DeleteTags` 권한이 포함됩니다. 권한이 필요하지 않은 경우 정책에서 권한을 제거할 수 있습니다.

액세스 권한을 제공하려면 사용자, 그룹 또는 역할에 권한을 추가하세요:

- 의 사용자 및 그룹 AWS IAM Identity Center:

  권한 세트를 생성합니다. AWS IAM Identity Center 사용자 안내서에서 권한 세트 생성의 지침을 따릅니다.
- ID 제공업체를 통해 IAM에서 관리되는 사용자:

  ID 페더레이션을 위한 역할을 생성합니다. IAM 사용자 설명서의 Create a role for a third-party identity provider (federation)의 지침을 따릅니다.

- IAM 사용자:
  - 사용자가 맡을 수 있는 역할을 생성합니다. IAM 사용자 설명서에서 Create a role for an IAM user의 지침을 따릅니다.
  - (권장되지 않음) 정책을 사용자에게 직접 연결하거나 사용자를 사용자 그룹에 추가합니다. IAM 사용 설명서에서 사용자(콘솔)에 권한 추가의 지침을 따릅니다.

휴지통을 사용하는 데 필요한 권한에 대한 자세한 내용은 휴지통 및 보존 규칙 작업을 위한 권한 섹션을 참조하세요.

휴지통의 스냅샷 보기

스냅샷이 휴지통에 있는 동안 다음과 같은 제한된 정보를 볼 수 있습니다.

- 스냅샷의 ID입니다.
- 스냅샷 설명입니다.
- 스냅샷이 생성된 볼륨의 ID입니다.
- 스냅샷이 삭제되고 휴지통에 들어간 날짜 및 시간입니다.
- 보존 기간이 만료되는 날짜 및 시간입니다. 이때 스냅샷은 휴지통에서 영구적으로 삭제됩니다.

다음 방법 중 하나를 사용하여 휴지통의 스냅샷을 볼 수 있습니다.

Recycle Bin console

콘솔을 사용하여 휴지통의 스냅샷 보기

1. https://console.aws.amazon.com/rbin/home/에서 휴지통 콘솔을 엽니다.
2. 탐색 창에서 Recycle Bin(휴지통)을 선택합니다.
3. 그리드에는 현재 휴지통에 있는 모든 스냅샷이 나열됩니다. 특정 스냅샷에 대한 세부 정보를 확인하려면 그리드에서 해당 스냅샷을 선택한 다음 작업(Actions), 세부 정보 보기(View details)를 선택합니다.

AWS CLI

를 사용하여 휴지통에서 스냅샷을 보려면 AWS CLI

list-snapshots-in-recycle-bin AWS CLI 명령을 사용합니다. 특정 스냅샷을 보려면 `--snapshot-id` 옵션을 포함합니다. 또는 휴지통의 모든 스냅샷을 보려면 `--snapshot-id` 옵션을 생략합니다.

```
aws ec2 list-snapshots-in-recycle-bin --snapshot-id snapshot_id
```

예를 들어 다음 명령은 휴지통에 있는 스냅샷 snap-01234567890abcdef에 대한 정보를 반환합니다.

```
aws ec2 list-snapshots-in-recycle-bin --snapshot-id snap-01234567890abcdef
```

출력 예시:

```
{
    "SnapshotRecycleBinInfo": [
        {
            "Description": "Monthly data backup snapshot",
            "RecycleBinEnterTime": "2021-12-01T13:00:00.000Z",
            "RecycleBinExitTime": "2021-12-15T13:00:00.000Z",
            "VolumeId": "vol-abcdef09876543210",
            "SnapshotId": "snap-01234567890abcdef"
        }
    ]
}
```

휴지통에서 스냅샷 복원

스냅샷이 휴지통에 있는 동안에는 어떤 식으로도 사용할 수 없습니다. 스냅샷을 사용하려면 먼저 복원해야 합니다. 휴지통에서 스냅샷을 복원하면 스냅샷을 즉시 사용할 수 있으며 휴지통에서 스냅샷이 제거됩니다. 계정의 다른 스냅샷을 사용하는 것과 동일한 방식으로 복원된 스냅샷을 사용할 수 있습니다.

다음 방법 중 하나를 사용하여 휴지통에서 스냅샷을 복원할 수 있습니다.

Recycle Bin console

콘솔을 사용하여 휴지통에서 스냅샷 복원

1. https://console.aws.amazon.com/rbin/home/에서 휴지통 콘솔을 엽니다.
2. 탐색 창에서 Recycle Bin(휴지통)을 선택합니다.
3. 그리드에는 현재 휴지통에 있는 모든 스냅샷이 나열됩니다. 복원할 스냅샷을 선택한 다음 복구(Recover)를 선택합니다.
4. 메시지가 나타나면 복구(Recover)를 선택합니다.

AWS CLI

를 사용하여 휴지통에서 삭제된 스냅샷을 복원하려면 AWS CLI

restore-snapshot-from-recycle-bin AWS CLI 명령을 사용합니다. `--snapshot-id`에 대해 복원할 스냅샷의 ID를 지정합니다.

```
aws ec2 restore-snapshot-from-recycle-bin --snapshot-id snapshot_id
```

예를 들어 다음 명령은 휴지통에서 스냅샷 `snap-01234567890abcdef`를 복원합니다.

```
aws ec2 restore-snapshot-from-recycle-bin --snapshot-id snap-01234567890abcdef
```

출력 예시:

```
{
    "SnapshotId": "snap-01234567890abcdef",
    "Description": "Monthly data backup snapshot",
    "Encrypted": false,
    "OwnerId": "111122223333",
    "Progress": "100%",
    "StartTime": "2021-12-01T13:00:00.000000+00:00",
    "State": "recovering",
    "VolumeId": "vol-ffffffff",
    "VolumeSize": 30
}
```

휴지통에서 삭제된 볼륨 복구

이 주제에서는 휴지통에서 Amazon EBS 볼륨을 복구하는 방법을 설명합니다.

주제

- 휴지통의 볼륨 작업에 대한 권한
- 휴지통에서 볼륨 보기
- 휴지통에서 볼륨 복원

휴지통의 볼륨 작업에 대한 권한

기본적으로 사용자는 휴지통에 있는 볼륨으로 작업할 수 있는 권한이 없습니다. 사용자가 이러한 리소스로 작업하도록 허용하려면 특정 리소스 및 API 작업을 사용할 권한을 부여하는 IAM 정책을 생성해야 합니다. 정책이 생성된 후에는 사용자, 그룹 또는 역할에 권한을 추가해야 합니다.

휴지통에 있는 볼륨을 보고 복구하려면 사용자에게 다음 권한이 있어야 합니다.

- `ec2:ListVolumesInRecycleBin`
- `ec2:RestoreVolumeFromRecycleBin`

휴지통의 볼륨에 대한 태그를 관리하려면 사용자에게 다음과 같은 추가 권한이 필요합니다.

- `ec2:CreateTags`
- `ec2:DeleteTags`

휴지통 콘솔을 사용하려면 사용자에게 `ec2:DescribeTags` 권한이 필요합니다.

다음은 예시 IAM 정책입니다. 여기에는 콘솔 사용자에 대한 `ec2:DescribeTags` 권한이 포함되며 태그 관리를 위한 `ec2:CreateTags` 및 `ec2:DeleteTags` 권한이 포함됩니다. 권한이 필요하지 않은 경우 정책에서 권한을 제거할 수 있습니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowRecycleBinVolumeOperations",
      "Effect": "Allow",
      "Action": [
        "ec2:ListVolumesInRecycleBin",
        "ec2:RestoreVolumeFromRecycleBin"
      ],
      "Resource": "arn:aws:ec2:*:123456789012:volume/*"
    },
    {
      "Sid": "AllowVolumeTagOperations",
      "Effect": "Allow",
      "Action": [
        "ec2:CreateTags",
        "ec2:DeleteTags",
```

```
        "ec2:DescribeTags"
      ],
      "Resource": "arn:aws:ec2:*:123456789012:volume/*"
    }
  ]
}
```

액세스 권한을 제공하려면 사용자, 그룹 또는 역할에 권한을 추가하세요:

- 의 사용자 및 그룹 AWS IAM Identity Center:

  권한 세트를 생성합니다. AWS IAM Identity Center 사용자 안내서에서 권한 세트 생성의 지침을 따릅니다.
- ID 제공업체를 통해 IAM에서 관리되는 사용자:

  ID 페더레이션을 위한 역할을 생성합니다. IAM 사용자 설명서의 Create a role for a third-party identity provider (federation)의 지침을 따릅니다.
- IAM 사용자:
  - 사용자가 맡을 수 있는 역할을 생성합니다. IAM 사용자 설명서에서 Create a role for an IAM user의 지침을 따릅니다.
  - (권장되지 않음) 정책을 사용자에게 직접 연결하거나 사용자를 사용자 그룹에 추가합니다. IAM 사용 설명서에서 사용자(콘솔)에 권한 추가의 지침을 따릅니다.

휴지통을 사용하는 데 필요한 권한에 대한 자세한 내용은 휴지통 및 보존 규칙 작업을 위한 권한 섹션을 참조하세요.

휴지통에서 볼륨 보기

볼륨이 휴지통에 있는 동안 다음을 포함하여 볼륨에 대한 제한된 정보를 볼 수 있습니다.

- 볼륨의 ID입니다.
- 볼륨의 크기입니다.
- 볼륨 유형입니다.
- 볼륨이 삭제되고 휴지통에 들어간 날짜와 시간입니다.
- 보존 기간이 만료되는 날짜 및 시간입니다. 현재 볼륨은 휴지통에서 영구적으로 삭제됩니다.

다음 방법 중 하나를 사용하여 휴지통에서 볼륨을 볼 수 있습니다.

Recycle Bin console

콘솔을 사용하여 휴지통에서 볼륨을 보려면

1. https://console.aws.amazon.com/rbin/home/에서 휴지통 콘솔을 엽니다.
2. 탐색 창에서 Recycle Bin(휴지통)을 선택합니다.
3. 그리드에는 현재 휴지통에 있는 모든 볼륨이 나열됩니다. 특정 볼륨의 세부 정보를 보려면 그리드에서 해당 볼륨을 선택하고 작업, 세부 정보 보기를 선택합니다.

AWS CLI

를 사용하여 휴지통에서 볼륨을 보려면 AWS CLI

list-volumes-in-recycle-bin AWS CLI 명령을 사용합니다. 특정 볼륨을 보는 `--volume-id` 옵션을 포함합니다. 또는 휴지통의 모든 볼륨을 보는 `--volume-id` 옵션을 생략합니다.

```
aws ec2 list-volumes-in-recycle-bin --volume-id volume_id
```

예를 들어 다음 명령은 휴지통의 볼륨`vol-01234567890abcdef`에 대한 정보를 제공합니다.

```
aws ec2 list-volumes-in-recycle-bin --volume-id vol-01234567890abcdef
```

출력 예시:

```
{
    "VolumeRecycleBinInfo": [
        {
            "VolumeId": "vol-01234567890abcdef",
            "RecycleBinEnterTime": "2021-12-01T13:00:00.000Z",
            "RecycleBinExitTime": "2021-12-08T13:00:00.000Z"
        }
    ]
}
```

휴지통에서 볼륨 복원

볼륨이 휴지통에 있는 동안에는 어떤 식으로든 볼륨을 사용할 수 없습니다. 볼륨을 사용하려면 먼저 볼륨을 복원해야 합니다. 휴지통에서 볼륨을 복원하면 볼륨을 즉시 사용할 수 있으며 휴지통에서 볼륨이 제거됩니다. 계정의 다른 볼륨을 사용하는 것과 동일한 방식으로 복원된 볼륨을 사용할 수 있습니다.

다음 방법 중 하나를 사용하여 휴지통에서 볼륨을 복원할 수 있습니다.

Recycle Bin console

콘솔을 사용하여 휴지통에서 볼륨을 복원하려면

1. https://console.aws.amazon.com/rbin/home/에서 휴지통 콘솔을 엽니다.
2. 탐색 창에서 Recycle Bin(휴지통)을 선택합니다.
3. 그리드에는 현재 휴지통에 있는 모든 볼륨이 나열됩니다. 복원할 볼륨을 선택하고 복구를 선택합니다.
4. 메시지가 나타나면 복구(Recover)를 선택합니다.

AWS CLI

를 사용하여 휴지통에서 삭제된 볼륨을 복원하려면 AWS CLI

restore-volume-from-recycle-bin AWS CLI 명령을 사용합니다. 에서 복원할 볼륨의 ID를 `--volume-id`지정합니다.

```
aws ec2 restore-volume-from-recycle-bin --volume-id volume_id
```

예를 들어 다음 명령은 휴지통`vol-01234567890abcdef`에서 볼륨을 복원합니다.

```
aws ec2 restore-volume-from-recycle-bin --volume-id vol-01234567890abcdef
```

출력 예시:

```
{
    "VolumeId": "vol-01234567890abcdef",
    "State": "available",
    "Size": 100,
    "VolumeType": "gp3",
    "AvailabilityZone": "us-east-1a",
    "CreateTime": "2021-12-01T13:00:00.000000+00:00",
    "Encrypted": false
}
```

삭제된 AMI를 휴지통에서 복구

이 주제에서는 휴지통에서 Amazon EBS 지원 AMIs를 복구하는 방법을 설명합니다.

주제

- 휴지통의 AMI 작업을 위한 권한
- 휴지통의 AMI 보기
- 휴지통에서 AMI 복원

휴지통의 AMI 작업을 위한 권한

기본적으로 사용자는 휴지통에 있는 AMI로 작업할 권한이 없습니다. 사용자가 이러한 리소스로 작업하도록 허용하려면 특정 리소스 및 API 작업을 사용할 권한을 부여하는 IAM 정책을 생성해야 합니다. 정책이 생성된 후에는 사용자, 그룹 또는 역할에 권한을 추가해야 합니다.

휴지통에 있는 AMI를 보고 복구하려면 사용자에게 다음과 같은 권한이 있어야 합니다.

- `ec2:ListImagesInRecycleBin`
- `ec2:RestoreImageFromRecycleBin`

휴지통의 AMI에 대한 태그를 관리하려면 사용자에게 다음과 같은 추가 권한이 필요합니다.

- `ec2:CreateTags`
- `ec2:DeleteTags`

휴지통 콘솔을 사용하려면 사용자에게 `ec2:DescribeTags` 권한이 필요합니다.

다음은 예시 IAM 정책입니다. 여기에는 콘솔 사용자에 대한 `ec2:DescribeTags` 권한이 포함되며 태그 관리를 위한 `ec2:CreateTags` 및 `ec2:DeleteTags` 권한이 포함됩니다. 권한이 필요하지 않은 경우 정책에서 권한을 제거할 수 있습니다.

액세스 권한을 제공하려면 사용자, 그룹 또는 역할에 권한을 추가하세요:

- 의 사용자 및 그룹 AWS IAM Identity Center:

  권한 세트를 생성합니다. AWS IAM Identity Center 사용자 안내서에서 권한 세트 생성의 지침을 따릅니다.

- ID 제공업체를 통해 IAM에서 관리되는 사용자:

  ID 페더레이션을 위한 역할을 생성합니다. IAM 사용자 설명서의 Create a role for a third-party identity provider (federation)의 지침을 따릅니다.

- IAM 사용자:
  - 사용자가 맡을 수 있는 역할을 생성합니다. IAM 사용자 설명서에서 Create a role for an IAM user의 지침을 따릅니다.
  - (권장되지 않음) 정책을 사용자에게 직접 연결하거나 사용자를 사용자 그룹에 추가합니다. IAM 사용 설명서에서 사용자(콘솔)에 권한 추가의 지침을 따릅니다.

휴지통을 사용하는 데 필요한 권한에 대한 자세한 내용은 휴지통 및 보존 규칙 작업을 위한 권한 섹션을 참조하세요.

휴지통의 AMI 보기

AMI가 휴지통에 있는 동안 다음과 같은 제한된 AMI 정보를 볼 수 있습니다.

- AMI의 이름, 설명 및 고유 ID
- AMI가 삭제되고 휴지통에 들어간 날짜 및 시간
- 보존 기간이 만료되는 날짜 및 시간입니다. 이때 AMI는 영구적으로 삭제됩니다.

다음 방법 중 하나를 사용하여 휴지통의 AMI를 볼 수 있습니다.

Recycle Bin console

콘솔을 사용하여 휴지통의 AMI 보기

1. console.aws.amazon.com/rbin/home/에서 휴지통 콘솔을 엽니다.
2. 탐색 창에서 Recycle Bin(휴지통)을 선택합니다.
3. 그리드에는 현재 휴지통에 있는 모든 리소스가 나열됩니다. 특정 AMI에 대한 세부 정보를 확인하려면 그리드에서 해당 AMI를 선택한 다음 작업(Actions), 세부 정보 보기(View details)를 선택합니다.

AWS CLI

를 사용하여 휴지통에서 삭제된 AMIs를 보려면 AWS CLI

list-images-in-recycle-bin AWS CLI 명령을 사용합니다. 특정 AMI를 보려면 `--image-id` 옵션을 포함하고 보려는 AMI의 ID를 지정합니다. 단일 요청에 최대 20개의 ID를 지정할 수 있습니다.

휴지통의 모든 AMI를 보려면 `--image-id` 옵션을 생략합니다. `--max-items`에 대한 값을 지정하지 않으면 명령은 기본적으로 페이지당 1,000개의 항목을 반환합니다. 자세한 내용은 Amazon EC2 API Reference(Amazon EC2 API 레퍼런스)의 Pagination(페이지네이션)을 참조하세요.

```
aws ec2 list-images-in-recycle-bin --image-id ami_id
```

예를 들어 다음 명령은 휴지통에 있는 AMI `ami-01234567890abcdef`에 대한 정보를 반환합니다.

```
aws ec2 list-images-in-recycle-bin --image-id ami-01234567890abcdef
```

출력 예시:

```
{
    "Images": [
        {
            "ImageId": "ami-0f740206c743d75df",
            "Name": "My AL2 AMI",
            "Description": "My Amazon Linux 2 AMI",
            "RecycleBinEnterTime": "2021-11-26T21:04:50+00:00",
            "RecycleBinExitTime": "2022-03-06T21:04:50+00:00"
        }
    ]
}
```

Important

다음 오류가 발생하면 AWS CLI 버전을 업데이트해야 할 수 있습니다. 자세한 내용은 명령을 찾을 수 없음 오류를 참조하세요.

```
aws.exe: error: argument operation: Invalid choice, valid choices are: ...
```

휴지통에서 AMI 복원

AMI가 휴지통에 있는 동안에는 AMI를 사용할 수 없습니다. AMI를 사용하려면 먼저 복원해야 합니다. 휴지통에서 AMI를 복원하면 AMI를 즉시 사용할 수 있으며 휴지통에서 AMI가 제거됩니다. 계정의 다른 AMI를 사용하는 것과 동일한 방식으로 복원된 AMI를 사용할 수 있습니다.

다음 방법 중 하나를 사용하여 휴지통에서 AMI를 복원할 수 있습니다.

Recycle Bin console

콘솔을 사용하여 휴지통에서 AMI 복원

1. console.aws.amazon.com/rbin/home/에서 휴지통 콘솔을 엽니다.
2. 탐색 창에서 Recycle Bin(휴지통)을 선택합니다.
3. 그리드에는 현재 휴지통에 있는 모든 리소스가 나열됩니다. 복원할 AMI를 선택하고 복구(Recover)를 선택합니다.
4. 메시지가 나타나면 복구(Recover)를 선택합니다.

AWS CLI

를 사용하여 휴지통에서 삭제된 AMI를 복원하려면 AWS CLI

restore-image-from-recycle-bin AWS CLI 명령을 사용합니다. `--image-id`에 대해 복원할 AMI의 ID를 지정합니다.

```
aws ec2 restore-image-from-recycle-bin --image-id ami_id
```

예를 들어 다음 명령은 휴지통에서 AMI `ami-01234567890abcdef`를 복원합니다.

```
aws ec2 restore-image-from-recycle-bin --image-id ami-01234567890abcdef
```

명령은 성공 시 출력을 반환하지 않습니다.

> **Important**
>
> 다음 오류가 발생하면 AWS CLI 버전을 업데이트해야 할 수 있습니다. 자세한 내용은 명령을 찾을 수 없음 오류를 참조하세요.

```
aws.exe: error: argument operation: Invalid choice, valid choices are: ...
```

Amazon EventBridge를 사용하여 휴지통 모니터링

휴지통은 보존 규칙에 따라 수행된 작업에 대한 이벤트를 Amazon EventBridge로 전송합니다. EventBridge에서는 이러한 이벤트에 대한 응답으로 프로그래밍 작업을 시작하는 규칙을 설정할 수 있습니다. 예를 들어 보존 규칙이 잠금 해제되고 해당 보존 규칙이 잠금 해제 지연 기간에 접어들면, 이메일로 알림을 보내는 EventBridge 규칙을 생성할 수 있습니다. 자세한 내용은 이벤트에 응답하는 Amazon EventBridge 규칙 생성을 참조하세요.

EventBridge의 이벤트는 JSON 객체로 표현됩니다. 이 이벤트에 고유한 필드는 JSON 객체의 `detail` 섹션에 포함되어 있습니다. `event` 필드에는 이벤트 이름이 포함됩니다. `result` 필드에는 이벤트를 시작한 작업의 완료 상태가 포함됩니다. 자세한 내용은 Amazon EventBridge 사용 설명서의 Amazon EventBridge 이벤트 패턴을 참조하세요.

Amazon EventBridge에 대한 자세한 내용은 Amazon EventBridge 사용 설명서에서 Amazon EventBridge란 무엇입니까?를 참조하세요.

이벤트

- RuleLocked
- RuleChangeAttempted
- RuleUnlockScheduled
- RuleUnlockingNotice
- RuleUnlocked

RuleLocked

다음은 보존 규칙이 성공적으로 잠겼을 때 휴지통에서 생성되는 이벤트의 예입니다. 이 이벤트는 CreateRule 및 LockRule 요청을 통해 생성될 수 있습니다. 이벤트를 생성한 API가 `api-name` 필드에 기록됩니다.

```
{
 "version": "0",
 "id": "exampleb-b491-4cf7-a9f1-bf370example",
```

```
 "detail-type": "Recycle Bin Rule Locked",
 "source": "aws.rbin",
 "account": "123456789012",
 "time": "2022-08-10T16:37:50Z",
 "region": "us-west-2",
 "resources": [
 "arn:aws:rbin:us-west-2:123456789012:rule/a12345abcde"
 ],
 "detail":
 {
 "detail-version": " 1.0.0",
 "rule-id": "a12345abcde",
 "rule-description": "locked account level rule",
 "unlock-delay-period": "30 days",
 "api-name": "CreateRule"
 }
 }
```


RuleChangeAttempted

다음은 잠긴 규칙을 수정하거나 삭제하려는 시도가 실패할 경우 휴지통에서 생성하는 이벤트의 예입니다. 이 이벤트는 DeleteRule 및 UpdateRule 요청을 통해 생성될 수 있습니다. 이벤트를 생성한 API가 `api-name` 필드에 기록됩니다.


```
{
 "version": "0",
 "id": "exampleb-b491-4cf7-a9f1-bf370example",
 "detail-type": "Recycle Bin Rule Change Attempted",
 "source": "aws.rbin",
 "account": "123456789012",
 "time": "2022-08-10T16:37:50Z",
 "region": "us-west-2",
 "resources": [
 "arn:aws:rbin:us-west-2:123456789012:rule/a12345abcde"
 ],
 "detail":
 {
 "detail-version": " 1.0.0",
 "rule-id": "a12345abcde",
 "rule-description": "locked account level rule",
 "unlock-delay-period": "30 days",
 "api-name": "DeleteRule"
 }
```

```
 }
```

RuleUnlockScheduled

다음은 보존 규칙이 잠금 해제되고 잠금 해제 지연 기간이 시작될 때 휴지통에서 생성되는 이벤트의 예입니다.

```
{
 "version": "0",
 "id": "exampleb-b491-4cf7-a9f1-bf370example",
 "detail-type": "Recycle Bin Rule Unlock Scheduled",
 "source": "aws.rbin",
 "account": "123456789012",
 "time": "2022-08-10T16:37:50Z",
 "region": "us-west-2",
 "resources": [
 "arn:aws:rbin:us-west-2:123456789012:rule/a12345abcde"
 ],
 "detail":
 {
 "detail-version": " 1.0.0",
 "rule-id": "a12345abcde",
 "rule-description": "locked account level rule",
 "unlock-delay-period": "30 days",
 "scheduled-unlock-time": "2022-09-10T16:37:50Z",
 }
 }
```

RuleUnlockingNotice

다음은 보존 규칙이 잠금 해제 지연 기간 내에 있는 동안 잠금 해제 지연 기간이 만료되기 전날까지 휴지통에서 매일 생성되는 이벤트의 예입니다.

```
{
 "version": "0",
 "id": "exampleb-b491-4cf7-a9f1-bf370example",
 "detail-type": "Recycle Bin Rule Unlocking Notice",
 "source": "aws.rbin",
 "account": "123456789012",
 "time": "2022-08-10T16:37:50Z",
 "region": "us-west-2",
```

```
 "resources": [
 "arn:aws:rbin:us-west-2:123456789012:rule/a12345abcde"
 ],
 "detail":
 {
 "detail-version": " 1.0.0",
 "rule-id": "a12345abcde",
 "rule-description": "locked account level rule",
 "unlock-delay-period": "30 days",
 "scheduled-unlock-time": "2022-09-10T16:37:50Z"
 }
 }
```

RuleUnlocked

다음은 보존 규칙의 잠금 해제 지연 기간이 만료되어 보존 규칙을 수정하거나 삭제할 수 있을 때 휴지통에서 생성되는 이벤트의 예입니다.

```
{
 "version": "0",
 "id": "exampleb-b491-4cf7-a9f1-bf370example",
 "detail-type": "Recycle Bin Rule Unlocked",
 "source": "aws.rbin",
 "account": "123456789012",
 "time": "2022-08-10T16:37:50Z",
 "region": "us-west-2",
 "resources": [
 "arn:aws:rbin:us-west-2:123456789012:rule/a12345abcde"
 ],
 "detail":
 {
 "detail-version": " 1.0.0",
 "rule-id": "a12345abcde",
 "rule-description": "locked account level rule",
 "unlock-delay-period": "30 days",
 "scheduled-unlock-time": "2022-09-10T16:37:50Z"
 }
 }
```

를 사용하여 휴지통 모니터링 AWS CloudTrail

휴지통 서비스는와 통합됩니다 AWS CloudTrail. CloudTrail은 사용자, 역할 또는 서비스가 수행한 작업에 대한 레코드를 제공하는 AWS 서비스입니다. CloudTrail은 휴지통에서 수행되는 모든 API 호출을 이벤트로 캡처합니다. 추적을 생성하는 경우 CloudTrail 이벤트를 Amazon Simple Storage Service(Amazon S3) 버킷으로 지속적으로 전송할 수 있습니다. 추적을 구성하지 않은 경우에도 CloudTrail 콘솔의 [이벤트 기록(Event history)]에서 최신 관리 이벤트를 볼 수 있습니다. CloudTrail에서 수집한 정보를 사용하여 휴지통에 수행된 요청, 요청이 수행된 IP 주소, 요청을 수행한 사람, 요청이 수행된 시간 및 추가 세부 정보를 확인할 수 있습니다.

CloudTrail에 대한 자세한 내용은 AWS CloudTrail 사용자 안내서를 참조하세요.

CloudTrail의 휴지통 정보

AWS 계정을 생성할 때 계정에서 CloudTrail이 활성화됩니다. 지원되는 이벤트 활동이 휴지통에서 발생하면 해당 활동은 이벤트 기록의 다른 AWS 서비스 이벤트와 함께 CloudTrail 이벤트에 기록됩니다. AWS 계정에서 최근 이벤트를 보고 검색하고 다운로드할 수 있습니다. 자세한 설명은 CloudTrail 이벤트 기록으로 이벤트 보기를 참조하세요.

휴지통에 대한 이벤트를 포함하여 AWS 계정에 이벤트를 지속적으로 기록하려면 추적을 생성합니다. 추적은 CloudTrail이 S3 버킷으로 로그 파일을 전송할 수 있도록 합니다. 기본적으로 콘솔에서 추적을 생성하면 추적이 모든 AWS 리전에 적용됩니다. 추적은 AWS 파티션의 모든 리전에서 이벤트를 로깅하고 지정한 S3 버킷으로 로그 파일을 전송합니다. 또한 CloudTrail 로그에서 수집된 이벤트 데이터를 추가로 분석하고 조치를 취하도록 다른 AWS 서비스를 구성할 수 있습니다. 자세한 내용은 AWS CloudTrail 사용 설명서의 추적 생성 개요를 참조하세요.

지원되는 API 작업

휴지통의 경우 CloudTrail을 사용하여 다음 API 작업을 관리 이벤트로 기록할 수 있습니다.

- CreateRule
- UpdateRule
- GetRules
- ListRule
- DeleteRule
- TagResource
- UntagResource
- ListTagsForResource

- LockRule
- UnlockRule

관리 이벤트 로깅에 대한 자세한 내용은 CloudTrail 사용 설명서의 추적에 대한 관리 이벤트 로깅을 참조하세요.

자격 증명 정보

모든 이벤트 및 로그 항목에는 요청을 생성한 사용자에 대한 정보가 들어 있습니다. 자격 증명을 이용하면 다음을 쉽게 판단할 수 있습니다.

- 요청을 루트 사용자로 했는지 사용자 보안 인증으로 했는지 여부.
- 역할 또는 페더레이션 사용자의 임시 자격 증명을 사용하여 요청이 생성되었는지 여부.
- 요청이 다른 AWS 서비스에 의해 이루어졌는지 여부입니다.

자세한 내용은 CloudTrail userIdentityElement를 참조하세요.

휴지통 로그 파일 항목 이해

추적이란 지정한 S3 버킷에 이벤트를 로그 파일로 입력할 수 있도록 하는 구성입니다. CloudTrail 로그 파일에는 하나 이상의 로그 항목이 포함될 수 있습니다. 이벤트는 모든 소스로부터의 단일 요청을 나타내며 요청 작업, 작업 날짜와 시간, 요청 파라미터 등에 대한 정보가 들어 있습니다. CloudTrail 로그 파일은 퍼블릭 API 직접 호출의 주문 스택 트레이스가 아니므로 특정 순서로 표시되지 않습니다.

다음은 CloudTrail 로그 항목의 예제입니다.

CreateRule

```
{
 "eventVersion": "1.08",
 "userIdentity": {
 "type": "AssumedRole",
 "principalId": "123456789012",
 "arn": "arn:aws:iam::123456789012:root",
 "accountId": "123456789012",
 "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
 "sessionContext": {
   "sessionIssuer": {
     "type": "Role",
     "principalId": "123456789012",
```

```
    "arn": "arn:aws:iam::123456789012:role/Admin",
    "accountId": "123456789012",
    "userName": "Admin"
  },
  "webIdFederationData": {},
  "attributes": {
    "mfaAuthenticated": "false",
    "creationDate": "2021-08-02T21:43:38Z"
  }
}
},
"eventTime": "2021-08-02T21:45:22Z",
"eventSource": "rbin.amazonaws.com",
"eventName": "CreateRule",
"awsRegion": "us-west-2",
"sourceIPAddress": "123.123.123.123",
"userAgent": "aws-cli/1.20.9 Python/3.6.14
Linux/4.9.230-0.1.ac.224.84.332.metal1.x86_64 botocore/1.21.9",
"requestParameters": {
"retentionPeriod": {
  "retentionPeriodValue": 7,
  "retentionPeriodUnit": "DAYS"
},
"description": "Match all snapshots",
"resourceType": "EBS_SNAPSHOT"
},
"responseElements": {
"identifier": "jkrnexample"
},
"requestID": "ex0577a5-amc4-pl4f-ef51-50fdexample",
"eventID": "714fafex-2eam-42pl-913e-926d4example",
"readOnly": false,
"eventType": "AwsApiCall",
"managementEvent": true,
"eventCategory": "Management",
"recipientAccountId": "123456789012",
"tlsDetails": {
"tlsVersion": "TLSv1.2",
"cipherSuite": "ECDHE-RSA-AES128-GCM-SHA256",
"clientProvidedHostHeader": "rbin.us-west-2.amazonaws.com"
}
}
```

GetRule


```
{
 "eventVersion": "1.08",
 "userIdentity": {
 "type": "AssumedRole",
 "principalId": "123456789012",
 "arn": "arn:aws:iam::123456789012:root",
 "accountId": "123456789012",
 "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
 "sessionContext": {
   "sessionIssuer": {
     "type": "Role",
     "principalId": "123456789012",
     "arn": "arn:aws:iam::123456789012:role/Admin",
     "accountId": "123456789012",
     "userName": "Admin"
   },
   "webIdFederationData": {},
   "attributes": {
     "mfaAuthenticated": "false",
     "creationDate": "2021-08-02T21:43:38Z"
   }
 }
 },
 "eventTime": "2021-08-02T21:45:33Z",
 "eventSource": "rbin.amazonaws.com",
 "eventName": "GetRule",
 "awsRegion": "us-west-2",
 "sourceIPAddress": "123.123.123.123",
 "userAgent": "aws-cli/1.20.9 Python/3.6.14
Linux/4.9.230-0.1.ac.224.84.332.metal1.x86_64 botocore/1.21.9",
 "requestParameters": {
 "identifier": "jkrnexample"
 },
 "responseElements": null,
 "requestID": "ex0577a5-amc4-pl4f-ef51-50fdexample",
 "eventID": "714fafex-2eam-42pl-913e-926d4example",
 "readOnly": true,
 "eventType": "AwsApiCall",
 "managementEvent": true,
 "eventCategory": "Management",
 "recipientAccountId": "123456789012",
 "tlsDetails": {
```

```
 "tlsVersion": "TLSv1.2",
 "cipherSuite": "ECDHE-RSA-AES128-GCM-SHA256",
 "clientProvidedHostHeader": "rbin.us-west-2.amazonaws.com"
 }
 }
```


ListRules


```
{
 "eventVersion": "1.08",
 "userIdentity": {
 "type": "AssumedRole",
 "principalId": "123456789012",
 "arn": "arn:aws:iam::123456789012:root",
 "accountId": "123456789012",
 "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
 "sessionContext": {
   "sessionIssuer": {
     "type": "Role",
     "principalId": "123456789012",
     "arn": "arn:aws:iam::123456789012:role/Admin",
     "accountId": "123456789012",
     "userName": "Admin"
   },
   "webIdFederationData": {},
   "attributes": {
     "mfaAuthenticated": "false",
     "creationDate": "2021-08-02T21:43:38Z"
   }
 }
 },
 "eventTime": "2021-08-02T21:44:37Z",
 "eventSource": "rbin.amazonaws.com",
 "eventName": "ListRules",
 "awsRegion": "us-west-2",
 "sourceIPAddress": "123.123.123.123",
 "userAgent": "aws-cli/1.20.9 Python/3.6.14
 Linux/4.9.230-0.1.ac.224.84.332.metal1.x86_64 botocore/1.21.9",
 "requestParameters": {
 "resourceTags": [
   {
     "resourceTagKey": "test",
     "resourceTagValue": "test"
```

```
    }
  ]
  },
  "responseElements": null,
  "requestID": "ex0577a5-amc4-pl4f-ef51-50fdexample",
  "eventID": "714fafex-2eam-42pl-913e-926d4example",
  "readOnly": true,
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "eventCategory": "Management",
  "recipientAccountId": "123456789012",
  "tlsDetails": {
  "tlsVersion": "TLSv1.2",
  "cipherSuite": "ECDHE-RSA-AES128-GCM-SHA256",
  "clientProvidedHostHeader": "rbin.us-west-2.amazonaws.com"
  }
  }
```

UpdateRule

```
{
  "eventVersion": "1.08",
  "userIdentity": {
  "type": "AssumedRole",
  "principalId": "123456789012",
  "arn": "arn:aws:iam::123456789012:root",
  "accountId": "123456789012",
  "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
  "sessionContext": {
    "sessionIssuer": {
      "type": "Role",
      "principalId": "123456789012",
      "arn": "arn:aws:iam::123456789012:role/Admin",
      "accountId": "123456789012",
      "userName": "Admin"
    },
    "webIdFederationData": {},
    "attributes": {
      "mfaAuthenticated": "false",
      "creationDate": "2021-08-02T21:43:38Z"
    }
  }
  },
```

```
 "eventTime": "2021-08-02T21:46:03Z",
 "eventSource": "rbin.amazonaws.com",
 "eventName": "UpdateRule",
 "awsRegion": "us-west-2",
 "sourceIPAddress": "123.123.123.123",
 "userAgent": "aws-cli/1.20.9 Python/3.6.14
 Linux/4.9.230-0.1.ac.224.84.332.metal1.x86_64 botocore/1.21.9",
 "requestParameters": {
 "identifier": "jkrnexample",
 "retentionPeriod": {
   "retentionPeriodValue": 365,
   "retentionPeriodUnit": "DAYS"
 },
 "description": "Match all snapshots",
 "resourceType": "EBS_SNAPSHOT"
 },
 "responseElements": null,
 "requestID": "ex0577a5-amc4-pl4f-ef51-50fdexample",
 "eventID": "714fafex-2eam-42pl-913e-926d4example",
 "readOnly": false,
 "eventType": "AwsApiCall",
 "managementEvent": true,
 "eventCategory": "Management",
 "recipientAccountId": "123456789012",
 "tlsDetails": {
 "tlsVersion": "TLSv1.2",
 "cipherSuite": "ECDHE-RSA-AES128-GCM-SHA256",
 "clientProvidedHostHeader": "rbin.us-west-2.amazonaws.com"
 }
 }
```


DeleteRule


```
{
 "eventVersion": "1.08",
 "userIdentity": {
 "type": "AssumedRole",
 "principalId": "123456789012",
 "arn": "arn:aws:iam::123456789012:root",
 "accountId": "123456789012",
 "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
 "sessionContext": {
   "sessionIssuer": {
```

```
    "type": "Role",
    "principalId": "123456789012",
    "arn": "arn:aws:iam::123456789012:role/Admin",
    "accountId": "123456789012",
    "userName": "Admin"
  },
  "webIdFederationData": {},
  "attributes": {
    "mfaAuthenticated": "false",
    "creationDate": "2021-08-02T21:43:38Z"
  }
}
},
"eventTime": "2021-08-02T21:46:25Z",
"eventSource": "rbin.amazonaws.com",
"eventName": "DeleteRule",
"awsRegion": "us-west-2",
"sourceIPAddress": "123.123.123.123",
"userAgent": "aws-cli/1.20.9 Python/3.6.14
Linux/4.9.230-0.1.ac.224.84.332.metal1.x86_64 botocore/1.21.9",
"requestParameters": {
"identifier": "jkrnexample"
},
"responseElements": null,
"requestID": "ex0577a5-amc4-pl4f-ef51-50fdexample",
"eventID": "714fafex-2eam-42pl-913e-926d4example",
"readOnly": false,
"eventType": "AwsApiCall",
"managementEvent": true,
"eventCategory": "Management",
"recipientAccountId": "123456789012",
"tlsDetails": {
"tlsVersion": "TLSv1.2",
"cipherSuite": "ECDHE-RSA-AES128-GCM-SHA256",
"clientProvidedHostHeader": "rbin.us-west-2.amazonaws.com"
}
}
```


TagResource


```
{
 "eventVersion": "1.08",
 "userIdentity": {
```

```
"type": "AssumedRole",
"principalId": "123456789012",
"arn": "arn:aws:iam::123456789012:root",
"accountId": "123456789012",
"accessKeyId": "AKIAIOSFODNN7EXAMPLE",
"sessionContext": {
  "sessionIssuer": {
    "type": "Role",
    "principalId": "123456789012",
    "arn": "arn:aws:iam::123456789012:role/Admin",
    "accountId": "123456789012",
    "userName": "Admin"
  },
  "webIdFederationData": {},
  "attributes": {
    "mfaAuthenticated": "false",
    "creationDate": "2021-10-22T21:38:34Z"
  }
}
},
"eventTime": "2021-10-22T21:43:15Z",
"eventSource": "rbin.amazonaws.com",
"eventName": "TagResource",
"awsRegion": "us-west-2",
"sourceIPAddress": "123.123.123.123",
"userAgent": "aws-cli/1.20.26 Python/3.6.14
Linux/4.9.273-0.1.ac.226.84.332.metal1.x86_64 botocore/1.21.26",
"requestParameters": {
"resourceArn": "arn:aws:rbin:us-west-2:123456789012:rule/ABCDEF01234",
"tags": [
  {
    "key": "purpose",
    "value": "production"
  }
]
},
"responseElements": null,
"requestID": "examplee-7962-49ec-8633-795efexample",
"eventID": "example4-6826-4c0a-bdec-0bab1example",
"readOnly": false,
"eventType": "AwsApiCall",
"managementEvent": true,
"eventCategory": "Management",
"recipientAccountId": "123456789012",
```

```
 "tlsDetails": {
 "tlsVersion": "TLSv1.2",
 "cipherSuite": "ECDHE-RSA-AES128-GCM-SHA256",
 "clientProvidedHostHeader": "rbin.us-west-2.amazonaws.com"
 }
 }
```

UntagResource

```
{
 "eventVersion": "1.08",
 "userIdentity": {
 "type": "AssumedRole",
 "principalId": "123456789012",
 "arn": "arn:aws:iam::123456789012:root",
 "accountId": "123456789012",
 "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
 "sessionContext": {
   "sessionIssuer": {
     "type": "Role",
     "principalId": "123456789012",
     "arn": "arn:aws:iam::123456789012:role/Admin",
     "accountId": "123456789012",
     "userName": "Admin"
   },
   "webIdFederationData": {},
   "attributes": {
     "mfaAuthenticated": "false",
     "creationDate": "2021-10-22T21:38:34Z"
   }
 }
 },
 "eventTime": "2021-10-22T21:44:16Z",
 "eventSource": "rbin.amazonaws.com",
 "eventName": "UntagResource",
 "awsRegion": "us-west-2",
 "sourceIPAddress": "123.123.123.123",
 "userAgent": "aws-cli/1.20.26 Python/3.6.14
 Linux/4.9.273-0.1.ac.226.84.332.metal1.x86_64 botocore/1.21.26",
 "requestParameters": {
 "resourceArn": "arn:aws:rbin:us-west-2:123456789012:rule/ABCDEF01234",
 "tagKeys": [
   "purpose"
```

```
]
},
"responseElements": null,
"requestID": "example7-6c1e-4f09-9e46-bb957example",
"eventID": "example6-75ff-4c94-a1cd-4d5f5example",
"readOnly": false,
"eventType": "AwsApiCall",
"managementEvent": true,
"eventCategory": "Management",
"recipientAccountId": "123456789012",
"tlsDetails": {
"tlsVersion": "TLSv1.2",
"cipherSuite": "ECDHE-RSA-AES128-GCM-SHA256",
"clientProvidedHostHeader": "rbin.us-west-2.amazonaws.com"
}
}
```

ListTagsForResource

```
{
"eventVersion": "1.08",
"userIdentity": {
"type": "AssumedRole",
"principalId": "123456789012",
"arn": "arn:aws:iam::123456789012:root",
"accountId": "123456789012",
"accessKeyId": "AKIAIOSFODNN7EXAMPLE",
"sessionContext": {
  "sessionIssuer": {
    "type": "Role",
    "principalId": "123456789012",
    "arn": "arn:aws:iam::123456789012:role/Admin",
    "accountId": "123456789012",
    "userName": "Admin"
  },
  "webIdFederationData": {},
  "attributes": {
    "mfaAuthenticated": "false",
    "creationDate": "2021-10-22T21:38:34Z"
  }
}
},
"eventTime": "2021-10-22T21:42:31Z",
```

```
"eventSource": "rbin.amazonaws.com",
"eventName": "ListTagsForResource",
"awsRegion": "us-west-2",
"sourceIPAddress": "123.123.123.123",
"userAgent": "aws-cli/1.20.26 Python/3.6.14
Linux/4.9.273-0.1.ac.226.84.332.metal1.x86_64 botocore/1.21.26",
"requestParameters": {
"resourceArn": "arn:aws:rbin:us-west-2:123456789012:rule/ABCDEF01234"
},
"responseElements": null,
"requestID": "example8-10c7-43d4-b147-3d9d9example",
"eventID": "example2-24fc-4da7-a479-c9748example",
"readOnly": true,
"eventType": "AwsApiCall",
"managementEvent": true,
"eventCategory": "Management",
"recipientAccountId": "123456789012",
"tlsDetails": {
"tlsVersion": "TLSv1.2",
"cipherSuite": "ECDHE-RSA-AES128-GCM-SHA256",
"clientProvidedHostHeader": "rbin.us-west-2.amazonaws.com"
}
}
```


LockRule


```
{
"eventVersion": "1.08",
"userIdentity": {
"type": "AssumedRole",
"principalId": "123456789012",
"arn": "arn:aws:iam::123456789012:root",
"accountId": "123456789012",
"accessKeyId": "AKIAIOSFODNN7EXAMPLE",
"sessionContext": {
  "sessionIssuer": {
    "type": "Role",
    "principalId": "123456789012",
    "arn": "arn:aws:iam::123456789012:role/Admin",
    "accountId": "123456789012",
    "userName": "Admin"
  },
  "webIdFederationData": {},
```

```
  "attributes": {
    "creationDate": "2022-10-25T00:45:11Z",
    "mfaAuthenticated": "false"
  }
}
},
"eventTime": "2022-10-25T00:45:19Z",
"eventSource": "rbin.amazonaws.com",
"eventName": "LockRule",
"awsRegion": "us-west-2",
"sourceIPAddress": "123.123.123.123",
"userAgent": "python-requests/2.25.1",
"requestParameters": {
"identifier": "jkrnexample",
"lockConfiguration": {
  "unlockDelay": {
    "unlockDelayValue": 7,
    "unlockDelayUnit": "DAYS"
  }
}
},
"responseElements": {
"identifier": "jkrnexample",
"description": "",
"resourceType": "EBS_SNAPSHOT",
"retentionPeriod": {
  "retentionPeriodValue": 7,
  "retentionPeriodUnit": "DAYS"
},
"resourceTags": [],
"status": "available",
"lockConfiguration": {
  "unlockDelay": {
    "unlockDelayValue": 7,
    "unlockDelayUnit": "DAYS"
  }
},
"lockState": "locked"
},
"requestID": "ex0577a5-amc4-pl4f-ef51-50fdexample",
"eventID": "714fafex-2eam-42pl-913e-926d4example",
"readOnly": false,
"eventType": "AwsApiCall",
"managementEvent": true,
```

```
 "recipientAccountId": "123456789012",
 "eventCategory": "Management",
 "tlsDetails": {
 "tlsVersion": "TLSv1.2",
 "cipherSuite": "ECDHE-RSA-AES128-GCM-SHA256",
 "clientProvidedHostHeader": "rbin.us-west-2.amazonaws.com"
 }
 }
```


UnlockRule


```
{
 "eventVersion": "1.08",
 "userIdentity": {
 "type": "AssumedRole",
 "principalId": "123456789012",
 "arn": "arn:aws:iam::123456789012:root",
 "accountId": "123456789012",
 "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
 "sessionContext": {
   "sessionIssuer": {
     "type": "Role",
     "principalId": "123456789012",
     "arn": "arn:aws:iam::123456789012:role/Admin",
     "accountId": "123456789012",
     "userName": "Admin"
   },
   "webIdFederationData": {},
   "attributes": {
     "creationDate": "2022-10-25T00:45:11Z",
     "mfaAuthenticated": "false"
   }
 }
 },
 "eventTime": "2022-10-25T00:46:17Z",
 "eventSource": "rbin.amazonaws.com",
 "eventName": "UnlockRule",
 "awsRegion": "us-west-2",
 "sourceIPAddress": "123.123.123.123",
 "userAgent": "python-requests/2.25.1",
 "requestParameters": {
 "identifier": "jkrnexample"
 },
```

```
"responseElements": {
"identifier": "jkrnexample",
"description": "",
"resourceType": "EC2_IMAGE",
"retentionPeriod": {
  "retentionPeriodValue": 7,
  "retentionPeriodUnit": "DAYS"
},
"resourceTags": [],
"status": "available",
"lockConfiguration": {
  "unlockDelay": {
    "unlockDelayValue": 7,
    "unlockDelayUnit": "DAYS"
  }
},
"lockState": "pending_unlock",
"lockEndTime": "Nov 1, 2022, 12:46:17 AM"
},
"requestID": "ex0577a5-amc4-pl4f-ef51-50fdexample",
"eventID": "714fafex-2eam-42pl-913e-926d4example",
"readOnly": false,
"eventType": "AwsApiCall",
"managementEvent": true,
"recipientAccountId": "123456789012",
"eventCategory": "Management",
"tlsDetails": {
"tlsVersion": "TLSv1.2",
"cipherSuite": "ECDHE-RSA-AES128-GCM-SHA256",
"clientProvidedHostHeader": "rbin.us-west-2.amazonaws.com"
}
}
```


휴지통의 서비스 엔드포인트

엔드포인트는 AWS 웹 서비스의 진입점 역할을 하는 URL입니다. 휴지통은 다음과 같은 엔드포인트 유형을 지원합니다.

- IPv4 엔드포인트
- IPv4 및 IPv6를 모두 지원하는 이중 스택 엔드포인트
- FIPS 엔드포인트

요청 시에, 사용할 엔드포인트와 리전을 지정할 수 있습니다. 엔드포인트를 지정하지 않으면 기본적으로 IPv4 엔드포인트가 사용됩니다. 다른 엔드포인트 유형을 사용하려면 요청에서 이를 지정해야 합니다. 이렇게 하는 방법의 예제는 엔드포인트 지정 섹션을 참조하세요.

휴지통에 대한 자세한 내용은 Amazon Web Services 일반 참조의 휴지통 엔드포인트를 참조하세요.

주제

- IPv4 엔드포인트
- 이중 스택(IPv4 및 IPv6) 엔드포인트
- FIPS 엔드포인트
- 엔드포인트 지정

IPv4 엔드포인트

IPv4 엔드포인트는 IPv4 트래픽만 지원합니다. IPv4 엔드포인트는 모든 리전에 사용할 수 있습니다.

엔드포인트 이름의 일부로 리전을 지정해야 합니다. 엔드포인트 이름에는 다음 명명 규칙이 사용됩니다.

- rbin.*region*.amazonaws.com

예를 들어 미국 동부(버지니아 북부) 리전의 IPv4 엔드포인트는 `rbin.us-east-1.amazonaws.com`입니다.

이중 스택(IPv4 및 IPv6) 엔드포인트

이중 스택 엔드포인트는 IPv4 트래픽과 IPv6 트래픽을 모두 지원합니다. 이중 스택 엔드포인트는 모든 리전에 사용할 수 있습니다.

IPv6를 사용하려면 이중 스택 엔드포인트를 사용해야 합니다. 이중 스택 엔드포인트에 요청하는 경우, 엔드포인트 URL이 네트워크 및 클라이언트에서 사용하는 프로토콜에 따라 IPv6 또는 IPv4 주소로 확인됩니다.

엔드포인트 이름의 일부로 리전을 지정해야 합니다. 이중 스택 엔드포인트 이름에는 다음 명명 규칙이 사용됩니다.

- `rbin.`*region*`.api.aws`

예를 들어 미국 동부(버지니아 북부) 리전의 듀얼 스택 엔드포인트는 rbin.us-east-1.api.aws입니다.

FIPS 엔드포인트

휴지통은 다음 리전에 대해 FIPS 검증을 거친 IPv4 및 이중 스택(IPv4 및 IPv6) 엔드포인트를 제공합니다.

- us-east-1 - 미국 동부(버지니아 북부)
- us-east-2 - 미국 동부(오하이오)
- us-west-1 - 미국 서부(캘리포니아 북부)
- us-west-2 - 미국 서부(오레곤)
- ca-central-1 - 캐나다(중부)
- ca-west-1 - 캐나다 서부(캘거리)
- us-gov-east-1 — AWS GovCloud (US-East)
- us-gov-west-1 — AWS GovCloud (US-West)

FIPS IPv4 엔드포인트에는 rbin-fips.*region*.amazonaws.com이라는 명명 규칙이 사용됩니다. 예를 들어 미국 동부(버지니아 북부) 리전의 FIPS IPv4 엔드포인트는 rbin-fips.us-east-1.amazonaws.com입니다.

FIPS 이중 스택 엔드포인트에는 rbin-fips.*region*.api.aws라는 명명 규칙이 사용됩니다. 예를 들어 미국 동부(버지니아 북부) 리전의 FIPS 듀얼 스택 엔드포인트는 rbin-fips.us-east-1.api.aws입니다.

엔드포인트 지정

다음 예는 AWS CLI를 사용하여 us-east-2 리전의 엔드포인트를 지정하는 방법을 보여줍니다.

- 이중 스택

```
aws rbin get-rule \
--identifier rule_id \
--endpoint-url https://rbin.us-east-2.api.aws
```

- IPv4

```
aws rbin get-rule \
```

```
--identifier rule_id \
--endpoint-url https://rbin.us-east-2.amazonaws.com
```

VPC와 휴지통 간에 프라이빗 연결 생성

AWS PrivateLink에서 제공되는 인터페이스 VPC 엔드포인트를 생성하여 VPC와 휴지통 간에 프라이빗 연결을 설정할 수 있습니다. 인터넷 게이트웨이, NAT 디바이스, VPN 연결 또는 Direct Connect 연결을 사용하지 않고 VPC에 있는 것처럼 휴지통에 액세스할 수 있습니다. VPC의 인스턴스는 휴지통과 통신하는 데 퍼블릭 IP 주소를 필요로 하지 않습니다.

인터페이스 엔드포인트에 대해 사용 설정하는 각 서브넷에서 엔드포인트 네트워크 인터페이스를 생성합니다.

자세한 내용은 AWS PrivateLink 가이드의를 통한 AWS 서비스 액세스를 AWS PrivateLink 참조하세요.

휴지통용 인터페이스 VPC 엔드포인트 생성

Amazon VPC 콘솔 또는 AWS CLI를 사용하여 휴지통용 VPC 엔드포인트를 생성할 수 있습니다. 자세한 정보는 AWS PrivateLink 가이드의 VPC 엔드포인트 생성을 참조하세요.

다음 서비스 이름을 사용하여 휴지통용 VPC 엔드포인트를 생성합니다.
`com.amazonaws.region.rbin`

엔드포인트에 프라이빗 DNS를 사용하도록 설정하는 경우, 리전의 기본 DNS 이름(예: `rbin.us-east-1.amazonaws.com`)을 사용하여 휴지통에 API 요청을 할 수 있습니다.

휴지통에 대한 VPC 엔드포인트 정책 생성

기본적으로 엔드포인트를 통해 휴지통에 대한 전체 액세스가 허용됩니다. VPC 엔드포인트 정책을 사용하여 인터페이스 엔드포인트에 대한 액세스를 제어할 수 있습니다. 휴지통에 대한 액세스를 제어하는 VPC 엔드포인트에 엔드포인트 정책을 연결할 수 있습니다. 이 정책은 다음 정보를 지정합니다.

- 작업을 수행할 수 있는 위탁자.
- 수행할 수 있는 작업.
- 작업을 수행할 수 있는 리소스.

자세한 내용은 Amazon VPC 사용 설명서의 VPC 종단점을 통해 서비스에 대한 액세스 제어를 참조하세요.

```
{
    "Statement": [
    {
        "Effect": "Allow",
        "Action": "rbin:*",
        "Resource": "*",
        "Principal": "*"
    },
    {
        "Effect": "Deny",
        "Action": "rbin:DeleteRule",
        "Resource": "*",
        "Principal": "*",
        "Condition": {
            "StringEquals" : {
                "rbin:Attribute/ResourceType": "EBS_SNAPSHOT"
            }
        }
    }]
}
```

Amazon EBS의 보안

의 클라우드 보안 AWS 이 최우선 순위입니다. AWS 고객은 보안에 가장 민감한 조직의 요구 사항을 충족하도록 구축된 데이터 센터 및 네트워크 아키텍처의 이점을 누릴 수 있습니다.

보안은 AWS 와 사용자 간의 공동 책임입니다. 공동 책임 모델은 이 사항을 클라우드의 보안 및 클라우드 내 보안으로 설명합니다.

- 클라우드 보안 - AWS 는에서 AWS 서비스를 실행하는 인프라를 보호할 책임이 있습니다 AWS 클라우드. AWS 또한는 안전하게 사용할 수 있는 서비스를 제공합니다. 타사 감사자는 AWS 규정 준수 프로그램 일환으로 보안의 효과를 정기적으로 테스트하고 확인합니다. Amazon Elastic Block Store에 적용되는 규정 준수 프로그램에 대한 자세한 내용은 규정 준수 프로그램 AWS 제공 범위 내 서비스 규정 준수 프로그램 .
- 클라우드의 보안 - 사용자의 책임은 사용하는 AWS 서비스에 따라 결정됩니다. 또한 귀하는 귀사의 데이터 민감도, 귀사의 요구 사항, 관련 법률 및 규정을 비롯한 기타 요소에 대해서도 책임이 있습니다.

이 설명서는 Amazon EBS를 사용할 때 공동 책임 모델을 적용하는 방법을 이해하는 데 도움이 됩니다. 다음 주제에서는 보안 및 규정 준수 목표에 맞게 Amazon EBS를 구성하는 방법을 보여줍니다. 또한 Amazon EBS 리소스를 모니터링하고 보호하는 데 도움이 되는 다른 AWS 서비스를 사용하는 방법을 알아봅니다.

주제

- Amazon EBS의 데이터 보호
- Amazon EBS의 자격 증명 및 액세스 관리
- Amazon EBS에 대한 규정 준수 확인
- Amazon EBS의 데이터 복원력

Amazon EBS의 데이터 보호

AWS 공동 책임 모델 Amazon Elastic Block Store의 데이터 보호에 적용됩니다. 이 모델에 설명된 대로 AWS 는 모든를 실행하는 글로벌 인프라를 보호할 책임이 있습니다 AWS 클라우드. 사용자는 이 인프라에 호스팅되는 콘텐츠에 대한 통제 권한을 유지할 책임이 있습니다. 사용하는 AWS 서비스 의 보안 구성과 관리 태스크에 대한 책임도 사용자에게 있습니다. 데이터 프라이버시에 대한 자세한 내용은 데

이터 프라이버시 FAQ 참조하세요. 유럽의 데이터 보호에 대한 자세한 내용은 일반 데이터 보호 규정 (GDPR) 센터를 참조하세요.

데이터 보호를 위해 자격 증명을 보호하고 AWS 계정 AWS IAM Identity Center 또는 AWS Identity and Access Management (IAM)를 사용하여 개별 사용자를 설정하는 것이 좋습니다. 이렇게 하면 개별 사용자에게 자신의 직무를 충실히 이행하는 데 필요한 권한만 부여됩니다. 또한 다음과 같은 방법으로 데이터를 보호하는 것이 좋습니다.

- 각 계정에 다중 인증(MFA)을 사용합니다.
- SSL/TLS를 사용하여 AWS 리소스와 통신합니다. TLS 1.2는 필수이며 TLS 1.3을 권장합니다.
- 를 사용하여 API 및 사용자 활동 로깅을 설정합니다 AWS CloudTrail. CloudTrail 추적을 사용하여 AWS 활동을 캡처하는 방법에 대한 자세한 내용은 AWS CloudTrail 사용 설명서의 CloudTrail 추적 작업을 참조하세요.
- 내부의 모든 기본 보안 제어와 함께 AWS 암호화 솔루션을 사용합니다 AWS 서비스.
- Amazon S3에 저장된 민감한 데이터를 검색하고 보호하는 데 도움이 되는 Amazon Macie와 같은 고급 관리형 보안 서비스를 사용합니다.
- 명령줄 인터페이스 또는 API를 AWS 통해에 액세스할 때 FIPS 140-3 검증 암호화 모듈이 필요한 경우 FIPS 엔드포인트를 사용합니다. 사용 가능한 FIPS 엔드포인트에 대한 자세한 내용은 연방 정보 처리 표준(FIPS) 140-3을 참조하세요.

고객의 이메일 주소와 같은 기밀 정보나 중요한 정보는 태그나 이름 필드와 같은 자유 형식 텍스트 필드에 입력하지 않는 것이 좋습니다. 여기에는 Amazon EBS 또는 기타 AWS 서비스 에서 콘솔, API AWS CLI또는 AWS SDKs를 사용하여 작업하는 경우가 포함됩니다. 이름에 사용되는 태그 또는 자유 형식 텍스트 필드에 입력하는 모든 데이터는 청구 또는 진단 로그에 사용될 수 있습니다. 외부 서버에 URL을 제공할 때 해당 서버에 대한 요청을 검증하기 위해 자격 증명을 URL에 포함해서는 안 됩니다.

주제

- Amazon EBS 데이터 보안
- 저장 데이터 및 전송 데이터 암호화
- KMS 키 관리

Amazon EBS 데이터 보안

Amazon EBS 볼륨은 포맷되지 않은 원시 블록 디바이스로 제공됩니다. 이러한 디바이스는 EBS 인프라에서 생성된 논리적 디바이스입니다. Amazon EBS 서비스는 고객이 사용하거나 재사용하기 전에 디

바이스가 논리적으로 비어 있는지(즉, 원시 블록이 0으로 설정되거나 암호화 방식으로 가상랜덤 데이터가 포함되어 있는지) 확인합니다.

절차에서 DoD 5220.22-M(국립 산업 보안 프로그램 운영 설명서) 또는 NIST 800-88(미디어 삭제 지침)에 설명된 것과 같은 특정 방법을 사용하여 모든 데이터를 삭제해야 하는 경우 Amazon EBS에서 삭제할 수 있습니다. 사용 전후 또는 절차의 요구 사항에 따라 둘 다 데이터를 지울 수 있습니다. 해당 블록 수준 활동은 Amazon EBS 서비스 내의 기본 스토리지 미디어에 반영됩니다.

저장 데이터 및 전송 데이터 암호화

Amazon EBS 암호화는 암호화 키를 사용하여 Amazon EBS 볼륨과 Amazon EBS 스냅샷을 암호화할 수 있는 AWS Key Management Service 암호화 솔루션입니다. EBS 암호화 작업은 Amazon EC2 인스턴스를 호스팅하는 서버에서 발생합니다. 이렇게 하면 인스턴스와 연결된 볼륨 및 후속 스냅샷 간에 data-at-rest data-in-transit 모두 보호할 수 있습니다. 자세한 내용은 Amazon EBS 암호화 단원을 참조하십시오.

KMS 키 관리

암호화된 Amazon EBS 볼륨 또는 스냅샷을 생성할 때 AWS Key Management Service 키를 지정합니다. 기본적으로 Amazon EBS는 계정 및 리전()의 Amazon EBS에 관리 AWS 형 KMS 키를 사용합니다`aws/ebs`. 그러나 본인이 생성하고 관리하는 고객 관리형 KMS 키를 지정할 수 있습니다. 고객 관리형 KMS 키를 사용하면 KMS 키 생성, 교체 및 비활성화 기능을 포함하여 더 많은 유연성이 부여됩니다.

고객 관리형 KMS 키를 사용하려면 KMS 키 사용 권한을 사용자에게 부여해야 합니다. 자세한 내용은 사용자의 권한 단원을 참조하십시오.

> **Important**
>
> Amazon EBS에서는 대칭 KMS 키만 지원됩니다. 비대칭 KMS 키를 사용하여 Amazon EBS 볼륨과 스냅샷을 암호화할 수 없습니다. KMS 키의 대칭 또는 비대칭 여부를 확인하는 데 도움이 필요하면 비대칭 KMS 키 식별을 참조하세요.

Amazon EBS는 각 볼륨에 대해 사용자가 지정한 KMS 키로 암호화된 고유한 데이터 키를 생성 AWS KMS 하도록 요청합니다. Amazon EBS는 볼륨과 함께 암호화된 데이터 키를 저장합니다. 그런 다음 볼륨을 Amazon EC2 인스턴스에 연결하면 Amazon EBS가를 호출 AWS KMS 하여 데이터 키를 해독합니다. Amazon EBS에서는 하이퍼바이저 메모리의 일반 텍스트 데이터 키를 사용하여 모든 I/O를 볼륨으로 암호화합니다. 자세한 내용은 Amazon EBS의 암호화 방식 단원을 참조하십시오.

Amazon EBS의 자격 증명 및 액세스 관리

AWS Identity and Access Management (IAM)는 관리자가 AWS 리소스에 대한 액세스를 안전하게 제어하는 데 도움이 AWS 서비스 되는 입니다. IAM 관리자는 Amazon EBS 리소스를 사용하도록 인증(로그인)하고 권한을 부여(권한 있음)할 사용자를 제어합니다. IAM은 추가 비용 없이 사용할 수 AWS 서비스 있는 입니다.

주제

- 대상
- ID를 통한 인증
- 정책을 사용하여 액세스 관리
- Amazon EBS에서 IAM을 사용하는 방법
- Amazon EBS에 대한 IAM 정책 예제
- Amazon EBS 권한 부여 문제 해결

대상

AWS Identity and Access Management (IAM)를 사용하는 방법은 역할에 따라 다릅니다.

- 서비스 사용자 - 기능에 액세스할 수 없는 경우 관리자에게 권한 요청(참조Amazon EBS 권한 부여 문제 해결)
- 서비스 관리자 - 사용자 액세스 결정 및 권한 요청 제출(Amazon EBS에서 IAM을 사용하는 방법 참조)
- IAM 관리자 - 액세스를 관리하기 위한 정책 작성(Amazon EBS에 대한 IAM 정책 예제 참조)

ID를 통한 인증

인증은 자격 증명 자격 증명을 AWS 사용하여에 로그인하는 방법입니다. AWS 계정 루트 사용자, IAM 사용자 또는 IAM 역할을 수임하여 인증되어야 합니다.

AWS IAM Identity Center (IAM Identity Center), Single Sign-On 인증 또는 Google/Facebook 자격 증명과 같은 자격 증명 소스의 자격 증명을 사용하여 페더레이션 자격 증명으로 로그인할 수 있습니다. 로그인하는 방법에 대한 자세한 내용은 AWS 로그인 사용 설명서의 AWS 계정에 로그인하는 방법 섹션을 참조하세요.

프로그래밍 방식 액세스를 위해는 요청에 암호화 방식으로 서명할 수 있는 SDK 및 CLI를 AWS 제공합니다. 자세한 내용은 IAM 사용 설명서의 API 요청용AWS Signature Version 4 섹션을 참조하세요.

AWS 계정 루트 사용자

를 생성할 때 모든 AWS 서비스 및 리소스에 대한 완전한 액세스 권한이 있는 AWS 계정 theroot 사용자라는 하나의 로그인 자격 증명으로 AWS 계정시작합니다. 일상적인 태스크에 루트 사용자를 사용하지 않을 것을 강력히 권장합니다. 루트 사용자 자격 증명이 필요한 작업은 IAM 사용 설명서의 루트 사용자 자격 증명이 필요한 작업 섹션을 참조하세요.

페더레이션 ID

가장 좋은 방법은 인간 사용자에게 자격 증명 공급자와의 페더레이션을 사용하여 임시 자격 증명을 AWS 서비스 사용하여에 액세스하도록 요구하는 것입니다.

페더레이션 자격 증명은 엔터프라이즈 디렉터리, 웹 자격 증명 공급자 또는 자격 증명 소스의 자격 증명을 AWS 서비스 사용하여 Directory Service 에 액세스하는 사용자입니다. 페더레이션 ID는 임시 자격 증명을 제공하는 역할을 수임합니다.

중앙 집중식 액세스 관리를 위해 AWS IAM Identity Center를 추천합니다. 자세한 정보는 AWS IAM Identity Center 사용 설명서의 What is IAM Identity Center?를 참조하세요.

IAM 사용자 및 그룹

IAM 사용자는 단일 개인 또는 애플리케이션에 대한 특정 권한을 가진 ID입니다. 장기 자격 증명이 있는 IAM 사용자 대신 임시 자격 증명을 사용하는 것이 좋습니다. 자세한 내용은 IAM 사용 설명서의 자격 증명 공급자와의 페더레이션을 사용하여 임시 자격 증명을 AWS 사용하여에 액세스하도록 인간 사용자에게 요구하기를 참조하세요.

IAM 그룹은 IAM 사용자 모음을 지정하고 대규모 사용자 집합에 대한 관리 권한을 더 쉽게 만듭니다. 자세한 내용은 IAM 사용 설명서의 IAM 사용자 사용 사례 섹션을 참조하세요.

IAM 역할

IAM 역할은 임시 자격 증명을 제공하는 특정 권한이 있는 자격 증명입니다. 사용자에서 IAM 역할(콘솔)로 전환하거나 또는 API 작업을 호출하여 역할을 수임할 수 있습니다. AWS CLI AWS 자세한 내용은 IAM 사용 설명서의 역할 수임 방법을 참조하세요.

IAM 역할은 페더레이션 사용자 액세스, 임시 IAM 사용자 권한, 교차 계정 액세스, 교차 서비스 액세스 및 Amazon EC2에서 실행되는 애플리케이션에 유용합니다. 자세한 내용은 IAM 사용 설명서의 교차 계정 리소스 액세스를 참조하세요.

정책을 사용하여 액세스 관리

정책을 AWS 생성하고 자격 증명 또는 리소스에 연결하여 AWS 에서 액세스를 제어합니다. 정책은 자격 증명 또는 리소스와 연결될 때 권한을 정의합니다.는 보안 주체가 요청할 때 이러한 정책을 AWS 평가합니다. 대부분의 정책은에 JSON 문서 AWS 로 저장됩니다. JSON 정책 문서에 대한 자세한 내용은 IAM 사용 설명서의 JSON 정책 개요 섹션을 참조하세요.

정책을 사용하여 관리자는 어떤 보안 주체가 어떤 리소스에 대해 어떤 조건에서 작업을 수행할 수 있는지 정의하여 누가 무엇을 액세스할 수 있는지 지정합니다.

기본적으로 사용자 및 역할에는 어떠한 권한도 없습니다. IAM 관리자는 IAM 정책을 생성하고 사용자가 수임할 수 있는 역할에 추가합니다. IAM 정책은 작업을 수행하기 위해 사용하는 방법과 관계없이 작업에 대한 권한을 정의합니다.

ID 기반 정책

ID 기반 정책은 ID(사용자, 사용자 그룹 또는 역할)에 연결하는 JSON 권한 정책 문서입니다. 이러한 정책은 자격 증명이 수행할 수 있는 작업, 대상 리소스 및 이에 관한 조건을 제어합니다. ID 기반 정책을 생성하는 방법을 알아보려면 IAM 사용 설명서에서 고객 관리형 정책으로 사용자 지정 IAM 권한 정의를 참조하세요.

ID 기반 정책은 인라인 정책(단일 ID에 직접 포함) 또는 관리형 정책(여러 ID에 연결된 독립 실행형 정책)일 수 있습니다. 관리형 정책 또는 인라인 정책을 선택하는 방법을 알아보려면 IAM 사용 설명서의 관리형 정책 및 인라인 정책 중에서 선택 섹션을 참조하세요.

리소스 기반 정책

리소스 기반 정책은 리소스에 연결하는 JSON 정책 설명서입니다. 예를 들어 IAM 역할 신뢰 정책 및 Amazon S3 버킷 정책이 있습니다. 리소스 기반 정책을 지원하는 서비스에서 서비스 관리자는 이러한 정책을 사용하여 특정 리소스에 대한 액세스를 통제할 수 있습니다. 리소스 기반 정책에서 보안 주체를 지정해야 합니다.

리소스 기반 정책은 해당 서비스에 있는 인라인 정책입니다. 리소스 기반 정책에서는 IAM의 AWS 관리형 정책을 사용할 수 없습니다.

기타 정책 유형

AWS 는 보다 일반적인 정책 유형에서 부여한 최대 권한을 설정할 수 있는 추가 정책 유형을 지원합니다.

- 권한 경계 - ID 기반 정책에서 IAM 엔터티에 부여할 수 있는 최대 권한을 설정합니다. 자세한 정보는 IAM 사용 설명서의 IAM 엔터티의 권한 범위를 참조하세요.
- 서비스 제어 정책(SCP) - AWS Organizations내 조직 또는 조직 단위에 대한 최대 권한을 지정합니다. 자세한 내용은AWS Organizations 사용 설명서의 서비스 제어 정책을 참조하세요.
- 리소스 제어 정책(RCP) – 계정의 리소스에 사용할 수 있는 최대 권한을 설정합니다. 자세한 내용은 AWS Organizations 사용 설명서의 리소스 제어 정책(RCP)을 참조하세요.
- 세션 정책 – 역할 또는 페더레이션 사용자에 대해 임시 세션을 프로그래밍 방식으로 생성할 때 파라미터로 전달하는 고급 정책입니다. 자세한 내용은 IAM 사용 설명서의 세션 정책을 참조하세요.

여러 정책 유형

여러 정책 유형이 요청에 적용되는 경우, 결과 권한은 이해하기가 더 복잡합니다. 에서 여러 정책 유형이 관련될 때 요청을 허용할지 여부를 AWS 결정하는 방법을 알아보려면 IAM 사용 설명서의 정책 평가 로직을 참조하세요.

Amazon EBS에서 IAM을 사용하는 방법

IAM을 사용하여 Amazon EBS에 대한 액세스를 관리하기 전에 Amazon EBS에서 사용할 수 있는 IAM 기능을 알아봅니다.

Amazon Elastic Block Store에서 사용할 수 있는 IAM 기능

IAM 특성	Amazon EBS 지원
자격 증명 기반 정책	예
리소스 기반 정책	아니요
정책 작업	예
정책 리소스	예
정책 조건 키	예
ACL	아니요
ABAC(정책 내 태그)	부분적
임시 자격 증명	예

IAM 특성	Amazon EBS 지원
엔터티 권한	예
서비스 역할	예
서비스 연결 역할	아니요

Amazon EBS 및 기타 AWS 서비스가 대부분의 IAM 기능과 작동하는 방식을 전체적으로 알아보려면 IAM 사용 설명서의 AWS IAM으로 작업하는 서비스를 참조하세요.

Amazon EBS의 자격 증명 기반 정책

ID 기반 정책 지원: 예

ID 기반 정책은 IAM 사용자, 사용자 그룹 또는 역할과 같은 ID에 연결할 수 있는 JSON 권한 정책 문서입니다. 이러한 정책은 사용자 및 역할이 어떤 리소스와 어떤 조건에서 어떤 작업을 수행할 수 있는지를 제어합니다. 자격 증명 기반 정책을 생성하는 방법을 알아보려면 IAM 사용 설명서에서 고객 관리형 정책으로 사용자 지정 IAM 권한 정의를 참조하세요.

IAM ID 기반 정책을 사용하면 허용되거나 거부되는 작업과 리소스뿐 아니라 작업이 허용되거나 거부되는 조건을 지정할 수 있습니다. JSON 정책에서 사용할 수 있는 모든 요소에 대해 알아보려면 IAM 사용 설명서의 IAM JSON 정책 요소 참조를 참조하세요.

Amazon EBS의 자격 증명 기반 정책 예시

Amazon EBS 자격 증명 기반 정책 예시를 보려면 Amazon EBS에 대한 IAM 정책 예제 섹션을 참조하세요.

Amazon EBS 내의 리소스 기반 정책

리소스 기반 정책 지원: 아니요

리소스 기반 정책은 리소스에 연결하는 JSON 정책 설명서입니다. 리소스 기반 정책의 예제는 IAM 역할 신뢰 정책과 Amazon S3 버킷 정책입니다. 리소스 기반 정책을 지원하는 서비스에서 서비스 관리자는 이러한 정책을 사용하여 특정 리소스에 대한 액세스를 통제할 수 있습니다. 정책이 연결된 리소스의 경우 정책은 지정된 보안 주체가 해당 리소스와 어떤 조건에서 어떤 작업을 수행할 수 있는지를 정의합니다. 리소스 기반 정책에서 보안 주체를 지정해야 합니다. 보안 주체에는 계정, 사용자, 역할, 페더레이션 사용자 또는이 포함될 수 있습니다 AWS 서비스.

교차 계정 액세스를 활성화하려는 경우, 전체 계정이나 다른 계정의 IAM 개체를 리소스 기반 정책의 보안 주체로 지정할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 IAM에서 교차 계정 리소스 액세스를 참조하세요.

Amazon EBS의 정책 작업

정책 작업 지원: 예

관리자는 AWS JSON 정책을 사용하여 누가 무엇에 액세스할 수 있는지 지정할 수 있습니다. 즉, 어떤 보안 주체가 어떤 리소스와 어떤 조건에서 작업을 수행할 수 있는지를 지정할 수 있습니다.

JSON 정책의 `Action`요소는 정책에서 액세스를 허용하거나 거부하는 데 사용할 수 있는 작업을 설명합니다. 연결된 작업을 수행할 수 있는 권한을 부여하기 위한 정책에 작업을 포함하세요.

EBS 작업의 목록을 보려면 서비스 승인 참조의 Amazon EC2를 위한 작업, 리소스 및 조건 키 및 Amazon EBS를 위한 작업, 리소스 및 조건 키를 참조하세요.

Amazon EBS의 정책 작업은 작업 앞에 `ec2` 또는 `ebs` 접두사를 사용합니다.

단일 문에서 여러 작업을 지정하려면 쉼표로 구분합니다.

```
"Action": [
  "ec2:action1",
  "ec2:action2"
]
```

Amazon EBS 자격 증명 기반 정책 예시를 보려면 Amazon EBS에 대한 IAM 정책 예제 섹션을 참조하세요.

Amazon EBS의 정책 리소스

정책 리소스 지원: 예

관리자는 AWS JSON 정책을 사용하여 누가 무엇에 액세스할 수 있는지 지정할 수 있습니다. 즉, 어떤 보안 주체가 어떤 리소스와 어떤 조건에서 작업을 수행할 수 있는지를 지정할 수 있습니다.

`Resource` JSON 정책 요소는 작업이 적용되는 하나 이상의 객체를 지정합니다. 모범 사례에 따라 Amazon 리소스 이름(ARN)을 사용하여 리소스를 지정합니다. 리소스 수준 권한을 지원하지 않는 작업의 경우, 와일드카드(*)를 사용하여 해당 문이 모든 리소스에 적용됨을 나타냅니다.

```
"Resource": "*"
```

일부 Amazon EBS API 작업에서는 여러 리소스가 지원됩니다. 단일 문에서 여러 리소스를 지정하려면 ARN을 쉼표로 구분합니다. 예를 들면 `DescribeVolumes`에서는 vol-01234567890abcdef와 vol-09876543210fedcba에 액세스하므로 두 리소스에 모두 액세스하는 권한이 보안 주체에게 있어야 합니다.

```
"Resource": [
  "arn:aws:ec2:us-east-1:123456789012:volume/vol-01234567890abcdef",
  "arn:aws:ec2:us-east-1:123456789012:volume/vol-09876543210fedcba"
]
```

Amazon EBS의 정책 조건 키

서비스별 정책 조건 키 지원: 예

관리자는 AWS JSON 정책을 사용하여 누가 무엇에 액세스할 수 있는지 지정할 수 있습니다. 즉, 어떤 보안 주체가 어떤 리소스와 어떤 조건에서 작업을 수행할 수 있는지를 지정할 수 있습니다.

`Condition` 요소는 정의된 기준에 따라 문이 실행되는 시기를 지정합니다. 같음(equals) 또는 미만(less than)과 같은 조건 연산자를 사용하여 정책의 조건을 요청의 값과 일치시키는 조건식을 생성할 수 있습니다. 모든 AWS 전역 조건 키를 보려면 IAM 사용 설명서의 AWS 전역 조건 컨텍스트 키를 참조하세요.

예를 들면 다음 조건에서는 볼륨 유형이 gp2인 경우에만 보안 주체가 볼륨에 대한 작업을 수행할 수 있습니다.

```
"Condition":{
  "StringLikeIfExists":{
    "ec2:VolumeType":"gp2"
  }
}
```

Amazon EBS 조건 키의 목록을 보려면 서비스 승인 참조의 Actions, resources, and condition keys를 참조하세요.

Amazon EBS의 ACL

ACL 지원: 아니요

액세스 제어 목록(ACL)은 어떤 보안 주체(계정 멤버, 사용자 또는 역할)가 리소스에 액세스할 수 있는 권한을 가지고 있는지를 제어합니다. ACL은 JSON 정책 문서 형식을 사용하지 않지만 리소스 기반 정책과 유사합니다.

Amazon EBS 포함 ABAC

ABAC 지원(정책의 태그): 부분적

속성 기반 액세스 제어(ABAC)는 태그라고 불리는 속성을 기반으로 권한을 정의하는 권한 부여 전략입니다. IAM 엔터티 및 AWS 리소스에 태그를 연결한 다음 보안 주체의 태그가 리소스의 태그와 일치할 때 작업을 허용하는 ABAC 정책을 설계할 수 있습니다.

태그에 근거하여 액세스를 제어하려면 `aws:ResourceTag/key-name`, `aws:RequestTag/key-name`또는 `aws:TagKeys`조건 키를 사용하여 정책의 조건 요소에 태그 정보를 제공합니다.

서비스가 모든 리소스 유형에 대해 세 가지 조건 키를 모두 지원하는 경우, 값은 서비스에 대해 예입니다. 서비스가 일부 리소스 유형에 대해서만 세 가지 조건 키를 모두 지원하는 경우, 값은 부분적입니다.

ABAC에 대한 자세한 내용은 IAM 사용 설명서의 ABAC 권한 부여를 통한 권한 정의를 참조하세요. ABAC 설정 단계가 포함된 자습서를 보려면 IAM 사용 설명서의 속성 기반 액세스 제어(ABAC) 사용을 참조하세요.

Amazon EBS에서 임시 보안 인증 사용

임시 자격 증명 지원: 예

임시 자격 증명은 AWS 리소스에 대한 단기 액세스를 제공하며 페더레이션 또는 전환 역할을 사용할 때 자동으로 생성됩니다. 장기 액세스 키를 사용하는 대신 임시 자격 증명을 동적으로 생성하는 것이 AWS 좋습니다. 자세한 내용은 IAM 사용 설명서의 IAM의 임시 보안 자격 증명 및 IAM으로 작업하는 AWS 서비스 섹션을 참조하세요.

Amazon EBS에 대한 교차 서비스 보안 주체 권한

전달 액세스 세션(FAS) 지원: 예

전달 액세스 세션(FAS)은를 호출하는 보안 주체의 권한을 다운스트림 서비스에 AWS 서비스 대한 요청과 AWS 서비스함께 사용합니다. FAS 요청 시 정책 세부 정보는 전달 액세스 세션을 참조하세요.

Amazon EBS의 서비스 역할

서비스 역할 지원: 예

서비스 역할은 서비스가 사용자를 대신하여 작업을 수행하는 것으로 가정하는 IAM 역할입니다. IAM 관리자는 IAM 내에서 서비스 역할을 생성, 수정 및 삭제할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 AWS 서비스 AWS에 권한을 위임할 역할 생성을 참조하세요.

> **Warning**
>
> 서비스 역할에 대한 권한을 변경하면 Amazon EBS 기능이 중단될 수 있습니다. Amazon EBS에서 관련 지침을 제공하는 경우에만 서비스 역할을 편집합니다.

Amazon EBS의 서비스 연결 역할

서비스 연결 역할 지원: 아니요

서비스 연결 역할은에 연결된 서비스 역할의 한 유형입니다 AWS 서비스. 서비스는 사용자를 대신하여 태스크를 수행하기 위해 역할을 수임할 수 있습니다. 서비스 연결 역할은에 나타나 AWS 계정 며 서비스가 소유합니다. IAM 관리자는 서비스 연결 역할의 권한을 볼 수 있지만 편집은 할 수 없습니다.

서비스 연결 역할 생성 또는 관리에 대한 자세한 내용은 IAM으로 작업하는AWS 서비스를 참조하세요. 서비스 연결 역할 열에서 `Yes`가 포함된 서비스를 테이블에서 찾습니다. 해당 서비스에 대한 서비스 연결 역할 설명서를 보려면 예(Yes) 링크를 선택합니다.

Amazon EBS에 대한 IAM 정책 예제

기본적으로 사용자 및 역할은 Amazon EBS 리소스를 생성하거나 수정할 수 있는 권한이 없습니다. 또한 AWS Management Console, AWS Command Line Interface (AWS CLI) 또는 AWS API를 사용하여 작업을 수행할 수 없습니다. 사용자에게 사용자가 필요한 리소스에서 작업을 수행할 권한을 부여하려면 IAM 관리자가 IAM 정책을 생성하면 됩니다. 그런 다음 관리자가 IAM 정책을 역할에 추가하고, 사용자가 역할을 맡을 수 있습니다.

이러한 예제 JSON 정책 문서를 사용하여 IAM ID 기반 정책을 생성하는 방법을 알아보려면 IAM 사용 설명서의 IAM 정책 생성을 참조하세요.

주제

- 정책 모범 사례
- 사용자가 Amazon EBS 콘솔을 사용하도록 허용
- 사용자가 자신의 고유한 권한을 볼 수 있도록 허용
- 사용자가 볼륨 작업을 수행할 수 있도록 허용
- 사용자가 스냅샷 작업을 수행할 수 있도록 허용

정책 모범 사례

자격 증명 기반 정책에 따라 해당 계정에서 누가 Amazon EBS 리소스를 생성, 액세스 또는 삭제할 수 있는지 결정됩니다. 이 작업으로 인해 AWS 계정에 비용이 발생할 수 있습니다. ID 기반 정책을 생성하거나 편집할 때는 다음 지침과 권장 사항을 따르세요.

- AWS 관리형 정책을 시작하고 최소 권한으로 전환 - 사용자 및 워크로드에 권한 부여를 시작하려면 많은 일반적인 사용 사례에 대한 권한을 부여하는 AWS 관리형 정책을 사용합니다. 에서 사용할 수 있습니다 AWS 계정. 사용 사례에 맞는 AWS 고객 관리형 정책을 정의하여 권한을 추가로 줄이는 것이 좋습니다. 자세한 내용은 IAM 사용 설명서의 AWS 관리형 정책 또는 AWS 직무에 대한 관리형 정책을 참조하세요.
- 최소 권한 적용 – IAM 정책을 사용하여 권한을 설정하는 경우, 작업을 수행하는 데 필요한 권한만 부여합니다. 이렇게 하려면 최소 권한으로 알려진 특정 조건에서 특정 리소스에 대해 수행할 수 있는 작업을 정의합니다. IAM을 사용하여 권한을 적용하는 방법에 대한 자세한 정보는 IAM 사용 설명서에 있는 IAM의 정책 및 권한을 참조하세요.
- IAM 정책의 조건을 사용하여 액세스 추가 제한 – 정책에 조건을 추가하여 작업 및 리소스에 대한 액세스를 제한할 수 있습니다. 예를 들어, SSL을 사용하여 모든 요청을 전송해야 한다고 지정하는 정책 조건을 작성할 수 있습니다. AWS 서비스와 같은 특정를 통해 사용되는 경우 조건을 사용하여 서비스 작업에 대한 액세스 권한을 부여할 수도 있습니다 CloudFormation. 자세한 내용은 IAM 사용 설명서의 IAM JSON 정책 요소: 조건을 참조하세요.
- IAM Access Analyzer를 통해 IAM 정책을 확인하여 안전하고 기능적인 권한 보장 - IAM Access Analyzer에서는 IAM 정책 언어(JSON)와 모범 사례가 정책에서 준수되도록 새로운 및 기존 정책을 확인합니다. IAM Access Analyzer는 100개 이상의 정책 확인 항목과 실행 가능한 추천을 제공하여 안전하고 기능적인 정책을 작성하도록 돕습니다. 자세한 내용은 IAM 사용 설명서의 IAM Access Analyzer에서 정책 검증을 참조하세요.
- 다중 인증(MFA) 필요 -에서 IAM 사용자 또는 루트 사용자가 필요한 시나리오가 있는 경우 추가 보안을 위해 MFA를 AWS 계정켭니다. API 작업을 직접적으로 호출할 때 MFA가 필요하면 정책에 MFA 조건을 추가합니다. 자세한 내용은 IAM 사용 설명서의 MFA를 통한 보안 API 액세스를 참조하세요.

IAM의 모범 사례에 대한 자세한 내용은 IAM 사용 설명서의 IAM의 보안 모범 사례를 참조하세요.

사용자가 Amazon EBS 콘솔을 사용하도록 허용

Amazon Elastic Block Store 콘솔에 액세스하려면 최소 권한 세트가 있어야 합니다. 이러한 권한은에서 Amazon EBS 리소스에 대한 세부 정보를 나열하고 볼 수 있도록 허용해야 합니다 AWS 계정. 최소 필수 권한보다 더 제한적인 ID 기반 정책을 생성하는 경우, 콘솔이 해당 정책에 연결된 엔티티(사용자 또는 역할)에 대해 의도대로 작동하지 않습니다.

AWS CLI 또는 AWS API만 호출하는 사용자에게는 최소 콘솔 권한을 허용할 필요가 없습니다. 대신, 수행하려는 API 작업과 일치하는 작업에만 액세스할 수 있도록 합니다.

사용자와 역할이 Amazon EBS 콘솔을 계속 사용할 수 있도록 하려면 Amazon EBS *ConsoleAccess* 또는 *ReadOnly* AWS 관리형 정책도 엔터티에 연결합니다. 자세한 내용은 IAM 사용 설명서의 사용자에게 권한 추가를 참조하세요.

사용자가 자신의 고유한 권한을 볼 수 있도록 허용

이 예제는 IAM 사용자가 자신의 사용자 ID에 연결된 인라인 및 관리형 정책을 볼 수 있도록 허용하는 정책을 생성하는 방법을 보여줍니다. 이 정책에는 콘솔에서 또는 AWS CLI 또는 AWS API를 사용하여 프로그래밍 방식으로이 작업을 완료할 수 있는 권한이 포함됩니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
```

```
    ]
}
```

사용자가 볼륨 작업을 수행할 수 있도록 허용

예제:

- 예: 볼륨 연결 및 분리
- 예: 볼륨 생성
- 예: 태그를 사용하여 볼륨 생성
- 예: Amazon EC2 콘솔을 사용하여 볼륨과 연동

예: 볼륨 연결 및 분리

API 작업의 호출자가 여러 리소스를 지정해야 하는 경우 사용자가 필요한 모든 리소스에 액세스하도록 허용하는 정책 명령문을 생성해야 합니다. 이러한 리소스가 하나 이상 포함된 `Condition` 요소를 사용해야 하는 경우 이 예제와 같이 여러 명령문을 생성해야 합니다.

다음 정책은 "`volume_user=iam-user-name`" 태그가 있는 볼륨을 "`department=dev`" 태그가 있는 인스턴스에 연결하고 해당 볼륨을 해당 인스턴스에서 분리하도록 허용합니다. IAM 그룹에 이 정책을 연결하면 `aws:username` 정책 변수가 그룹의 각 사용자에게 자신의 사용자 이름을 값으로 하는 `volume_user`라는 태그가 있는 인스턴스에 볼륨을 연결하거나 분리할 권한을 부여합니다.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:AttachVolume",
                "ec2:DetachVolume"
            ],
            "Resource": "arn:aws:ec2:us-east-1:111122223333:instance/*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/department": "dev"
                }
```

```
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:AttachVolume",
                "ec2:DetachVolume"
            ],
            "Resource": "arn:aws:ec2:us-east-1:111122223333:volume/*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/volume_user": "${aws:username}"
                }
            }
        }
    ]
}
```

예: 볼륨 생성

다음 정책은 사용자가 CreateVolume API 작업을 사용하는 것을 허용합니다. 사용자는 볼륨이 암호화되고 볼륨 크기가 20GB 미만인 경우에만 볼륨을 생성할 수 있습니다.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateVolume"
            ],
            "Resource": "arn:aws:ec2:us-east-1:111122223333:volume/*",
            "Condition": {
                "NumericLessThan": {
                    "ec2:VolumeSize": "20"
                },
                "Bool": {
                    "ec2:Encrypted": "true"
                }
            }
```

```
        }
    ]
}
```

예: 태그를 사용하여 볼륨 생성

다음 정책에는 사용자가 태그 aws:RequestTag 및 costcenter=115를 사용하여 생성하는 볼륨에 태그를 지정해야 하는 stack=prod 조건 키가 포함됩니다. 사용자가 이 특정 키들을 전달하지 않거나 태그를 전혀 지정하지 않으면 요청은 실패합니다.

태그를 적용하는 리소스 생성 작업의 경우, 사용자가 CreateTags 작업을 사용할 권한도 가지고 있어야 합니다. 두 번째 문은 ec2:CreateAction 조건 키를 사용하여 사용자가 CreateVolume의 컨텍스트에서만 태그를 생성하도록 허용합니다. 사용자는 기존의 볼륨이나 다른 어떤 리소스에도 태그를 지정할 수 없습니다.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowCreateTaggedVolumes",
            "Effect": "Allow",
            "Action": "ec2:CreateVolume",
            "Resource": "arn:aws:ec2:us-east-1:111122223333:volume/*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/costcenter": "115",
                    "aws:RequestTag/stack": "prod"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateTags"
            ],
            "Resource": "arn:aws:ec2:us-east-1:111122223333:volume/*",
            "Condition": {
                "StringEquals": {
                    "ec2:CreateAction": "CreateVolume"
```

```
                }
            }
        }
    ]
}
```

다음 정책은 사용자가 태그를 지정하지 않고 볼륨을 생성하는 것을 허용합니다. CreateTags 작업은 CreateVolume 요청에서 태그가 지정되는 경우에만 평가됩니다. 사용자가 태그를 지정하는 경우, 태그는 purpose=test여야 합니다. 다른 어떤 태그도 요청에서 허용되지 않습니다.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ec2:CreateVolume",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateTags"
            ],
            "Resource": "arn:aws:ec2:us-east-1:111122223333:volume/*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/purpose": "test",
                    "ec2:CreateAction": "CreateVolume"
                },
                "ForAllValues:StringEquals": {
                    "aws:TagKeys": "purpose"
                }
            }
        }
    ]
}
```

예: Amazon EC2 콘솔을 사용하여 볼륨과 연동

다음 정책에서는 볼륨을 보고 생성하며, Amazon EC2 콘솔을 사용하여 특정 인스턴스에 볼륨을 연결 및 분리하는 권한을 사용자에게 부여합니다.

사용자는 "purpose=test" 태그가 있는 인스턴스에 볼륨을 연결하고 해당 인스턴스에서 볼륨을 분리할 수 있습니다. Amazon EC2 콘솔을 사용하여 볼륨을 연결하려는 경우 사용자에게 `ec2:DescribeInstances` 작업 사용 권한을 부여하는 것이 좋습니다. 이렇게 하면 볼륨 연결 대화 상자의 미리 구성된 목록에서 인스턴스를 선택할 수 있습니다. 그러나 이렇게 하면 사용자가 콘솔의 인스턴스 페이지에서 모든 인스턴스를 조회할 수 있으므로 이 작업을 생략할 수 있습니다.

첫 번째 문에서 작업은 사용자가 볼륨을 생성할 때 가용 영역을 선택할 수 있도록 하는 데 `ec2:DescribeAvailabilityZones` 필요합니다.

사용자는 볼륨 생성 중 또는 생성 후 본인이 작성한 볼륨에 태그를 지정할 수 없습니다.

사용자가 스냅샷 작업을 수행할 수 있도록 허용

다음은 `CreateSnapshot`(EBS 볼륨의 특정 시점 스냅샷)과 `CreateSnapshots`(다중 볼륨 스냅샷)에 대한 정책 예제입니다.

예제:

- 예: 스냅샷 생성
- 예: 스냅샷 생성
- 예: 태그를 사용하여 스냅샷 생성
- 예: 태그를 사용하여 볼륨 생성
- 예: 스냅샷 복사
- 예: 스냅샷 권한 설정 수정

예: 스냅샷 생성

다음 정책은 고객이 CreateSnapshot API 작업을 사용하는 것을 허용합니다. 고객은 볼륨이 암호화되고 볼륨 크기가 20GiB 미만인 경우에만 스냅샷을 생성할 수 있습니다.

JSON

```
{
    "Version":"2012-10-17",
```

```
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ec2:CreateSnapshot",
            "Resource": "arn:aws:ec2:us-east-1::snapshot/*"
        },
        {
            "Effect": "Allow",
            "Action": "ec2:CreateSnapshot",
            "Resource": "arn:aws:ec2:us-east-1:111122223333:volume/*",
            "Condition": {
                "NumericLessThan": {
                    "ec2:VolumeSize": "20"
                },
                "Bool": {
                    "ec2:Encrypted": "true"
                }
            }
        }
    ]
}
```

예: 스냅샷 생성

다음 정책은 고객이 CreateSnapshots API 작업을 사용하는 것을 허용합니다. 고객은 인스턴스의 모든 볼륨이 GP2 유형인 경우에만 스냅샷을 생성할 수 있습니다.

JSON

```
{
   "Version":"2012-10-17",
   "Statement": [
      {
         "Effect":"Allow",
         "Action":"ec2:CreateSnapshots",
         "Resource":[
"arn:aws:ec2:us-east-1::snapshot/*",
"arn:aws:ec2:*:*:instance/*"
   ]
      },
      {
         "Effect":"Allow",
```

```
         "Action":"ec2:CreateSnapshots",
         "Resource":"arn:aws:ec2:us-east-1:*:volume/*",
         "Condition":{
            "StringLikeIfExists":{
               "ec2:VolumeType":"gp2"
             }
     }

      }
   ]
}
```

예: 태그를 사용하여 스냅샷 생성

다음 정책에는 고객이 태그 aws:RequestTag 및 costcenter=115를 모든 새로운 스냅샷에 적용해야 하는 stack=prod 조건 키가 포함됩니다. 사용자가 이 특정 키들을 전달하지 않거나 태그를 전혀 지정하지 않으면 요청은 실패합니다.

태그를 적용하는 리소스 생성 작업의 경우, 고객이 CreateTags 작업을 사용할 권한도 가지고 있어야 합니다. 세 번째 문은 ec2:CreateAction 조건 키를 사용하여 고객이 CreateSnapshot의 컨텍스트에서만 태그를 생성하도록 허용합니다. 고객은 기존의 볼륨이나 다른 어떤 리소스에도 태그를 지정할 수 없습니다.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ec2:CreateSnapshot",
            "Resource": "arn:aws:ec2:us-east-1:111122223333:volume/*"
        },
        {
            "Sid": "AllowCreateTaggedSnapshots",
            "Effect": "Allow",
            "Action": "ec2:CreateSnapshot",
            "Resource": "arn:aws:ec2:us-east-1::snapshot/*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/costcenter": "115",
```

```
                    "aws:RequestTag/stack": "prod"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "ec2:CreateTags",
            "Resource": "arn:aws:ec2:us-east-1::snapshot/*",
            "Condition": {
                "StringEquals": {
                    "ec2:CreateAction": "CreateSnapshot"
                }
            }
        }
    ]
}
```

예: 태그를 사용하여 볼륨 생성

다음 정책에는 다중 볼륨 스냅샷 세트를 생성할 때 고객이 태그 aws:RequestTag 및 costcenter=115를 모든 새로운 스냅샷에 적용해야 하는 stack=prod 조건 키가 포함됩니다. 사용자가 이 특정 키들을 전달하지 않거나 태그를 전혀 지정하지 않으면 요청은 실패합니다.

JSON

```
{
   "Version":"2012-10-17",
   "Statement": [
      {
         "Effect":"Allow",
         "Action":"ec2:CreateSnapshots",
         "Resource":[
"arn:aws:ec2:us-east-1::snapshot/*",
"arn:aws:ec2:*:*:instance/*",
"arn:aws:ec2:*:*:volume/*"

   ]
      },
      {
         "Sid":"AllowCreateTaggedSnapshots",
         "Effect":"Allow",
```

```
         "Action":"ec2:CreateSnapshots",
         "Resource":"arn:aws:ec2:us-east-1::snapshot/*",
         "Condition":{
            "StringEquals":{
               "aws:RequestTag/costcenter":"115",
               "aws:RequestTag/stack":"prod"
            }
         }
      },
      {
         "Effect":"Allow",
         "Action":"ec2:CreateTags",
         "Resource":"arn:aws:ec2:us-east-1::snapshot/*",
         "Condition":{
            "StringEquals":{
               "ec2:CreateAction":"CreateSnapshots"
            }
         }
      }
   ]
}
```

다음 정책은 고객이 태그를 지정하지 않고 스냅샷을 생성하는 것을 허용합니다. CreateTags 작업은 CreateSnapshot 또는 CreateSnapshots 요청에서 태그가 지정되는 경우에만 평가됩니다. 요청에서 태그를 생략할 수 있습니다. 태그가 지정된 경우 태그는 purpose=test여야 합니다. 다른 어떤 태그도 요청에서 허용되지 않습니다.

JSON

```
{
   "Version":"2012-10-17",
   "Statement": [
      {
         "Effect":"Allow",
         "Action":"ec2:CreateSnapshot",
         "Resource":"*"
      },
      {
         "Effect":"Allow",
         "Action":"ec2:CreateTags",
         "Resource":"arn:aws:ec2:us-east-1::snapshot/*",
```

```
         "Condition":{
            "StringEquals":{
               "aws:RequestTag/purpose":"test",
               "ec2:CreateAction":"CreateSnapshot"
            },
            "ForAllValues:StringEquals":{
               "aws:TagKeys":"purpose"
            }
         }
      }
   ]
}
```

다음 정책은 고객이 태그를 지정하지 않고 다중 볼륨 스냅샷 세트를 생성하는 것을 허용합니다. CreateTags 작업은 CreateSnapshot 또는 CreateSnapshots 요청에서 태그가 지정되는 경우에만 평가됩니다. 요청에서 태그를 생략할 수 있습니다. 태그가 지정된 경우 태그는 purpose=test여야 합니다. 다른 어떤 태그도 요청에서 허용되지 않습니다.

JSON

```
{
   "Version":"2012-10-17",
   "Statement": [
      {
         "Effect":"Allow",
         "Action":"ec2:CreateSnapshots",
         "Resource":"*"
      },
      {
         "Effect":"Allow",
         "Action":"ec2:CreateTags",
         "Resource":"arn:aws:ec2:us-east-1::snapshot/*",
         "Condition":{
            "StringEquals":{
               "aws:RequestTag/purpose":"test",
               "ec2:CreateAction":"CreateSnapshots"
            },
            "ForAllValues:StringEquals":{
               "aws:TagKeys":"purpose"
            }
         }
```

```
      }
   ]
}
```

다음 정책은 소스 볼륨이 고객에 대해 `User:username`으로 태그 지정된 경우, 그리고 스냅샷 자체가 `Environment:Dev` 및 `User:username`으로 태그 지정된 경우에만 스냅샷 생성을 허용합니다. 고객은 스냅샷에 추가 태그를 추가할 수 있습니다.

`CreateSnapshots`에 대한 다음 정책은 고객에 대해 `User:username`으로 태그 지정된 경우, 그리고 스냅샷 자체가 `Environment:Dev` 및 `User:username`으로 태그 지정된 경우에만 스냅샷 생성을 허용합니다.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ec2:CreateSnapshots",
            "Resource": "arn:aws:ec2:us-east-1:*:instance/*"
        },
        {
            "Effect": "Allow",
            "Action": "ec2:CreateSnapshots",
            "Resource": "arn:aws:ec2:us-east-1:111122223333:volume/*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/User": "${aws:username}"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "ec2:CreateSnapshots",
            "Resource": "arn:aws:ec2:us-east-1::snapshot/*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/Environment": "Dev",
                    "aws:RequestTag/User": "${aws:username}"
                }
```

```
            }
        },
        {
            "Effect": "Allow",
            "Action": "ec2:CreateTags",
            "Resource": "arn:aws:ec2:us-east-1::snapshot/*"
        }
    ]
}
```

다음 정책은 스냅샷이 고객에 대해 User:username으로 태그 지정된 경우에만 스냅샷 삭제를 허용합니다.

JSON

```
{
   "Version":"2012-10-17",
   "Statement": [
            {
         "Effect":"Allow",
         "Action":"ec2:DeleteSnapshot",
         "Resource":"arn:aws:ec2:us-east-1::snapshot/*",
         "Condition":{
            "StringEquals":{
               "aws:ResourceTag/User":"${aws:username}"
            }
         }
      }
   ]
}
```

다음 정책은 고객의 스냅샷 생성을 허용하지만 생성되는 스냅샷이 태그 키 `value=stack`을 보유한 경우에는 작업을 거부합니다.

JSON

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

```
      {
         "Effect":"Allow",
         "Action":[
            "ec2:CreateSnapshot",
            "ec2:CreateTags"
         ],
         "Resource":"*"
      },
      {
         "Effect":"Deny",
         "Action":"ec2:CreateSnapshot",
         "Resource":"arn:aws:ec2:us-east-1::snapshot/*",
         "Condition":{
            "ForAnyValue:StringEquals":{
               "aws:TagKeys":"stack"
            }
         }
      }
   ]
}
```

다음 정책은 고객의 스냅샷 생성을 허용하지만 생성되는 스냅샷이 태그 키 `value=stack`을 보유한 경우에는 작업을 거부합니다.

JSON

```
{
   "Version":"2012-10-17",
   "Statement": [
      {
         "Effect":"Allow",
         "Action":[
            "ec2:CreateSnapshots",
            "ec2:CreateTags"
         ],
         "Resource":"*"
      },
      {
         "Effect":"Deny",
         "Action":"ec2:CreateSnapshots",
         "Resource":"arn:aws:ec2:us-east-1::snapshot/*",
```

```
         "Condition":{
            "ForAnyValue:StringEquals":{
               "aws:TagKeys":"stack"
            }
         }
      }
   ]
}
```

다음 정책은 여러 작업을 정책 하나로 결합하도록 허용합니다. 스냅샷이 us-east-1 리전에서 생성된 경우에만 스냅샷을 생성할 수 있습니다(CreateSnapshots의 컨텍스트에서). 스냅샷이 CreateSnapshots 리전에서 생성되고 인스턴스 유형이 us-east-1인 경우에만 (t2*의 컨텍스트에서) 스냅샷을 생성할 수 있습니다.

JSON

```
{
   "Version":"2012-10-17",
   "Statement": [
          {
         "Effect":"Allow",
         "Action":[
            "ec2:CreateSnapshots",
            "ec2:CreateSnapshot",
            "ec2:CreateTags"
         ],
         "Resource": [
            "arn:aws:ec2:*:*:instance/*",
            "arn:aws:ec2:*:*:snapshot/*",
            "arn:aws:ec2:*:*:volume/*"
         ],
         "Condition":{
            "StringEqualsIgnoreCase": {
              "ec2:Region": "us-east-1"
            },
            "StringLikeIfExists": {
              "ec2:InstanceType": ["t2.*"]
            }
         }
      }
   ]
```

```
}
```

예: 스냅샷 복사

CopySnapshot 작업에 대해 지정된 리소스 수준 권한은 새 스냅샷과 소스 스냅샷 모두에 적용됩니다.

다음 예제 정책은 태그 키 purpose 및 태그 값 production(purpose=production)을 사용하여 새 스냅샷이 생성된 경우에만 보안 주체의 스냅샷 복사를 허용합니다.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowCopySnapshotWithTags",
            "Effect": "Allow",
            "Action": "ec2:CopySnapshot",
            "Resource": "arn:aws:ec2:*:111122223333:snapshot/*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/purpose": "production"
                }
            }
        }
    ]
}
```

다음 예제 정책은 AWS 계정에서 소스 스냅샷을 소유한 경우에만 보안 주체가 스냅샷을 복사하도록 허용합니다123456789012.

다음 예제 정책은 태그 키 CSISnapshotName을 사용하여 새 스냅샷이 생성된 경우에만 위탁자가 스냅샷을 복사하도록 허용합니다.

```
{
    "Effect": "Allow",
    "Action": "ec2:CopySnapshot",
    "Resource": "arn:aws:ec2:*::snapshot/${*}",
    "Condition": {
        "StringLike": {
```

```
            "aws:RequestTag/CSISnapshotName": "*"
        }
     }
},

{
    "Effect": "Allow",
    "Action": "ec2:CopySnapshot",
    "Resource": "arn:aws:ec2:*::snapshot/snap-*"
}
```

예: 스냅샷 권한 설정 수정

다음 정책은 스냅샷에 태그가 지정된 경우에만 스냅샷 수정을 허용합니다. `User:username`여기서 *username*은 고객의 AWS 계정 사용자 이름입니다. 이 조건이 충족되지 않으면 요청은 실패합니다.

JSON

```
{
   "Version":"2012-10-17",
   "Statement": [
           {
         "Effect":"Allow",
         "Action":"ec2:ModifySnapshotAttribute",
         "Resource":"arn:aws:ec2:us-east-1::snapshot/*",
         "Condition":{
            "StringEquals":{
               "aws:ResourceTag/user-name":"${aws:username}"
            }
         }
      }
   ]
}
```

Amazon EBS 권한 부여 문제 해결

다음 정보를 사용하여 Amazon EBS 및 IAM으로 작업할 때 발생할 수 있는 일반적인 문제를 진단하고 수정할 수 있습니다.

문제

- Amazon EBS에서 작업을 수행할 권한이 없음
- iam:PassRole을 수행하도록 인증되지 않음
- 내 외부의 사람이 내 Amazon EBS 리소스 AWS 계정 에 액세스하도록 허용하고 싶습니다.

Amazon EBS에서 작업을 수행할 권한이 없음

에서 작업을 수행할 권한이 없다는 AWS Management Console 메시지가 표시되면 관리자에게 문의하여 지원을 받아야 합니다. 관리자는 로그인 보안 인증 정보를 제공한 사람입니다.

다음 예시 오류는 `mateojackson` IAM 사용자가 콘솔을 사용하여 볼륨에 대한 세부 정보를 보려고 하지만 `ec2:DescribeVolumes` 권한이 없는 경우에 발생합니다.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 ec2:DescribeVolumes on resource: volume-id
```

이 경우 Mateo는 AWS 관리자에게 볼륨을 설명하도록 허용해 달라고 요청합니다.

iam:PassRole을 수행하도록 인증되지 않음

`iam:PassRole` 작업을 수행할 수 있는 권한이 없다는 오류가 수신되면 Amazon EBS에 역할을 전달할 수 있도록 정책을 업데이트해야 합니다.

일부 AWS 서비스 에서는 새 서비스 역할 또는 서비스 연결 역할을 생성하는 대신 기존 역할을 해당 서비스에 전달할 수 있습니다. 이렇게 하려면 역할을 서비스에 전달할 권한이 있어야 합니다.

다음 예시 오류는 이름이 `marymajor`인 IAM 사용자가 콘솔을 사용하여 Amazon EBS에서 작업을 수행하려고 하는 경우에 발생합니다. 하지만 작업을 수행하려면 서비스 역할이 부여한 권한이 서비스에 있어야 합니다. Mary는 서비스에 역할을 전달할 권한이 없습니다.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

이 경우, Mary가 `iam:PassRole`작업을 수행할 수 있도록 Mary의 정책을 업데이트해야 합니다.

도움이 필요한 경우 AWS 관리자에게 문의하세요. 관리자는 로그인 자격 증명을 제공한 사람입니다.

내 외부의 사람이 내 Amazon EBS 리소스 AWS 계정 에 액세스하도록 허용하고 싶습니다.

다른 계정의 사용자 또는 조직 외부의 사람이 리소스에 액세스할 때 사용할 수 있는 역할을 생성할 수 있습니다. 역할을 수임할 신뢰할 수 있는 사람을 지정할 수 있습니다. 리소스 기반 정책 또는 액세스 제어 목록(ACL)을 지원하는 서비스의 경우, 이러한 정책을 사용하여 다른 사람에게 리소스에 대한 액세스 권한을 부여할 수 있습니다.

자세한 내용은 다음을 참조하세요.

- Amazon EBS에서 이러한 기능의 지원 여부는 Amazon EBS에서 IAM을 사용하는 방법 섹션을 참조하세요.
- 소유 AWS 계정 한의 리소스에 대한 액세스 권한을 제공하는 방법을 알아보려면 IAM 사용 설명서의 소유한 다른의 IAM 사용자에게 액세스 권한 제공을 참조 AWS 계정 하세요.
- 타사에 리소스에 대한 액세스 권한을 제공하는 방법을 알아보려면 IAM 사용 설명서의 타사가 AWS 계정 소유한에 대한 액세스 권한 제공을 AWS 계정참조하세요.
- ID 페더레이션을 통해 액세스 권한을 제공하는 방법을 알아보려면 IAM 사용 설명서의 외부에서 인증된 사용자에게 액세스 권한 제공(ID 페더레이션)을 참조하세요.
- 크로스 계정 액세스에 대한 역할과 리소스 기반 정책 사용의 차이점을 알아보려면 IAM 사용 설명서의 IAM의 크로스 계정 리소스 액세스를 참조하세요.

Amazon EBS에 대한 규정 준수 확인

새로운 AWS 가입 환경은 규제 워크로드용으로 설계되지 않았습니다. 새로운 AWS 가입 환경을 사용하고 있지만 규제 대상 워크로드 AWS 에를 사용하려는 경우 AWS 환경에 가입 AWS (고급)하거나 고급 기능을 활성화할 수 있습니다.

AWS 서비스 가 특정 규정 준수 프로그램의 범위 내에 있는지 알아보려면 AWS 서비스 규정 준수 프로그램 범위 내 참조하고 관심 있는 규정 준수 프로그램을 선택합니다. 일반 정보는 AWS 규정 준수 프로그램.

를 사용하여 타사 감사 보고서를 다운로드할 수 있습니다 AWS Artifact. 자세한 내용은 Downloading Reports inDownloading AWS Artifact을 참조하세요.

사용 시 규정 준수 책임은 데이터의 민감도, 회사의 규정 준수 목표 및 관련 법률과 규정에 따라 AWS 서비스 결정됩니다. 사용 시 규정 준수 책임에 대한 자세한 내용은 AWS 보안 설명서를 AWS 서비스참조하세요.

Amazon EBS의 데이터 복원력

AWS 글로벌 인프라는 AWS 리전 및 가용 영역을 중심으로 구축됩니다.는 물리적으로 분리되고 격리된 여러 가용 영역을 AWS 리전 제공하며,이 가용 영역은 지연 시간이 짧고 처리량이 높으며 중복성이 높은 네트워킹과 연결됩니다. 가용 영역을 사용하면 중단 없이 영역 간에 자동으로 장애 극복 조치가 이루어지는 애플리케이션 및 데이터베이스를 설계하고 운영할 수 있습니다. 가용 영역은 기존의 단일 또는 다중 데이터 센터 인프라보다 가용성, 내결함성, 확장성이 뛰어납니다.

AWS 리전 및 가용 영역에 대한 자세한 내용은 AWS 글로벌 인프라를 참조하세요.

AWS 글로벌 인프라 외에도 Amazon EBS는 데이터 복원력 및 백업 요구 사항을 지원하는 몇 가지 기능을 제공합니다.

- Amazon Data Lifecycle Manager를 사용하여 EBS 스냅샷 자동화
- 리전 간 EBS 스냅샷 복사

Amazon EBS 모니터링 도구

모니터링은 Amazon Elastic Block Store 및 기타 AWS 솔루션의 안정성, 가용성 및 성능을 유지하는 데 중요한 부분입니다.는 Amazon EBS를 모니터링하고, 이상이 있을 때 보고하고, 필요한 경우 자동 조치를 취할 수 있도록 다음 모니터링 도구를 AWS 제공합니다.

- AWS CloudTrail는에 의해 또는를 대신하여 수행된 API 호출 및 관련 이벤트를 캡처 AWS 계정 하고 사용자가 지정한 Amazon S3 버킷에 로그 파일을 전송합니다. 호출한 사용자 및 계정 AWS, 호출이 수행된 소스 IP 주소, 호출이 발생한 시기를 식별할 수 있습니다. EBS 볼륨 및 스냅샷을 관리하기 위한 API는 Amazon EC2 API의 일부입니다. CloudTrail 및 Amazon EC2 API에 대한 자세한 내용은 Amazon EC2 사용 설명서의 AWS CloudTrail을 사용하여 Amazon EC2 API 직접 호출 로깅을 참조하세요.
- Amazon CloudWatch는 AWS 리소스와 AWS 에서 실행하는 애플리케이션을 실시간으로 모니터링합니다. 지표를 수집 및 추적하고, 사용자 지정 대시보드를 생성할 수 있으며, 지정된 지표가 지정한 임곗값에 도달하면 사용자에게 알리거나 조치를 취하도록 경보를 설정할 수 있습니다. 예를 들어 CloudWatch에서 Amazon EC2 인스턴스의 CPU 사용량 또는 기타 지표를 추적하고 필요할 때 자동으로 새 인스턴스를 시작할 수 있습니다. 자세한 내용은 the section called “Amazon CloudWatch” 단원을 참조하십시오.
- Amazon EventBridge를 사용하여 AWS 서비스를 자동화하고 애플리케이션 가용성 문제 또는 리소스 변경과 같은 시스템 이벤트에 자동으로 대응할 수 있습니다. AWS 서비스의 이벤트는 거의 실시간으로 EventBridge로 전달됩니다. 원하는 이벤트만 표시하도록 간단한 규칙을 작성한 후 규칙과 일치하는 이벤트 발생 시 실행할 자동화 작업을 지정할 수 있습니다. 자세한 내용은 the section called “Amazon EventBridge” 단원을 참조하십시오.
- Amazon EBS 세부 성능 통계는 Nitro 기반 Amazon EC2 인스턴스에 연결된 Amazon EBS 볼륨에 대한 실시간 I/O 성능 통계를 제공합니다. 자세한 설명은 Amazon EBS 세부 성능 통계 섹션을 참조하세요.
- Amazon GuardDuty는 EC2 인스턴스에서 잠재적으로 악의적인 활동을 탐지하는 데 도움이 됩니다. EC2용 GuardDuty 맬웨어 보호는 EC2 인스턴스에 연결된 EBS 볼륨을 스캔합니다. 자세한 내용은 the section called “Amazon GuardDuty” 단원을 참조하십시오.

Amazon EBS에 대한 Amazon CloudWatch 지표

Amazon CloudWatch 지표는 볼륨의 작동 동작을 살펴보고, 분석하고, 경보를 설정하는 데 사용할 수 있는 통계 데이터입니다.

1분간의 무료 데이터가 자동으로 제공됩니다.

CloudWatch에서 데이터를 가져올 때 반환되는 데이터의 세부 수준을 지정하는 Period 요청 파라미터를 포함할 수 있습니다. 이는 EBS가 데이터를 수집하는 데 사용하는 기간(1분 기간)과 다릅니다. 반환된 데이터가 유효한지 확인하려면 요청에 수집 기간보다 크거나 같은 기간을 지정하는 것이 좋습니다.

CloudWatch API 또는 Amazon EC2 콘솔을 사용하여 데이터를 가져올 수 있습니다. 이 콘솔은 CloudWatch API에서 원시 데이터를 가져오고 데이터를 기반으로 일련의 그래프를 표시합니다. 필요에 따라 API의 데이터나 콘솔의 그래프를 사용할 수 있습니다.

주제

- Amazon EBS 볼륨 지표
- Amazon EBS 스냅샷에 대한 지표
- Nitro 인스턴스 관련 지표
- 빠른 스냅샷 복원 관련 지표
- Amazon EC2 콘솔 그래프

Amazon EBS 볼륨 지표

AWS/EBS 네임스페이스에는 모든 인스턴스 유형에 연결된 EBS 볼륨에 대한 다음 지표가 포함됩니다. 모든 Amazon EBS 볼륨 유형은 볼륨이 인스턴스에 연결된 경우에만 1분 지표를 CloudWatch에 자동으로 전송합니다.

인스턴스의 운영 체제에서 사용 가능한 디스크 공간에 대한 자세한 내용은 여유 디스크 공간 보기 섹션을 참조하세요.

> **Note**
>
> Nitro System 기반 인스턴스에서는 지표 중 일부가 다릅니다. 이러한 인스턴스 유형의 목록은 Nitro System에 구축된 인스턴스를 참조하세요.

지표	설명	단위	측정 기준	의미 있는 통계
VolumeAvgIOPS *	**Note** Nitro 인스턴스에 연결된 모든 EBS 볼륨 유형에 지원됩니다. Amazon ECS 및 AWS Fargate 작업에 연결된 볼륨에 대해서는 게시되지 않습니다. 1분 동안 볼륨에서 실행되는 평균 읽기 및 쓰기 IOPS. 지난 1분 이내에 볼륨에서 실행된 작업이 없는 경우 지표 값은 0입니다. 자세한 내용은 CloudWatch를 사용하여 I/O 특성 모니터링 단원을 참조하십시오. 다중 연결 지원 볼륨의 경우 InstanceId 차원을 사용하여 특정 볼륨-인스턴스 연결의 평균 IOPS를 확인합니다.	Ops/s	VolumeId | InstanceId	• Sum • Average • Minimum | Maximum
VolumeAvgThroughput *	**Note** Nitro 인스턴스에 연결된 모든	KiB/s	VolumeId | InstanceId	• Sum • Average

지표	설명	단위	측정 기준	의미 있는 통계
	EBS 볼륨 유형에 지원됩니다. Amazon ECS 및 AWS Fargate 작업에 연결된 볼륨에 대해서는 게시되지 않습니다. 1분 동안 볼륨에서 실행되는 평균 읽기 및 쓰기 처리량. 지난 1분 이내에 볼륨에서 실행된 작업이 없는 경우 지표 값은 0입니다. 자세한 내용은 CloudWatch를 사용하여 I/O 특성 모니터링 단원을 참조하십시오. 다중 연결 지원 볼륨의 경우 InstanceId 차원을 사용하여 특정 볼륨-인스턴스 연결의 평균 처리량을 확인합니다.			• Minimum | Maximum

지표	설명	단위	측정 기준	의미 있는 통계
VolumeAvgReadLatency *	**Note** Nitro 인스턴스에 연결된 모든 볼륨 유형에 지원됩니다. Amazon ECS 및 AWS Fargate 작업에 연결된 볼륨에 대해서는 게시되지 않습니다. 1분 동안 읽기 작업을 완료하는 데 걸리는 평균 시간입니다. 이 지표를 사용하여 Amazon EC2 인스턴스에 연결된 EBS 볼륨의 평균 I/O 지연 시간을 모니터링합니다. 평균은 지난 1분 동안 완료된 I/O 작업을 기반으로 계산됩니다. 지난 1분 이내에 완료된 작업이 없는 경우 지표 값은 0입니다. 다중 연결 지원 볼륨의 경우 InstanceId 차원을 사용하여 특정 볼륨-인스턴스 연결의 평균 지연 시간을 확인합니다.	밀리초	VolumeId | InstanceId	Minimum | Maximum

지표	설명	단위	측정 기준	의미 있는 통계
VolumeAvgWriteLatency *	**Note** Nitro 인스턴스에 연결된 모든 볼륨 유형에 지원됩니다. Amazon ECS 및 AWS Fargate 작업에 연결된 볼륨에 대해서는 게시되지 않습니다. 1분 동안 쓰기 작업을 완료하는 데 걸리는 평균 시간입니다. 이 지표를 사용하여 Amazon EC2 인스턴스에 연결된 EBS 볼륨의 평균 I/O 지연 시간을 모니터링합니다. 평균은 지난 1분 동안 완료된 I/O 작업을 기반으로 계산됩니다. 지난 1분 이내에 완료된 작업이 없는 경우 지표 값은 0입니다. 다중 연결 지원 볼륨의 경우 InstanceId 차원을 사용하여 특정 볼륨-인스턴스 연결의 평균 지연 시간을 확인합니다.	밀리초	VolumeId | InstanceId	Minimum | Maximum

지표	설명	단위	측정 기준	의미 있는 통계
VolumeIOPSExceededCheck *	**Note** Nitro 인스턴스에 연결된, 마그네틱(standard)을 제외한 모든 볼륨 유형에 지원됩니다. 다중 연결 지원 볼륨에서 지원되지 않습니다. Amazon ECS 및 AWS Fargate 작업에 연결된 볼륨에 대해서는 게시되지 않습니다. 애플리케이션이 지난 1분 이내에 볼륨의 프로비저닝된 IOPS 성능을 초과하는 IOPS를 지속적으로 구동하려고 했는지 여부를 보고합니다. 이 지표는 0(프로비저닝된 IOPS를 초과하지 않음) 또는 1(프로비저닝된 IOPS를 초과함)일 수 있습니다. 자세한 내용은 CloudWatch를 사용하여 I/O 특성 모니터링 단원을 참조하십시오.	없음	VolumeId | InstanceId	• Sum • Average • Minimum | Maximum

지표	설명	단위	측정 기준	의미 있는 통계
VolumeThroughputExceededCheck *	**Note** Nitro 인스턴스에 연결된, 마그네틱(standard)을 제외한 모든 볼륨 유형에 지원됩니다. 다중 연결 지원 볼륨에서 지원되지 않습니다. Amazon ECS 및 AWS Fargate 작업에 연결된 볼륨에 대해서는 게시되지 않습니다. 애플리케이션이 지난 1분 이내에 볼륨의 프로비저닝된 처리량 성능을 초과하는 처리량을 지속적으로 구동하려고 했는지 여부를 보고합니다. 이 지표는 0(프로비저닝된 처리량을 초과하지 않음) 또는 1(프로비저닝된 처리량을 초과함)일 수 있습니다. 자세한 내용은 CloudWatch를 사용하여 I/O 특성 모니터링 섹션을 참조하세요.	없음	VolumeId | InstanceId	• Sum • Average • Minimum | Maximum

지표	설명	단위	측정 기준	의미 있는 통계
VolumeReadBytes	지정된 기간의 읽기 작업에 대한 정보를 제공합니다. • Sum 통계는 해당 기간 동안 전송된 총 바이트 수를 보고합니다. • Average 통계는 지정된 기간 동안 읽는 평균 바이트 수를 보고합니다. • SampleCount 통계는 통계 계산에 사용된 데이터 포인트의 수를 나타냅니다. Note Xen 인스턴스의 경우 볼륨에서 읽기 작업이 있을 때에만 데이터가 보고됩니다.	바이트	VolumeId	• Average • Sum • SampleCount • Minimum | Maximum - Nitro 기반 인스턴스에 연결된 볼륨만 해당

지표	설명	단위	측정 기준	의미 있는 통계
VolumeWriteBytes	지정된 기간의 쓰기 작업에 대한 정보를 제공함 • Sum 통계는 해당 기간 동안 전송된 총 바이트 수를 보고합니다. • Average 통계는 지정된 기간 동안 쓴 평균 바이트 수를 보고합니다. • SampleCount 통계는 통계 계산에 사용된 데이터 포인트의 수를 나타냅니다. Note Xen 인스턴스의 경우 볼륨에서 쓰기 작업이 있을 때에만 데이터가 보고됩니다.	바이트	VolumeId	• Average • Sum • SampleCount • Minimum | Maximum - Nitro 기반 인스턴스에 연결된 볼륨만 해당

지표	설명	단위	측정 기준	의미 있는 통계
VolumeReadOps	지정된 기간의 총 읽기 작업 수입니다. 읽기 작업은 완료 시 계산됩니다. 해당 기간의 초당 평균 읽기 작업 수(읽기 IOPS)를 계산하려면 해당 기간의 총 읽기 작업 수를 해당 기간의 초 수로 나누세요.	개수	VolumeId	• Average • Sum • Minimum | Maximum - Nitro 기반 인스턴스에 연결된 볼륨만 해당
VolumeWriteOps	지정된 기간의 총 쓰기 작업 수입니다. 쓰기 작업은 완료 시 계산됩니다. 해당 기간의 초당 평균 쓰기 작업 수(쓰기 IOPS)를 계산하려면 해당 기간의 총 쓰기 작업 수를 해당 기간의 초 수로 나누세요.	개수	VolumeId	• Average • Sum • Minimum | Maximum - Nitro 기반 인스턴스에 연결된 볼륨만 해당

지표	설명	단위	측정 기준	의미 있는 통계
VolumeTotalReadTime	**Note** 다중 연결 지원 볼륨에서 지원되지 않습니다. Xen 인스턴스의 경우 볼륨에서 읽기 작업이 있을 때에만 데이터가 보고됩니다. 지정된 기간 동안 완료된 모든 읽기 작업에서 사용한 총 시간(초)입니다. 여러 요청이 동시에 제출된 경우 이 총계가 기간 길이보다 클 수 있습니다. 예를 들어, 1분(60초) 동안 150개의 작업이 완료되고 작업당 1초가 걸린 경우 값은 150초입니다.	초	VolumeId	• Average - Nitro 기반 인스턴스에 연결된 볼륨과 무관함 • Sum • Minimum | Maximum - Nitro 기반 인스턴스에 연결된 볼륨만 해당

지표	설명	단위	측정 기준	의미 있는 통계
VolumeTotalWriteTime	**Note** 다중 연결 지원 볼륨에서 지원되지 않습니다. Xen 인스턴스의 경우 볼륨에서 쓰기 작업이 있을 때에만 데이터가 보고됩니다. 지정된 기간 동안 완료된 모든 쓰기 작업에서 사용한 총 시간(초)입니다. 여러 요청이 동시에 제출된 경우 이 총계가 기간 길이보다 클 수 있습니다. 예를 들어, 1분(60초) 동안 150개의 작업이 완료되고 작업당 1초가 걸린 경우 값은 150초입니다.	초	VolumeId	• Average - Nitro 기반 인스턴스에 연결된 볼륨과 무관함 • Sum • Minimum | Maximum - Nitro 기반 인스턴스에 연결된 볼륨만 해당

지표	설명	단위	측정 기준	의미 있는 통계
VolumeIdleTime	**Note** 다중 연결 지원 볼륨에서 지원되지 않습니다. 지정된 기간 동안 읽기 또는 쓰기 작업이 제출되지 않은 총 시간(초)입니다.	초	VolumeId	• Average - Nitro 기반 인스턴스에 연결된 볼륨과 무관함 • Sum • Minimum | Maximum - Nitro 기반 인스턴스에 연결된 볼륨만 해당

지표	설명	단위	측정 기준	의미 있는 통계
VolumeQueueLength	지정된 기간 동안 완료 대기 중인 읽기 및 쓰기 작업 요청 수입니다.	개수	VolumeId	• Average • Sum - Nitro 인스턴스에 연결된 볼륨과 무관함 • Minimum | Maximum - Nitro 인스턴스에 연결된 볼륨만 해당

지표	설명	단위	측정 기준	의미 있는 통계
VolumeStalledIOCheck *	**Note** Nitro 인스턴스에만 해당됩니다. Amazon ECS 및 AWS Fargate 작업에 연결된 볼륨에 대해서는 게시되지 않았습니다. 볼륨이 마지막 순간에 멈춘 IO 검사를 통과 또는 실패했는지 보고합니다. 이 지표는 0(통과) 또는 1(실패)일 수 있습니다. 자세한 내용은 CloudWatch를 사용하여 I/O 특성 모니터링 단원을 참조하십시오.	없음	VolumeId | InstanceId	• Sum • 평균 • 최소 • 최대

지표	설명	단위	측정 기준	의미 있는 통계
VolumeThroughputPercentage	**Note** 프로비저닝된 IOPS SSD 볼륨만 해당됩니다. 다중 연결 지원 볼륨에서 지원되지 않습니다. Amazon EBS 볼륨에 대해 프로비저닝된 총 IOPS 중 전송된 IOPS(초당 I/O 작업 수)의 비율(%)입니다. 프로비저닝된 IOPS SSD 볼륨은 99.9%의 시간 동안 프로비저닝된 성능을 제공합니다. 쓰기 중 1분 동안 보류 중인 다른 I/O 요청이 없으면 지표 값이 100%가 됩니다. 또한 사용자가 취한 조치(예: 사용량 피크 시 볼륨 스냅샷 생성, EBS에 최적화되지 않은 인스턴스에서 볼륨 실행, 볼륨 데이터에 최초로 액세스 등)로 인해 볼륨의 I/O 성능이 일시적으로 저하될 수 있습니다.	%	VolumeId	• Average • Minimum | Maximum

지표	설명	단위	측정 기준	의미 있는 통계
VolumeConsumedReadWriteOps	**Note** 프로비저닝된 IOPS SSD 볼륨만 해당됩니다. 지정된 시간 동안 소비한 총 읽기 및 쓰기 작업량(256,000 용량 단위로 정규화됨)입니다. 256,000보다 작은 I/O 작업은 각각 1개의 소비 IOPS로 계산되고, 256,000보다 큰 I/O 작업은 256,000 용량 단위로 계산됩니다. 예를 들어, 1,024K I/O는 소비 IOPS 4개로 계산됩니다.	개수	VolumeId	• Average • Sum • Minimum | Maximum

지표	설명	단위	측정 기준	의미 있는 통계
BurstBalance	**Note** gp2, st1 및 sc1 볼륨에만 해당됩니다. 버스트 버킷에 남아 있는 I/O 크레딧(gp2의 경우)의 비율 또는 처리량 크레딧의 비율(st1 및 sc1의 경우) 정보를 보여 줍니다. 볼륨이 활성 상태일 때만 CloudWatch에 데이터가 보고되고, 볼륨이 연결되지 않은 경우에는 데이터가 보고되지 않습니다. 볼륨의 기준 성능이 최대 버스트 성능을 초과하면 크레딧은 사용되지 않습니다. 다른 인스턴스의 경우 보고되는 버스트 밸런스는 100%입니다. 자세한 내용은 gp2 볼륨 성능 단원을 참조하십시오.	%	VolumeId	• Average • Sum - Nitro 인스턴스에 연결된 볼륨과는 무관합니다. • Minimum | Maximum

* Outposts, Local Zones 또는 Wavelength Zones에서는 이러한 지표를 사용할 수 없습니다.

Amazon EBS 스냅샷에 대한 지표

AWS/EBS 네임스페이스에는 Amazon EBS 스냅샷에 대한 다음 지표가 포함되어 있습니다.

지표	설명	단위	측정 기준	의미 있는 통계
SnapshotCopyBytesTransferred	AWS 리전에 복사된 스냅샷 데이터의 양.	바이트	sourceRegion	Sum

Nitro 인스턴스 관련 지표

AWS/EC2 네임스페이스에는 베어 메탈 인스턴스가 아닌 Nitro 기반 인스턴스에 연결된 볼륨에 대한 추가 Amazon EBS 지표가 포함됩니다.

지표	설명	단위	의미 있는 통계
InstanceEBSIOPSExceededCheck *	애플리케이션이 지난 1분 이내에 인스턴스의 최대 EBS IOPS 제한을 초과하는 IOPS를 시도했는지 보고합니다. 이 지표는 0(IOPS를 초과하지 않음) 또는 1(IOPS를 초과함)일 수 있습니다.	없음	• Sum • 평균 • 최소 • 최대
InstanceEBSThroughputExceededCheck *	애플리케이션이 지난 1분 이내에 인스턴스의 최대 EBS 처리량 제한을 초과하는 처리량을 시도했는지 보고합니다. 이 지표는 0(처리량 초과 안 함) 또는 1(처리량 초과함)일 수 있습니다.	없음	• Sum • 평균 • 최소 • 최대
EBSReadOps	지정된 기간 내에 인스턴스에 연결된 모든 Amazon EBS 볼륨에서 완료된 읽기 작업입니다. 해당 기간의 초당 평균 읽기 I/O 작업 수(읽기 IOPS)를 계산하려면 해당 기간의 총 작업 수를 해당 기간의 초 수로 나누세요. 기본(5분) 모니터링을 사용하는 경우, 이 숫자를 300으로 나누어 읽기 IOPS를 계산할 수 있습니다. 세부(1분) 모니터링의 경우 60으로 나눕니다. CloudWatch 지표 수학 함수 DIFF_TIME 을 사용하여 초	개수	• 합계 • 평균 • 최소 • 최대

지표	설명	단위	의미 있는 통계
	당 작업 수를 찾을 수도 있습니다. 예를 들어 CloudWatch에서 EBSReadOps 을 m1로 그래프로 표시한 경우 지표 수학 공식 m1/(DIFF_TIME(m1)) 은 지표(작업/초)를 반환합니다. DIFF_TIME 및 기타 지표 수학 함수에 대한 자세한 내용은 Amazon CloudWatch 사용 설명서의 지표 수학 사용을 참조하세요.		
EBSWriteOps	지정된 기간 내에 인스턴스에 연결된 모든 EBS 볼륨으로의 완료된 쓰기 작업입니다. 해당 기간의 초당 평균 쓰기 I/O 작업 수(쓰기 IOPS)를 계산하려면 해당 기간의 총 작업 수를 해당 기간의 초 수로 나누세요. 기본(5분) 모니터링을 사용하는 경우, 이 숫자를 300으로 나누어 쓰기 IOPS를 계산할 수 있습니다. 세부(1분) 모니터링의 경우 60으로 나눕니다. CloudWatch 지표 수학 함수 DIFF_TIME 을 사용하여 초당 작업 수를 찾을 수도 있습니다. 예를 들어 CloudWatch에서 EBSWriteOps 을 m1로 그래프로 표시한 경우 지표 수학 공식 m1/(DIFF_TIME(m1)) 은 지표(작업/초)를 반환합니다. DIFF_TIME 및 기타 지표 수학 함수에 대한 자세한 내용은 Amazon CloudWatch 사용 설명서의 지표 수학 사용을 참조하세요.	개수	• 합계 • 평균 • 최소 • 최대

지표	설명	단위	의미 있는 통계
EBSReadBytes	지정된 기간 내에 인스턴스에 연결된 모든 EBS 볼륨에서의 바이트 읽기 작업입니다. 보고된 숫자는 해당 기간에 읽은 바이트 수입니다. 기본(5분) 모니터링을 사용하는 경우, 이 숫자를 300으로 나누어 읽기 바이트/초를 찾을 수 있습니다. 세부(1분) 모니터링의 경우 60으로 나눕니다. CloudWatch 지표 수학 함수 DIFF_TIME 을 사용하여 초당 바이트 수를 찾을 수도 있습니다. 예를 들어 CloudWatch에서 EBSReadBytes 을 m1로 그래프로 표시한 경우 지표 수학 공식 m1/(DIFF_TIME(m1)) 은 지표(바이트/초)를 반환합니다. DIFF_TIME 및 기타 지표 수학 함수에 대한 자세한 내용은 Amazon CloudWatch 사용 설명서의 Use metric math를 참조하세요.	바이트	• Sum • 평균 • 최소 • 최대
EBSWriteBytes	지정된 기간 내에 인스턴스에 연결된 모든 EBS 볼륨으로의 바이트 쓰기 작업입니다. 보고된 숫자는 해당 기간에 써진 바이트 수입니다. 기본(5분) 모니터링을 사용하는 경우, 이 숫자를 300으로 나누어 쓰기 바이트/초를 찾을 수 있습니다. 세부(1분) 모니터링의 경우 60으로 나눕니다. CloudWatch 지표 수학 함수 DIFF_TIME 을 사용하여 초당 바이트 수를 찾을 수도 있습니다. 예를 들어 CloudWatch에서 EBSWriteBytes 을 m1로 그래프로 표시한 경우 지표 수학 공식 m1/(DIFF_TIME(m1)) 은 지표(바이트/초)를 반환합니다. DIFF_TIME 및 기타 지표 수학 함수에 대한 자세한 내용은 Amazon CloudWatch 사용 설명서의 Use metric math를 참조하세요.	바이트	• Sum • 평균 • 최소 • 최대

지표	설명	단위	의미 있는 통계
EBSIOBalance%	버스트 버킷에 남아 있는 I/O 크레딧의 비율에 대한 정보를 제공합니다. 기본 모니터링에서만 이 지표를 사용할 수 있습니다. 이 지표는 24시간에 한 번 이상 30분 동안만 최대 성능을 발휘하는 일부 *.4xlarge 인스턴스 크기 이하에서만 사용할 수 있습니다. 자세한 내용은 기본적으로 EBS에 최적화됨 섹션을 참조하세요. Sum 통계는 이 지표에 적용할 수 없습니다.	%	• 최소 • 최대
EBSByteBalance%	버스트 버킷에 남아 있는 처리량 크레딧의 비율에 대한 정보를 제공합니다. 기본 모니터링에서만 이 지표를 사용할 수 있습니다. 이 지표는 24시간에 한 번 이상 30분 동안만 최대 성능을 발휘하는 일부 *.4xlarge 인스턴스 크기 이하에서만 사용할 수 있습니다. 자세한 내용은 기본적으로 EBS에 최적화됨 섹션을 참조하세요. Sum 통계는 이 지표에 적용할 수 없습니다.	%	• 최소 • 최대

* Outposts, Local Zones 또는 Wavelength Zones에서는 이러한 지표를 사용할 수 없습니다.

빠른 스냅샷 복원 관련 지표

AWS/EBS 네임스페이스에는 빠른 스냅샷 복원에 대한 다음 지표가 포함되어 있습니다.

측정치	설명	단위	측정 기준	의미 있는 통계
FastSnapshotRestoreCreditsBucketSize	누적될 수 있는 최대 볼륨 생성 크레딧 수. 이 지표는 가용 영역당 스냅샷별로 보고됩니다.	없음	SnapshotId | AvailabilityZone	• Average • Minimum | Maximum

측정치	설명	단위	측정 기준	의미 있는 통계
				Note 가장 유용한 통계는 `Average`입니다. `Minimum` 및 `Maximum` 통계의 결과는 `Average`의 통계와 동일하며 대신 사용될 수 있습니다.
`FastSnapshotRestoreCreditsBalance`	사용 가능한 볼륨 생성 크레딧 수. 이 지표는 가용 영역당 스냅샷별로 보고됩니다.	없음	`SnapshotId` | `AvailabilityZone`	• `Average` • `Minimum` | `Maximum` **Note** 가장 유용한 통계는 `Average`입니다. `Minimum` 및 `Maximum` 통계의 결과는 `Average`의 통계와 동일하며 대신 사용될 수 있습니다.

Amazon EC2 콘솔 그래프

볼륨을 생성한 후 Amazon EC2 콘솔로 가서 볼륨의 모니터링 그래프를 볼 수 있습니다. 콘솔의 볼륨 페이지에서 볼륨을 선택하고 모니터링을 선택합니다. 다음 표에는 표시되는 그래프가 나열되어 있습니다. 오른쪽 열에는 CloudWatch API의 원시 데이터 측정치로 각 그래프가 생성되는 방법이 설명되어 있습니다. 모든 그래프의 기간은 5분입니다.

그래프	원시 지표를 사용하여 설명
읽기 처리량(KiB/s)	`Sum(VolumeReadBytes) / Period / 1024`
쓰기 처리량(KiB/s)	`Sum(VolumeWriteBytes) / Period / 1024`
읽기 작업(Ops/s)	`Sum(VolumeReadOps) / Period`
쓰기 작업(Ops/s)	`Sum(VolumeWriteOps) / Period`
평균 대기열 길이(작업)	`Avg(VolumeQueueLength)`
유휴 시간(%)	`Sum(VolumeIdleTime) / Period × 100`

평균 읽기 크기(KiB/op)	`Avg(VolumeReadBytes) / 1024`
	
	Nitro 기반 인스턴스의 경우, 다음 공식에서 CloudWatch 지표 수식을 사용하여 평균 읽기 크기를 도출합니다.
	
	`(Sum(VolumeReadBytes) / Sum(Volum eReadOps)) / 1024`
	
	`VolumeReadBytes` 및 `VolumeReadOps` 지표는 EBS CloudWatch 콘솔에서 사용할 수 있습니다.

평균 쓰기 크기(KiB/op)	`Avg(VolumeWriteBytes) / 1024`
	
	Nitro 기반 인스턴스의 경우, 다음 공식에서 CloudWatch 지표 수식을 사용하여 평균 쓰기 크기를 도출합니다.
	
	`(Sum(VolumeWriteBytes) / Sum(Volum eWriteOps)) / 1024`
	
	`VolumeWriteBytes` 및 `VolumeWriteOps` 지표는 EBS CloudWatch 콘솔에서 사용할 수 있습니다.

평균 읽기 지연 시간(ms/op)	`Avg(VolumeTotalReadTime) × 1000` Nitro 기반 인스턴스의 경우, 다음 공식에서 CloudWatch 지표 수식을 사용하여 평균 읽기 지연 시간을 도출합니다.

그래프	원시 지표를 사용하여 설명
	`(Sum(VolumeTotalReadTime) / Sum(VolumeReadOps)) × 1000` `VolumeTotalReadTime` 및 `VolumeReadOps` 지표는 EBS CloudWatch 콘솔에서 사용할 수 있습니다.

평균 쓰기 지연 시간(ms/op)	`Avg(VolumeTotalWriteTime) × 1000`
	
	Nitro 기반 인스턴스의 경우, 다음 공식에서 CloudWatch 지표 수식을 사용하여 평균 쓰기 지연 시간을 도출합니다.
	
	`(Sum(VolumeTotalWriteTime) / Sum(VolumeWriteOps)) * 1000`
	
	`VolumeTotalWriteTime` 및 `VolumeWriteOps` 지표는 EBS CloudWatch 콘솔에서 사용할 수 있습니다.

평균 지연 시간 그래프 및 평균 크기 그래프의 경우 기간 중 완료된 총 작업(그래프에 해당하는 읽기 또는 쓰기) 수를 기준으로 평균을 계산합니다.

Amazon EBS용 Amazon EventBridge 이벤트

Amazon EBS는 볼륨 및 스냅샷에 대해 수행된 작업의 이벤트를 Amazon EventBridge로 전송합니다. EventBridge에서는 이러한 이벤트에 대한 응답으로 프로그래밍 작업을 트리거하는 규칙을 설정할 수 있습니다. 예를 들어 스냅샷의 빠른 스냅샷 복원이 활성화된 경우 이메일로 알림을 보내는 규칙을 만들 수 있습니다.

EventBridge의 이벤트는 JSON 객체로 표현됩니다. 이 이벤트에 고유한 필드는 JSON 객체의 "세부 정보" 섹션에 포함되어 있습니다. "이벤트" 필드에는 이벤트 이름이 포함됩니다. "결과" 필드에는 이 벤트를 트리거한 작업의 완료 상태가 포함됩니다. 자세한 내용은 Amazon EventBridge 사용 설명서의 Amazon EventBridge 이벤트 패턴을 참조하세요.

자세한 내용은 Amazon EventBridge User Guide(Amazon EventBridge 사용 설명서)의 What Is Amazon EventBridge?(Amazon EventBridge란?)를 참조하세요.

이벤트

- EBS 볼륨 이벤트

- EBS 볼륨 초기화 이벤트
- EBS 볼륨 수정 이벤트
- EBS 스냅샷 이벤트
- EBS 스냅샷 아카이브 이벤트
- EBS 빠른 스냅샷 복원 이벤트
- AWS Lambda 를 사용하여 EventBridge 이벤트 처리

EBS 볼륨 이벤트

Amazon EBS는 다음과 같은 볼륨 이벤트가 발생하는 경우 EventBridge에 이벤트를 보냅니다.

이벤트

- 볼륨 생성(createVolume)
- 볼륨 삭제(deleteVolume)
- 볼륨 연결 또는 다시 연결(attachVolume, reattachVolume)
- 볼륨 분리(detachVolume)

볼륨 생성(createVolume)

볼륨 생성 작업이 완료되면 `createVolume` 이벤트가 AWS 계정으로 전송됩니다. 그러나 저장, 로깅, 아카이빙은 되지 않습니다. 이 이벤트에서 `available` 또는 `failed` 결과가 발생할 수 있습니다. 아래 예제와 같이 유효하지 않은가 AWS KMS key 제공되면 생성이 실패합니다.

이벤트 데이터

아래 목록에 성공적인 `createVolume` 이벤트에 대해 EBS가 발생시키는 JSON 객체의 예를 열거했습니다.

```
{
   "version": "0",
   "id": "01234567-0123-0123-0123-012345678901",
   "detail-type": "EBS Volume Notification",
   "source": "aws.ec2",
   "account": "012345678901",
   "time": "yyyy-mm-ddThh:mm:ssZ",
```

```
   "region": "us-east-1",
   "resources": [
      "arn:aws:ec2:us-east-1:012345678901:volume/vol-01234567"
   ],
   "detail": {
      "result": "available",
      "cause": "",
      "event": "createVolume",
      "request-id": "01234567-0123-0123-0123-0123456789ab"
   }
}
```

아래 목록에 createVolume 이벤트 실패 이후 EBS가 발생시키는 JSON 객체의 예를 열거했습니다. 실패의 원인은 비활성화된 KMS 키였습니다.

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-0123456789ab",
  "detail-type": "EBS Volume Notification",
  "source": "aws.ec2",
  "account": "012345678901",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "region": "sa-east-1",
  "resources": [
    "arn:aws:ec2:sa-east-1:0123456789ab:volume/vol-01234567",
  ],
  "detail": {
    "event": "createVolume",
    "result": "failed",
    "cause": "arn:aws:kms:sa-
east-1:0123456789ab:key/01234567-0123-0123-0123-0123456789ab is disabled.",
    "request-id": "01234567-0123-0123-0123-0123456789ab",
  }
}
```

아래에 createVolume 이벤트 실패 이후 EBS가 발생시키는 JSON 객체의 예를 제시했습니다. 실패의 원인은 보류 중인 KMS 키 가져오기였습니다.

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-0123456789ab",
  "detail-type": "EBS Volume Notification",
```

```
  "source": "aws.ec2",
  "account": "012345678901",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "region": "sa-east-1",
  "resources": [
    "arn:aws:ec2:sa-east-1:0123456789ab:volume/vol-01234567",
  ],
  "detail": {
    "event": "createVolume",
    "result": "failed",
    "cause": "arn:aws:kms:sa-
east-1:0123456789ab:key/01234567-0123-0123-0123-0123456789ab is pending import.",
    "request-id": "01234567-0123-0123-0123-0123456789ab",
  }
}
```

볼륨 삭제(deleteVolume)

볼륨 삭제 작업이 완료되면 deleteVolume 이벤트가 AWS 계정으로 전송됩니다. 그러나 저장, 로깅, 아카이빙은 되지 않습니다. 이 이벤트에 deleted 결과가 있습니다. 삭제가 완료되지 않으면 이벤트가 전송되지 않습니다.

이벤트 데이터

아래 목록에 성공적인 deleteVolume 이벤트에 대해 EBS가 발생시키는 JSON 객체의 예를 열거했습니다.

```
{
   "version": "0",
   "id": "01234567-0123-0123-0123-012345678901",
   "detail-type": "EBS Volume Notification",
   "source": "aws.ec2",
   "account": "012345678901",
   "time": "yyyy-mm-ddThh:mm:ssZ",
   "region": "us-east-1",
   "resources": [
      "arn:aws:ec2:us-east-1:012345678901:volume/vol-01234567"
   ],
   "detail": {
      "result": "deleted",
      "cause": "",
      "event": "deleteVolume",
      "request-id": "01234567-0123-0123-0123-0123456789ab"
```

```
    }
}
```

볼륨 연결 또는 다시 연결(attachVolume, reattachVolume)

볼륨이 인스턴스에 연결되거나 재연결될 때 `attachVolume` 또는 `reattachVolume` 이벤트가 AWS 계정으로 전송됩니다. 그러나 저장, 로깅, 아카이빙은 되지 않습니다. KMS 키를 사용하여 EBS 볼륨을 암호화할 경우 KMS 키이(가) 무효해지면, EBS는 해당 KMS 키이(가) 나중에 인스턴스에 연결하거나 다시 연결하는 데 사용될 경우 이벤트를 발송합니다(아래 예 참조).

이벤트 데이터

아래 목록에 `attachVolume` 이벤트 실패 이후 EBS가 발생시키는 JSON 객체의 예를 열거했습니다. 실패의 원인은 보류 중인 KMS 키 삭제였습니다.

> **Note**
>
> AWS 는 정기 서버 유지 관리 후 볼륨에 재연결을 시도할 수 있습니다.

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-0123456789ab",
  "detail-type": "EBS Volume Notification",
  "source": "aws.ec2",
  "account": "012345678901",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "region": "us-east-1",
  "resources": [
  "arn:aws:ec2:us-east-1:0123456789ab:volume/vol-01234567",
  "arn:aws:kms:us-east-1:0123456789ab:key/01234567-0123-0123-0123-0123456789ab"
  ],
  "detail": {
    "event": "attachVolume",
    "result": "failed",
    "cause": "arn:aws:kms:us-
east-1:0123456789ab:key/01234567-0123-0123-0123-0123456789ab is pending deletion.",
    "request-id": ""
  }
}
```

아래 목록에 reattachVolume 이벤트 실패 이후 EBS가 발생시키는 JSON 객체의 예를 열거했습니다. 실패의 원인은 보류 중인 KMS 키 삭제였습니다.

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-0123456789ab",
  "detail-type": "EBS Volume Notification",
  "source": "aws.ec2",
  "account": "012345678901",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "region": "us-east-1",
  "resources": [
  "arn:aws:ec2:us-east-1:0123456789ab:volume/vol-01234567",
  "arn:aws:kms:us-east-1:0123456789ab:key/01234567-0123-0123-0123-0123456789ab"
  ],
  "detail": {
    "event": "reattachVolume",
    "result": "failed",
    "cause": "arn:aws:kms:us-
east-1:0123456789ab:key/01234567-0123-0123-0123-0123456789ab is pending deletion.",
    "request-id": ""
  }
}
```

볼륨 분리(detachVolume)

Amazon EC2 인스턴스에서 볼륨이 분리되면 detachVolume 이벤트가 AWS 계정으로 전송됩니다.

이벤트 데이터

다음은 성공적인 detachVolume 이벤트의 예입니다.

```
{
  "version":"0",
  "id":"2ec37298-1234-e436-70fc-c96b1example",
  "detail-type":"AWS API Call via CloudTrail",
  "source":"aws.ec2",
  "account":"123456789012",
  "time":"2024-03-18T16:35:52Z",
  "region":"us-east-1",
  "resources":[],
  "detail":
  {
```

```
    "eventVersion":"1.09",
    "userIdentity":
    {
      "type":"IAMUser",
      "principalId":"AIDAJT12345SQ2EXAMPLE",
      "arn":"arn:aws:iam::123456789012:user/administrator",
      "accountId":"123456789012",
      "accessKeyId":"AKIAJ67890A6EXAMPLE",
      "userName":"administrator"
    },
    "eventTime":"2024-03-18T16:35:52Z",
    "eventSource":"ec2.amazonaws.com",
    "eventName":"DetachVolume",
    "awsRegion":"us-east-1",
    "sourceIPAddress":"12.12.123.12",
    "userAgent":"aws-cli/2.7.12 Python/3.9.11 Windows/10 exe/AMD64 prompt/off command/
ec2.detach-volume",
    "requestParameters":
    {
      "volumeId":"vol-072577c46bexample",
      "force":false
    },
    "responseElements":
    {
      "requestId":"1234513a-6292-49ea-83f8-85e95example",
      "volumeId":"vol-072577c46bexample",
      "instanceId":"i-0217f7eb3dexample",
      "device":"/dev/sdb",
      "status":"detaching",
      "attachTime":1710776815000
    },
    "requestID":"1234513a-6292-49ea-83f8-85e95example",
    "eventID":"1234551d-a15a-43eb-9e69-c983aexample",
    "readOnly":false,
    "eventType":"AwsApiCall",
    "managementEvent":true,
    "recipientAccountId":"123456789012",
    "eventCategory":"Management",
    "tlsDetails":
    {
      "tlsVersion":"TLSv1.3",
      "cipherSuite":"TLS_AES_128_GCM_SHA256",
      "clientProvidedHostHeader":"ec2.us-east-1.amazonaws.com"
    }
```

```
  }
}
```

EBS 볼륨 초기화 이벤트

스냅샷에서 Amazon EBS 볼륨을 생성하고 볼륨 초기화에 기본 볼륨 초기화 속도 또는 Amazon EBS 프로비저닝 속도를 사용하면 볼륨 초기화가 완료된 후 5분 이내에 `initializeVolume` 이벤트가 AWS 계정으로 전송됩니다. 자세한 내용은 볼륨 초기화에 Amazon EBS 프로비저닝 속도 사용 단원을 참조하십시오.

빠른 스냅샷 복원을 사용하여 생성된 볼륨에 대해서는 이벤트가 전송되지 않습니다.

> **Important**
>
> `completionTime`는 이벤트가 생성된 시기를 반영합니다. 이벤트는 초기화가 완료된 후 5분 이내에 생성되므로 `completionTime`은 초기화가 완료된 후 최대 5분이 될 수 있습니다.

다음은 이벤트 예제입니다.

```
{
   "version": "0",
   "id": "01234567-0123-0123-0123-012345678901",
   "detail-type": "EBS Volume Notification",
   "source": "aws.ec2",
   "account": "012345678901",
   "time": "yyyy-mm-ddThh:mm:ssZ",
   "region": "us-east-1",
   "resources": [
      "arn:aws:ec2:us-east-1:012345678901:volume/vol-01234567890abcdef"
   ],
   "detail": {
      "event": " initializeVolume",
      "result": "succeeded",
      "completionTime": "yyyy-mm-ddThh:mm:ssZ",
      "request-id": "01234567-0123-0123-0123-0123456789ab"
   }
}
```

EBS 볼륨 수정 이벤트

Amazon EBS는 볼륨이 수정될 때 EventBridge에 `modifyVolume` 이벤트를 보냅니다. 그러나 저장, 로깅, 아카이빙은 되지 않습니다.

```
{
   "version": "0",
   "id": "01234567-0123-0123-0123-012345678901",
   "detail-type": "EBS Volume Notification",
   "source": "aws.ec2",
   "account": "012345678901",
   "time": "yyyy-mm-ddThh:mm:ssZ",
   "region": "us-east-1",
   "resources": [
      "arn:aws:ec2:us-east-1:012345678901:volume/vol-03a55cf56513fa1b6"
   ],
   "detail": {
      "result": "optimizing",
      "cause": "",
      "event": "modifyVolume",
      "request-id": "01234567-0123-0123-0123-0123456789ab"
   }
}
```

EBS 스냅샷 이벤트

Amazon EBS는 다음과 같은 볼륨 이벤트가 발생하는 경우 EventBridge에 이벤트를 보냅니다.

이벤트

- 스냅샷 생성(createSnapshot)
- 스냅샷 생성(createSnapshots)
- 스냅샷 복사(copySnapshot)
- 스냅샷 공유(shareSnapshot)

스냅샷 생성(createSnapshot)

스냅샷 생성 작업이 완료되면 `createSnapshot` 이벤트가 AWS 계정으로 전송됩니다. 그러나 저장, 로깅, 아카이빙은 되지 않습니다. 이 이벤트에서 `succeeded` 또는 `failed` 결과가 발생할 수 있습니다.

이벤트 데이터

아래 목록에 성공적인 createSnapshot 이벤트에 대해 EBS가 발생시키는 JSON 객체의 예를 열거했습니다. detail 섹션에서 source 필드에는 소스 볼륨의 ARN이 들어 있습니다. startTime 및 endTime 필드는 스냅샷 생성이 시작된 시점과 완료된 시점을 나타냅니다.

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "EBS Snapshot Notification",
  "source": "aws.ec2",
  "account": "012345678901",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "region": "us-east-1",
  "resources": [
     "arn:aws:ec2::us-west-2:snapshot/snap-01234567"
  ],
  "detail": {
    "event": "createSnapshot",
    "result": "succeeded",
    "cause": "",
    "request-id": "",
    "snapshot_id": "arn:aws:ec2::us-west-2:snapshot/snap-01234567",
    "source": "arn:aws:ec2::us-west-2:volume/vol-01234567",
    "startTime": "yyyy-mm-ddThh:mm:ssZ",
    "endTime": "yyyy-mm-ddThh:mm:ssZ"  }
}
```

스냅샷 생성(createSnapshots)

다중 볼륨 스냅샷을 생성하는 작업이 완료되면 createSnapshots 이벤트가 AWS 계정으로 전송됩니다. 이 이벤트에서 succeeded 또는 failed 결과가 발생할 수 있습니다.

이벤트 데이터

아래 목록에 성공적인 createSnapshots 이벤트에 대해 EBS가 발생시키는 JSON 객체의 예를 열거했습니다. detail 섹션의 source 필드에는 다중 볼륨 스냅샷 세트의 소스 볼륨 ARN이 포함되어 있습니다. startTime 및 endTime 필드는 스냅샷 생성이 시작된 시점과 완료된 시점을 나타냅니다.

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

```
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "EBS Multi-Volume Snapshots Completion Status",
  "source": "aws.ec2",
  "account": "012345678901",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "region": "us-east-1",
  "resources": [
    "arn:aws:ec2::us-east-1:snapshot/snap-01234567",
    "arn:aws:ec2::us-east-1:snapshot/snap-012345678"
  ],
  "detail": {
    "event": "createSnapshots",
    "result": "succeeded",
    "cause": "",
    "request-id": "",
    "startTime": "yyyy-mm-ddThh:mm:ssZ",
    "endTime": "yyyy-mm-ddThh:mm:ssZ",
    "snapshots": [
      {
        "snapshot_id": "arn:aws:ec2::us-east-1:snapshot/snap-01234567",
        "source": "arn:aws:ec2::us-east-1:volume/vol-01234567",
        "status": "completed"
      },
      {
        "snapshot_id": "arn:aws:ec2::us-east-1:snapshot/snap-012345678",
        "source": "arn:aws:ec2::us-east-1:volume/vol-012345678",
        "status": "completed"
      }
    ]
  }
}
```

아래 목록에 createSnapshots 이벤트 실패 이후 EBS가 발생시키는 JSON 객체의 예를 열거했습니다. 실패 원인은 다중 볼륨 스냅샷 세트에 대한 하나 이상의 스냅샷이 완료되지 못했기 때문입니다. snapshot_id의 값은 실패한 스냅샷의 ARN입니다. startTime 및 endTime은 스냅샷 생성 작업이 시작 및 종료된 시기를 나타냅니다.

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "EBS Multi-Volume Snapshots Completion Status",
  "source": "aws.ec2",
  "account": "012345678901",
```

```
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "region": "us-east-1",
  "resources": [
    "arn:aws:ec2::us-east-1:snapshot/snap-01234567",
    "arn:aws:ec2::us-east-1:snapshot/snap-012345678"
  ],
"detail": {
    "event": "createSnapshots",
    "result": "failed",
    "cause": "Snapshot snap-01234567 is in status error",
   "request-id": "",
    "startTime": "yyyy-mm-ddThh:mm:ssZ",
    "endTime": "yyyy-mm-ddThh:mm:ssZ",
    "snapshots": [
      {
        "snapshot_id": "arn:aws:ec2::us-east-1:snapshot/snap-01234567",
        "source": "arn:aws:ec2::us-east-1:volume/vol-01234567",
        "status": "error"
      },
      {
        "snapshot_id": "arn:aws:ec2::us-east-1:snapshot/snap-012345678",
        "source": "arn:aws:ec2::us-east-1:volume/vol-012345678",
        "status": "error"
      }
    ]
  }
}
```

스냅샷 복사(copySnapshot)

스냅샷 복사 작업이 완료되면 copySnapshot 이벤트가 AWS 계정으로 전송됩니다. 그러나 저장, 로깅, 아카이빙은 되지 않습니다. 이 이벤트에서 succeeded 또는 failed 결과가 발생할 수 있습니다.

detail 섹션에서 source는 소스 스냅샷의 ARN이고 snapshot_id는 스냅샷 복사본의 ARN입니다. startTime 및 endTime은 복사 작업이 시작되고 종료된 시기를 나타냅니다. incremental은 스냅샷 복사가 증분 스냅샷(true)인지 아니면 전체 스냅샷(false)인지를 나타냅니다. transferType은 스냅샷 복사 작업이 표준 복사 작업인지 아니면 시간 기반 복사 작업인지를 나타냅니다. 자세한 내용은 Amazon EBS 스냅샷 및 EBS 지원 AMI의 시간 기반 복사본 단원을 참조하십시오.

리전 간에 스냅샷을 복사하는 경우 대상 리전에서 이벤트가 발생합니다.

시나리오 1: 표준 스냅샷 복사 작업이 완료됨

다음은 표준 스냅샷 복사 작업이 성공적으로 완료되었을 때 계정으로 전송되는 이벤트의 예제입니다. `transferType`은 `standard`임을 참고하세요.

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "EBS Snapshot Notification",
  "source": "aws.ec2",
  "account": "123456789012",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "region": "us-east-1",
  "resources": [
    "arn:aws:ec2::us-west-2:snapshot/snap-01234567"
  ],
  "detail": {
    "event": "copySnapshot",
    "result": "succeeded",
    "cause": "",
    "request-id": "",
    "snapshot_id": "arn:aws:ec2::us-west-2:snapshot/snap-01234567",
    "source": "arn:aws:ec2::eu-west-1:snapshot/snap-76543210",
    "startTime": "yyyy-mm-ddThh:mm:ssZ",
    "endTime": "yyyy-mm-ddThh:mm:ssZ",
    "incremental": "true",
    "transferType": "standard"
  }
}
```

시나리오 2: 시간 기반 스냅샷 복사 작업이 완료 기간 내에 완료됨

다음은 시간 기반 스냅샷 복사 작업이 완료 기간 내에 완료될 때 계정으로 전송되는 이벤트의 예제입니다. `transferType`은 시간 기반 스냅샷 복사 작업임을 나타내는 `time-based`입니다. `completionDurationStartTime`은 완료 기간이 시작된 시점을 나타냅니다.

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "EBS Snapshot Notification",
  "source": "aws.ec2",
  "account": "123456789012",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "region": "us-east-1",
```

```
  "resources": [
    "arn:aws:ec2::us-west-2:snapshot/snap-01234567"
  ],
  "detail": {
    "event": "copySnapshot",
    "result": "succeeded",
    "cause": "",
    "request-id": "",
    "startTime": "yyyy-mm-ddThh:mm:ssZ",
    "endTime": "yyyy-mm-ddThh:mm:ssZ",
    "snapshot_id": "arn:aws:ec2::us-west-2:snapshot/snap-01234567",
    "source": "arn:aws:ec2::eu-west-1:snapshot/snap-76543210",
    "incremental": "true",
    "completionDurationStartTime":"2024-11-16T06:27:33.816Z",
    "transferType": "time-based"
  }
}
```

시나리오 3: 시간 기반 스냅샷 복사 작업이 완료되었지만 요청된 완료 기간을 놓침

시간 기반 스냅샷 복사 작업이 완료되었지만 요청된 완료 기간을 충족하지 못하면 CloudWatch에서 두 개의 이벤트를 계정에 전송합니다. 다음은 이러한 이벤트의 예제입니다.

- 첫 번째 이벤트는 복사 작업이 아직 진행 중인 경우에도 완료 기간을 놓치는 즉시 계정으로 전송됩니다. 이 이벤트의 경우 `detail-type`은 `EBS Copy Snapshot Missed Completion Duration`이며 `missedCompletionDurationCause`는 해당 이유를 제공합니다.

```
{
 "version":"0",
 "id":"fd90eb95-0938-e02c-cf55-b81363b8ac12",
 "detail-type":"EBS Copy Snapshot Missed Completion Duration",
 "source":"aws.ec2",
 "account":"123456789012",
 "time":"2024-11-19T18:17:08Z",
 "region":"us-east-1",
 "resources": ["arn:aws:ec2:us-east-1:123456789012:snapshot/snap-01234567890abcedf"],
 "detail":{
   "event":"copySnapshot",
   "missedCompletionDurationCause":"Snapshot copy was not able to meet the specified
 completion duration because your snapshot copy operation throughput quota was
 exceeded.",
   "snapshot_id":"arn:aws:ec2:us-
east-1:123456789012:snapshot/snap-01234567890abcedf",
```

```
    "source":"arn:aws:ec2:us-east-1:123456789012:snapshot/snap-00987654321fedcba",
    "startTime": "Sun Nov 24 22:32:55 UTC 2024",
    "transferType": "time-based"
    }
}
```

- 두 번째 이벤트는 스냅샷이 완료된 후에만 계정에 전송됩니다. 이벤트는 해당 이유를 제공하는 missedCompletionDurationCause를 포함합니다.

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "EBS Snapshot Notification",
  "source": "aws.ec2",
  "account": "123456789012",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "region": "us-east-1",
  "resources": [
    "arn:aws:ec2::us-west-2:snapshot/snap-01234567"
  ],
  "detail": {
    "event": "copySnapshot",
    "result": "succeeded",
    "cause": "",
    "request-id": "",
    "startTime": "yyyy-mm-ddThh:mm:ssZ",
    "endTime": "yyyy-mm-ddThh:mm:ssZ",
    "snapshot_id": "arn:aws:ec2::us-west-2:snapshot/snap-01234567",
    "source": "arn:aws:ec2::eu-west-1:snapshot/snap-76543210",
    "incremental": "true",
    "completionDurationStartTime":"2024-11-16T06:27:33.816Z",
    "missedCompletionDurationCause":"Snapshot copy was not able to meet the specified
 completion duration because your snapshot copy operation throughput quota was
 exceeded.",
    "transferType": "time-based"
  }
}
```

시나리오 4: 스냅샷 복사 작업이 실패함

다음은 스냅샷 복사 작업이 실패할 때 계정으로 전송되는 이벤트의 예제입니다. result는 작업이 실패했음을 나타내는 failed입니다.

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "EBS Snapshot Notification",
  "source": "aws.ec2",
  "account": "123456789012",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "region": "us-east-1",
  "resources": [
    "arn:aws:ec2::us-west-2:snapshot/snap-01234567"
  ],
  "detail": {
    "event": "copySnapshot",
    "result": "failed",
    "cause": "Source snapshot ID is not valid",
    "request-id": "",
    "snapshot_id": "arn:aws:ec2::us-west-2:snapshot/snap-01234567",
    "source": "arn:aws:ec2::eu-west-1:snapshot/snap-76543210",
    "startTime": "yyyy-mm-ddThh:mm:ssZ",
    "endTime": "yyyy-mm-ddThh:mm:ssZ"
  }
}
```

스냅샷 공유(shareSnapshot)

다른 AWS 계정이 스냅샷을 공유하면 shareSnapshot 이벤트가 계정으로 전송됩니다. 그러나 저장, 로깅, 아카이빙은 되지 않습니다. 결과는 항상 succeeded입니다.

이벤트 데이터

아래에 shareSnapshot 이벤트 완료 이후 EBS가 발생시키는 JSON 객체의 예를 열거했습니다. detail 섹션에서의 값은 스냅샷을 공유한 사용자의 AWS 계정 번호source입니다. startTime 및는 스냅샷 공유 작업이 시작되고 종료된 시기를 endTime 나타냅니다. shareSnapshot 이벤트는 프라이빗 스냅샷이 다른 사용자와 공유될 때만 발생합니다. 퍼블릭 스냅샷 공유는 이벤트를 트리거하지 않습니다.

```
{
  "version": "0",
  "id": "01234567-01234-0123-0123-012345678901",
  "detail-type": "EBS Snapshot Notification",
  "source": "aws.ec2",
  "account": "012345678901",
```

```
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "region": "us-east-1",
  "resources": [
    "arn:aws:ec2::us-west-2:snapshot/snap-01234567"
  ],
  "detail": {
    "event": "shareSnapshot",
    "result": "succeeded",
    "cause": "",
    "request-id": "",
    "snapshot_id": "arn:aws:ec2::us-west-2:snapshot/snap-01234567",
    "source": 012345678901,
    "startTime": "yyyy-mm-ddThh:mm:ssZ",
    "endTime": "yyyy-mm-ddThh:mm:ssZ"
  }
}
```

EBS 스냅샷 아카이브 이벤트

Amazon EBS는 스냅샷 아카이빙 작업과 관련된 이벤트를 내보냅니다. 자세한 내용을 알아보려면 CloudWatch Events를 사용하여 Amazon EBS 스냅샷 아카이빙 모니터링 섹션을 참조하세요.

EBS 빠른 스냅샷 복원 이벤트

Amazon EBS는 스냅샷에 대한 빠른 스냅샷 복원 상태가 변경되면 이벤트를 EventBridge로 보냅니다. 이벤트는 최선의 작업을 기반으로 발생됩니다.

다음은 이 이벤트의 예제 데이터입니다.

```
{
   "version": "0",
   "id": "01234567-0123-0123-0123-012345678901",
   "detail-type": "EBS Fast Snapshot Restore State-change Notification",
   "source": "aws.ec2",
   "account": "123456789012",
   "time": "yyyy-mm-ddThh:mm:ssZ",
   "region": "us-east-1",
   "resources": [
      "arn:aws:ec2:us-east-1::snapshot/snap-03a55cf56513fa1b6"
   ],
   "detail": {
      "snapshot-id": "snap-1234567890abcdef0",
```

```
        "state": "optimizing",
        "zone": "us-east-1a",
        "message": "Client.UserInitiated - Lifecycle state transition",
    }
}
```

가능한 state 값은 enabling, optimizing, enabled, disabling, disabled입니다.

message에 가능한 값은 다음과 같습니다.

Client.InvalidSnapshot.InvalidState - The requested snapshot transitioned to an invalid state (Error)

빠른 스냅샷 복원을 요청하는 요청이 실패하고 상태가 disabling 또는 disabled로 전환되었습니다. 빠른 스냅샷 복원은 이 스냅 샷에 대해 활성화할 수 없습니다.

Client.UserInitiated

상태가 enabling 또는 disabling으로 전환되었습니다.

Client.UserInitiated - Lifecycle state transition

상태가 optimizing, enabled 또는 disabled로 전환되었습니다.

Server.InsufficientCapacity - There was insufficient capacity available to satisfy the request

불충분한 용량으로 인해 빠른 스냅샷 복원을 요청하는 요청이 실패하고 상태가 disabling 또는 disabled로 전환되었습니다. 잠시 기다렸다가 다시 시도하세요.

Server.InternalError - An internal error caused the operation to fail

내부 오류로 인해 빠른 스냅샷 복원을 요청하는 요청이 실패하고 상태가 disabling 또는 disabled로 전환되었습니다. 잠시 기다렸다가 다시 시도하세요.

Client.InvalidSnapshot.InvalidState - The requested snapshot was deleted or access permissions were revoked

스냅샷이 스냅샷 소유자에 의해 삭제되거나 공유 해제되었으므로 스냅샷에 대한 빠른 스냅샷 복원 상태가 disabling 또는 disabled로 전환되었습니다. 삭제되었거나 더 이상 본인에게 공유되지 않은 스냅샷에 대해서는 빠른 스냅샷 복원을 활성화할 수 없습니다.

AWS Lambda 를 사용하여 EventBridge 이벤트 처리

Amazon EBS와 Amazon EventBridge를 사용하여 데이터 백업 워크플로를 자동화할 수 있습니다. 이렇게 하려면 IAM 정책, 이벤트를 처리하는 AWS Lambda 함수, 수신 이벤트와 일치하고 이를 Lambda 함수로 라우팅하는 EventBridge 규칙을 생성해야 합니다.

다음 절차에서는 재해 복구를 위해 `createSnapshot` 이벤트를 사용하여 완료된 스냅샷을 다른 리전으로 자동으로 복사합니다.

완료된 스냅샷을 다른 리전으로 복사하려면

1. 다음 예에 표시된 것과 같이 `CopySnapshot` 작업을 사용하고 EventBridge 로그에 쓸 수 있는 권한을 제공하려면 IAM 정책을 생성합니다. EventBridge 이벤트를 처리할 사용자에게 정책을 할당합니다.

   JSON

   ```
   {
     "Version":"2012-10-17",
     "Statement": [
       {
         "Effect": "Allow",
         "Action": [
           "logs:CreateLogGroup",
           "logs:CreateLogStream",
           "logs:PutLogEvents"
         ],
         "Resource": "arn:aws:logs:*:*:*"
       },
       {
         "Effect": "Allow",
         "Action": [
           "ec2:CopySnapshot"
         ],
         "Resource": "*"
       }
     ]
   }
   ```

2. EventBridge 콘솔에서 사용할 수 있는 Lambda 함수를 정의합니다. 아래 Node.js로 작성된 샘플 Lambda 함수는 일치하는 `createSnapshot` 이벤트를 Amazon EBS가 발생시킬 때 EventBridge

에 의해 호출됩니다. 이는 스냅샷이 완료되었음을 의미합니다. 호출되면 함수가 us-east-2에서 us-east-1로 스냅샷을 복사합니다.

```
// Sample Lambda function to copy an EBS snapshot to a different Region

var AWS = require('aws-sdk');
var ec2 = new AWS.EC2();

// define variables
var destinationRegion = 'us-east-1';
var sourceRegion = 'us-east-2';
console.log ('Loading function');

//main function
exports.handler = (event, context, callback) => {

    // Get the EBS snapshot ID from the event details
    var snapshotArn = event.detail.snapshot_id.split('/');
    const snapshotId = snapshotArn[1];
    const description = `Snapshot copy from ${snapshotId} in ${sourceRegion}.`;
    console.log ("snapshotId:", snapshotId);

    // Load EC2 class and update the configuration to use destination Region to
 initiate the snapshot.
    AWS.config.update({region: destinationRegion});
    var ec2 = new AWS.EC2();

    // Prepare variables for ec2.modifySnapshotAttribute call
    const copySnapshotParams = {
        Description: description,
        DestinationRegion: destinationRegion,
        SourceRegion: sourceRegion,
        SourceSnapshotId: snapshotId
    };

    // Execute the copy snapshot and log any errors
    ec2.copySnapshot(copySnapshotParams, (err, data) => {
        if (err) {
            const errorMessage = `Error copying snapshot ${snapshotId} to Region
 ${destinationRegion}.`;
            console.log(errorMessage);
            console.log(err);
            callback(errorMessage);
```

```
        } else {
            const successMessage = `Successfully started copy of snapshot
 ${snapshotId} to Region ${destinationRegion}.`;
            console.log(successMessage);
            console.log(data);
            callback(null, successMessage);
        }
    });
};
```

EventBridge 콘솔에서 Lambda 함수를 사용할 수 있는지 확인하려면 EventBridge 이벤트가 발생할 리전에서 함수를 생성합니다. 자세한 내용은 개발자 안내서AWS Lambda를 참조하세요.

3. Amazon EventBridge 콘솔(https://console.aws.amazon.com/events/)을 엽니다.
4. 탐색 창에서 규칙(Rules)을 선택한 후 규칙 생성(Create rule)을 선택합니다.
5. Step 1: Define rule detail(1단계: 규칙 세부 정보 정의)에 대해 다음을 수행합니다.
   a. Name(이름)과 Description(설명)을 입력합니다.
   b. Event bus(이벤트 버스)의 경우 default(기본값)를 유지합니다.
   c. 선택한 이벤트 버스에서 규칙 활성화가 켜져 있는지 확인합니다.
   d. Event type(이벤트 유형)에서 Rule with an event pattern(이벤트 패턴이 있는 규칙)을 선택합니다.
   e. 다음을 선택합니다.
6. Step 2: Build event pattern(2단계: 이벤트 패턴 빌드)에서 다음을 수행합니다.
   a. Event source(이벤트 소스)에서 AWS events or EventBridge partner events를 선택합니다.
   b. 이벤트 패턴 섹션의 이벤트 소스에서 AWS 서비스가 선택되어 있는지 확인하고AWS , 서비스에서 EC2를 선택합니다.
   c. Event type(이벤트 유형)에서 EBS Snapshot Notification(EBS 스냅샷 알림)을 선택하고 Specific event(s)(특정 이벤트)를 선택한 다음 createSnapshot을 선택합니다.
   d. 특정 결과(Specific result(s))를 선택한 다음 성공(succeeded)을 선택합니다.
   e. 다음을 선택합니다.
7. Step 3: Select targets(3단계: 대상 선택)에서 다음을 수행합니다.
   a. 대상 유형에서 AWS 서비스를 선택합니다.
   b. Select target(대상 선택)에서 Lambda function(Lambda 함수)을 선택하고 Function(함수)에서 이전에 생성한 함수를 선택합니다.

c. Next(다음)를 선택합니다.

8. Step 4: Configure tags(4단계: 태그 구성)에서 필요한 경우 규칙에 대한 태그를 지정하고 Next(다음)를 선택합니다.
9. Step 5: Review and create(5단계: 검토 및 생성)에서 규칙을 검토한 다음 Create rule(규칙 생성)을 선택합니다.

이제 규칙 탭에 규칙이 표시됩니다. 표시된 예에서 구성된 이벤트는 다음에 스냅샷을 복사할 때 EBS가 발생시킵니다.

Amazon EBS 세부 성능 통계

Amazon EBS NVMe 블록 디바이스는 Nitro 기반 Amazon EC2 인스턴스에 연결된 Amazon EBS 볼륨에 대한 실시간의 고해상도 I/O 성능 통계를 제공합니다. 이러한 통계는 볼륨이 인스턴스에 연결된 기간 동안 유지되는 집계된 카운터로 표시됩니다. 통계는 누적 작업 수, 송수신된 바이트 수, 읽기 및 쓰기 I/O 작업에 소요된 시간에 대한 세부 정보를 제공합니다. 또한 통계에는 읽기 및 쓰기 I/O 작업에 대한 히스토그램이 포함됩니다. 또한 애플리케이션이 EBS 볼륨 또는 연결된 인스턴스의 프로비저닝된 IOPS 또는 처리량 제한을 초과한 총 시간을 보고합니다.

이러한 통계는 최대 1초 간격으로 세분화하여 수집할 수 있습니다. 요청이 1초 간격보다 더 자주 발생할 경우, NVMe 드라이버는 해당 요청을 다른 관리 명령어와 함께 대기열에 넣어 추후에 처리할 수 있습니다.

고려 사항

- 통계는 모든 Amazon EBS 볼륨 유형에 대해 지원됩니다.
- 통계는 AWS Nitro System에 구축된 인스턴스에 연결된 볼륨에 대해서만 지원됩니다.
- 통계는 다중 연결 지원 볼륨에 사용할 수 있습니다. 다중 연결 지원 볼륨에 대한 통계를 볼 때 통계는 해당 인스턴스 연결에 고유하며 해당 인스턴스의 사용량만 반영합니다.
- 통계는 추가 비용 없이 사용할 수 있습니다.

통계

Amazon EBS NVMe 블록 디바이스는 다음 통계를 제공합니다.

통계 이름	전체 이름	유형	설명
total_rea d_ops	총 읽기 작업	Counter	완료된 총 읽기 작업 수입니다.
total_wri te_ops	총 쓰기 작업	Counter	완료된 총 쓰기 작업 수입니다.
total_rea d_bytes	총 읽기 바이트	Counter	전송된 총 읽기 바이트 수입니다.
total_wri te_bytes	총 쓰기 바이트	Counter	전송된 총 쓰기 바이트 수입니다.
total_rea d_time	총 읽기 시간	Counter	모든 완료된 읽기 작업에 소요된 총 시간(밀리초)입니다.
total_wri te_time	총 쓰기 시간	Counter	모든 완료된 쓰기 작업에 소요된 총 시간(마이크로초)입니다.
ebs_volu e_perfor ance_exc eded_iop	수요가 볼륨의 프로비저닝된 IOPS를 초과한 총 시간	Counter	IOPS 수요가 볼륨의 프로비저닝된 IOPS 성능을 초과한 총 시간(마이크로초)입니다.
ebs_volu e_perfor ance_exc eded_tp	수요가 볼륨의 프로비저닝된 처리량을 초과한 총 시간	Counter	처리량 수요가 볼륨의 프로비저닝된 처리량 성능을 초과한 총 시간(마이크로초)입니다.
ec2_inst nce_ebs_ erforman e_exceed d_iops	수요가 EC2 인스턴스의 IOPS 성능을 초과한 총 시간	Counter	EBS 볼륨이 연결된 Amazon EC2 인스턴스의 최대 IOPS 성능을 초과한 총 시간(마이크로초)입니다.

통계 이름	전체 이름	유형	설명
ec2_insta nce_ebs_p erformanc e_exceede d_tp	수요가 EC2 인스턴스의 처리량 성능을 초과한 총 시간	Counter	EBS 볼륨이 연결된 Amazon EC2 인스턴스의 최대 처리량 성능을 초과한 총 시간(마이크로초)입니다.
volume_qu eue_lengt h	볼륨 대기열 길이	특정 시점	완료되기를 기다리는 읽기 및 쓰기 작업의 수입니다.
read_io_l atency_hi stogram	I/O 히스토그램 읽기	히스토그램 *	각 지연 시간 빈 내에서 마이크로초 단위로 완료된 읽기 작업 수입니다.
write_io_ latency_h istogram	I/O 히스토그램 쓰기	히스토그램 *	각 지연 시간 빈 내에서 마이크로초 단위로 완료된 쓰기 작업 수입니다.

> **Note**
>
> * 히스토그램 통계는 성공적으로 완료된 I/O 작업만 나타냅니다. 정지되거나 손상된 I/O 작업은 포함되지 않지만, 특정 시점의 통계로 제공되는 `volume_queue_length` 통계에서 확인할 수 있습니다.

통계 액세스

통계는 Amazon EBS 볼륨이 연결된 인스턴스에서 직접 액세스해야 합니다. 다음 방법 중 하나로 통계에 액세스할 수 있습니다.

Linux 인스턴스

Amazon CloudWatch

Amazon CloudWatch 에이전트를 구성하여 인스턴스에서 통계를 수집하고 CloudWatch에서 사용자 정의 메트릭으로 사용할 수 있도록 할 수 있습니다. 그런 다음 CloudWatch의 지표를 사용하여 I/

O 패턴을 분석하고, 성능 추세를 추적하며, 사용자 지정 대시보드를 생성하고, 성능 임계값을 기반으로 자동화된 경보를 설정할 수 있습니다.

CloudWatch 에이전트 구성에 대한 자세한 내용은 다음을 참조하세요.

- CloudWatch 에이전트 구성 파일 생성
- Amazon EBS NVMe 드라이버 지표 수집

Amazon CloudWatch Observability EKS 추가 기능 버전 `4.1.0` 이상에서는 Amazon EBS CSI 드라이버 지표가 활성화되면 통계가 자동 수집됩니다. 자세한 내용은 Amazon EBS NVMe 드라이버 지표를 참조하세요.

ebsnvme script

`ebsnvme` 스크립트는 amazon-ec2-utils Github 리포지토리에서 찾을 수 있습니다.

통계에 액세스하려면

1. 볼륨이 연결된 인스턴스에 연결합니다.
2. `amazon-ec2-utils` GitHub 리포지토리에서 `ebsnvme` 스크립트를 다운로드합니다.

```
wget https://raw.githubusercontent.com/amazonlinux/amazon-ec2-utils/refs/heads/main/ebsnvme
```

3. 스트립트가 실행 가능하도록 권한을 수정합니다.

```
sudo chmod +x ./ebsnvme
```

4. `ebsnvme` 스크립트를 실행하고 볼륨의 디바이스 이름을 지정합니다.

```
sudo ./ebsnvme stats /dev/nvme0n1
```

nvme-cli tool

통계에 액세스하려면

1. 볼륨이 연결된 인스턴스에 연결합니다.
2. 2024년 11월 12일 이후에 출시된 Amazon Linux AMI에는 최신 버전의 `nvme-cli` 도구가 포함됩니다. 이전 버전의 Amazon Linux AMI를 사용 중인 경우 `nvme-cli` 도구를 업데이트하세요.

```
sudo yum install nvme-cli
```

3. 다음 명령을 실행하고 볼륨의 디바이스 이름을 지정합니다.

```
nvme amzn stats /dev/nvme0n1
```

Prometheus

오픈 소스 모니터링 애플리케이션인 Prometheus 및 Amazon Managed Service for Prometheus를 사용하여 통계를 모니터링할 수 있습니다. 이는 컨테이너 및 Kubernetes 환경 전반에서 Amazon EBS 볼륨을 대규모로 더 쉽게 모니터링할 수 있도록 합니다. Amazon EBS CSI 드라이버 v1.37.0 이상에서는 세부 성능 통계가 Prometheus로 내보내기 위한 Prometheus 호환 `/metrics` 엔드포인트로 노출됩니다.

자세한 내용은 Amazon Managed Service for Prometheus 사용 설명서의 Amazon Managed Service for Prometheus 워크스페이스로 지표 수집을 참조하세요.

Windows 인스턴스

nvme_amzn.exe tool

통계에 액세스하려면

1. 볼륨이 연결된 인스턴스에 연결합니다.
2. AWSNVMe 드라이버 버전 `1.7.0` 이상을 사용하고 있는지 확인합니다. AWSNVMe 드라이버 업데이트에 대한 자세한 내용은 AWS NVMe 드라이버를 참조하세요.
3. EBS 볼륨의 디스크 번호를 가져옵니다. 자세한 내용은 Amazon EBS 볼륨을 NVMe 디바이스 이름에 매핑을 참조하세요.
4. 다음 명령을 관리자로 실행하고 볼륨의 디스크 번호를 지정합니다.

```
.\nvme_amzn.exe stats disk_number
```

Amazon EBS용 Amazon GuardDuty

Amazon GuardDuty는 환경 내의 계정, 컨테이너, 워크로드 및 데이터를 보호하는 데 도움이 되는 위협 탐지 서비스입니다 AWS . GuardDuty는 기계 학습(ML) 모델, 이상 및 위협 감지 기능을 통해 다양한 로

그 소스와 런타임 활동을 지속적으로 모니터링하여 사용자 환경의 잠재적 보안 위험과 악의적 활동을 식별하고 우선순위를 지정합니다.

GuardDuty 내의 맬웨어 보호 기능은 Amazon EC2 인스턴스 및 컨테이너 워크로드와 연결된 Amazon EBS 볼륨을 스캔하여 잠재적 위협을 탐지합니다. GuardDuty는 이를 수행하는 두 가지 방법을 제공합니다.

- 맬웨어 보호 활성화 - GuardDuty가 Amazon EC2 인스턴스 또는 컨테이너 워크로드에 맬웨어가 존재할 가능성을 나타내는 조사 결과를 생성하면 잠재적으로 손상된 리소스에 대한 맬웨어 스캔을 자동으로 시작합니다.
- 맬웨어 보호를 활성화하지 않고 온디맨드 맬웨어 스캔 사용 - Amazon EC2 인스턴스의 Amazon 리소스 이름(ARN)을 제공하여 온디맨드 스캔을 시작합니다.

자세한 내용은 Amazon GuardDuty 사용 설명서를 참조하세요.

결제 및 사용량 보고서의 Amazon EBS 코드 이해

Amazon EBS를 사용하는 경우 AWS 결제 및 사용 보고서에 관련 코드가 포함됩니다. 이러한 코드를 검토하면 Amazon EBS의 비용 및 사용량 패턴을 이해하는 데 도움이 됩니다. 비용 추적 및 관리는 비용 최적화에 있어 필수적인 요소입니다.

다음 표에서는 결제 및 사용량 보고서에 표시되는 Amazon EBS 코드를 설명합니다. 결제 및 사용량 보고서에 사용되는 리전 코드 목록은 AWS 리전 결제 코드를 참조하세요.

결제 코드:

- 스냅샷
- 볼륨 스토리지
- 프로비저닝된 성능
- EBS 다이렉트 API
- 빠른 스냅샷 복원(FSR)

관련 리소스

- Amazon EBS 요금 책정
- the section called “가격 책정”

스냅샷

코드	설명	단위	세부 수준
region-EBS:SnapshotUsage	표준 계층 스냅샷(증분 복사본).	GB-월	시간당
region-EBS:SnapshotArchiveStorage	아카이브 계층 스냅샷(전체 복사본).	GB-월	시간당
region-EBS:SnapshotArchiveEarlyDelete	최소 보존 기간이 끝나기 전에 아카이브된 스냅샷 삭제.	GB	삭제당
region-EBS:SnapshotArchiveRetrieval	아카이브된 스냅샷 검색.	GB	GB당

볼륨 스토리지

코드	설명	단위	세부 수준
region-EBS:VolumeUsage.gp3	범용 SSD(gp3) 볼륨을 위한 스토리지.	GB-월	시간당
region-EBS:Volu	범용 SSD(gp2) 볼륨을 위한 스토리지.	GB-월	시간당

코드	설명	단위	세부 수준
meUsage.gp2			
region-EBS:VolumeUsage.io2	프로비저닝된 IOPS SSD(io2) 볼륨을 위한 스토리지.	GB-월	시간당
region-EBS:VolumeUsage.piops	프로비저닝된 IOPS SSD(io1) 볼륨을 위한 스토리지.	GB-월	시간당
region-EBS:VolumeUsage.st1	처리량 최적화 HDD 볼륨을 위한 스토리지.	GB-월	시간당
region-EBS:VolumeUsage.sc1	콜드 HDD 볼륨을 위한 스토리지.	GB-월	시간당
region-EBS:VolumeUsage	마그네틱 볼륨을 위한 스토리지.	GB-월	시간당

프로비저닝된 성능

코드	설명	단위	세부 수준
region-EBS:Volu	gp3 볼륨의 125MB/s를 초과하여 프로비저닝된 처리량.	MB/s-월	시간당

코드	설명	단위	세부 수준
meP-Throughput.gp3			
region-EBS:VolumeP-IOPS.gp3	gp3 볼륨의 3,000 이상의 프로비저닝된 IOPS.	IOPS-월	시간당
region-EBS:VolumeP-IOPS.io2	io2 볼륨의 32,000까지의 프로버지닝된 IOPS.	IOPS-월	시간당
region-EBS:VolumeP-IOPS.io2.tier2	io2 볼륨의 32,001~64,000의 프로비저닝된 IOPS.	IOPS-월	시간당
region-EBS:VolumeP-IOPS.io2.tier3	io2 볼륨의 64,000을 초과하여 프로버지닝된 IOPS.	IOPS-월	시간당
region-EBS:VolumeP-IOPS.piops	io1 볼륨의 프로비저닝된 IOPS.	IOPS-월	시간당
region-EBS:VolumeIOUsage	레거시 IOPS.	IOPS-월	시간당

EBS 다이렉트 API

코드	설명	단위	세부 수준
region-EBS:directAPI.snapshot.List	ListChangedBlocks 및 ListSnapshotBlocks API 작업에 대한 호출.	요청 1,000개당	요청당
region-EBS:directAPI.snapshot.Get	GetSnapshotBlock API 작업에서 반환하는 블록.	반환된 블록 1,000개당	요청당
region-EBS:directAPI.snapshot.Put	PutSnapshotBlock API 작업에서 기록한 블록.	기록된 블록 1,000개당	요청당

빠른 스냅샷 복원(FSR)

코드	설명	단위	세부 수준
region-EBS:FastSnapshotRestore	빠른 스냅샷 복원을 위해 활성화된 스냅샷에 대한 데이터 서비스 단위-시간(DSU-시간)(AZ당).	WCU-시간	분당(최소 1시간)

EBS 볼륨의 인벤토리 생성

Amazon EBS는 EC2 인스턴스에서 사용할 수 있는 확장 가능한 고성능 블록 스토리지 리소스를 제공합니다. 단일 EBS 볼륨을 단일 EC2 인스턴스에 연결할 수 있습니다. 단일 EC2 인스턴스에서 단일 EBS 볼륨을 분리할 수도 있습니다.

다음은 EBS 스토리지의 주요 특성입니다. 시스템 관리자는 EBS 리소스의 이러한 특성에 대한 정보를 가져오고 이를 사용하여 온프레미스에서 실행되는 서버 또는 다른 클라우드 공급자의 서버에 기능적으로 동등한 스토리지를 구성할 수 있습니다. 또한 시스템 관리자는 EBS 리소스에 저장된 데이터를 검색한 다음 온프레미스 또는 다른 클라우드 공급자의 스토리지에 해당 데이터를 저장할 수 있습니다.

기능	설명
볼륨	EBS 볼륨은 내구성이 있는 블록 스토리지 디바이스이며 인스턴스를 연결하거나 포맷하거나 탑재할 수 있습니다.
I/O 성능	Amazon EBS는 성능 특성 및 가격이 다른 여러 볼륨 유형을 제공합니다.
EBS 데이터 볼륨의 콘텐츠	볼륨의 데이터에 직접 액세스하는 유일한 방법은 인스턴스에서 액세스하는 것입니다.
스냅샷	EBS 스냅샷은 EBS 볼륨에 있는 데이터의 특정 시점 복사본입니다. 스냅샷은 Amazon S3에서 사용자가 직접 액세스할 수 없는 버킷에 저장됩니다.
EBS 스냅샷의 콘텐츠	스냅샷에서 데이터를 읽거나 이를 사용하여 볼륨을 생성할 수 있습니다.

볼륨

EBS 볼륨을 생성할 때는 볼륨의 가용 영역을 선택합니다. 동일한 가용 영역에 있는 EC2 인스턴스에 EBS 볼륨을 연결해야 합니다. EC2 인스턴스를 시작할 때 볼륨을 생성하거나 먼저 볼륨을 생성한 다음 인스턴스에 연결할 수 있습니다. 볼륨이 인스턴스에 연결되면 상태가 `in-use`가 됩니다. 볼륨이 인스턴스에 연결되지 않았으면 상태가 `available`이 됩니다.

이것이 중요한 이유

EBS 루트 볼륨은 EC2 인스턴스의 운영 체제를 포함합니다. EBS 데이터 볼륨은 비즈니스 크리티컬 데이터를 포함할 수 있습니다. 모든 리전의 EBS 볼륨을 나열하고 해당 볼륨이 EC2 인스턴스에 연결되어 있는지 여부를 포함할 수 있습니다.

모든 리전의 EBS 볼륨에 대한 요약을 보려면

Amazon EC2 Global View를 사용하여 모든 리전에 있는 EBS 볼륨을 나열할 수 있습니다.

1. https://console.aws.amazon.com/ec2globalview/home에서 Amazon EC2 Global View 콘솔을 엽니다.
2. 리전 탐색기 탭의 요약에서 볼륨 수와 리전 수를 포함한 볼륨 리소스 수를 확인합니다. 밑줄이 그어진 텍스트를 선택하여 볼륨 수가 리전에 어떻게 분산되는지 확인합니다.
3. 전역 검색 탭에서 클라이언트 필터 리소스 유형 = 볼륨을 선택합니다. 리전 또는 태그를 지정하여 결과를 추가로 필터링할 수 있습니다.

사용 중인 EBS 볼륨을 설명하려면

describe-volumes 명령을 사용합니다. EBS 볼륨이 있는 각 리전에서 이 명령을 실행해야 합니다. `--filters` 파라미터는 결과의 범위를 `in-use`인 볼륨으로 지정합니다. `--query` 파라미터는 출력에 지정된 필드만 표시합니다. 필요에 따라 추가 필드를 포함할 수 있습니다.

```
aws ec2 describe-volumes \
    --filters Name=status,Values=in-use \
    --query "Volumes[].[VolumeId,Size,Attachments[0].InstanceId]" \
    --output table
```

다음은 예제 출력입니다. 열은 볼륨 ID, 볼륨 크기, 연결된 인스턴스의 ID 및 디바이스 이름입니다.

```
-------------------------------------------------------------------
|                          DescribeVolumes                        |
+-------------------------+-----+----------------------+---------+
|  vol-0992ee65bec96de19 |  8  |  i-00a7d9ec76a46a49f  |  /dev/xvda |
|  vol-04d631d22047db1cb |  30 |  i-0b1bf24fd4f297ab9  |  /dev/sda1 |
   ...
   ...
|  vol-0b811be5cbd1425cc | 100 |  i-0a8d998154b320257  |  /dev/xvde |
+-------------------------+-----+----------------------+---------+
```

사용 중이 아닌 EBS 볼륨을 설명하려면

이전 명령에서 다음과 같이 필터를 수정하여 상태가 `available`인 볼륨을 찾습니다.

```
--filters Name=status,Values=available
```

볼륨이 사용되지 않는 경우, 필요한 데이터가 볼륨에 있는지 확인해야 합니다. 예를 들어 테스트 또는 문제 해결을 위해 생성했던 볼륨을 삭제할 수 있습니다. 인스턴스에 볼륨을 연결한 블록 디바이스가 인스턴스 종료 후에도 지속되도록 구성된 경우에도 볼륨을 사용할 수 있습니다. 이 경우, 볼륨의 데이터가 여전히 필요한지 여부를 확인합니다.

I/O 성능

EBS 볼륨에 지정하는 볼륨 유형에 따라 성능이 결정됩니다. 볼륨 유형에 따라 볼륨의 크기, IOPS 및 처리량의 조합을 지정할 수 있습니다. 각 볼륨 유형의 최대 프로비저닝 IOPS 및 처리량에 대한 자세한 내용은 Amazon EBS 볼륨 유형을 참조하세요.

중요한 이유

EBS 볼륨의 성능을 결정한 후에는 기능적으로 동등한 서버에 필요한 최소 사양을 결정할 수 있습니다.

볼륨 성능에 대한 정보를 가져오려면

describe-volumes 명령을 사용하여 사용 중인 볼륨의 성능을 설명합니다. `--query` 파라미터는 출력에 지정된 필드만 표시합니다. 필요에 따라 추가 필드를 포함할 수 있습니다. EBS 최적화 성능에 대한 자세한 내용은 Amazon EC2 인스턴스 유형 안내서의 Amazon EC2 인스턴스 유형 사양을 참조하세요.

```
aws ec2 describe-volumes \
    --filters Name=status,Values=in-use \
    --query "Volumes[].[VolumeId,VolumeType,Iops,Throughput]" \
    --output table
```

다음은 예제 출력입니다. 열은 볼륨 ID, 볼륨 유형, IOPS 및 처리량(MiB/s)입니다.

```
-------------------------------------------------
|                DescribeVolumes                |
+-------------------------+------+-------+-------+
|  vol-0992ee65bec96de19 |  gp3 |  3000 |  125  |
|  vol-04d631d22047db1cb |  gp2 |  100  |  None |
   ...
   ...
|  vol-0b811be5cbd1425cc |  gp3 |  3000 |  125  |
+-------------------------+------+-------+-------+
```

EBS 데이터 볼륨의 콘텐츠

Amazon EBS는 EBS 볼륨에 저장하는 데이터에 액세스할 수 없습니다. 데이터 볼륨을 백업하는 것은 사용자의 책임입니다. 예를 들어 EBS 스냅샷이라고 하는 특정 시점 스냅샷을 생성할 수 있습니다.

EC2 인스턴스에 연결하고 EBS 볼륨에서 사용자 컴퓨터로 파일을 전송할 수도 있습니다. 이 작업을 수행하는 데 도움이 되는 많은 도구가 있습니다. 예를 들면 다음과 같습니다.

중요한 이유

EBS 볼륨에 비즈니스 크리티컬 데이터가 있는 경우, 해당 데이터를 기능적으로 동등한 스토리지에 복사할 수 있습니다.

Linux 인스턴스에 연결된 볼륨에서 파일을 전송하려면

Linux 인스턴스에 연결하고 SCP를 사용합니다. 자세한 내용은 SCP를 사용하여 파일 전송을 참조하세요.

Windows 인스턴스에 연결된 볼륨에서 파일을 전송하려면

Windows 인스턴스에 연결하고 RDP를 사용합니다. 자세한 내용은 RDP를 사용하여 파일 전송을 참조하세요.

스냅샷

EBS 스냅샷을 생성하여 EBS 데이터 볼륨을 백업할 수 있습니다. Amazon EBS, Amazon Data Lifecycle Manager 또는 AWS Backup을 사용하여 EBS 스냅샷을 생성할 수 있습니다.

중요한 이유

EBS 볼륨에 비즈니스 크리티컬 데이터가 있는 경우, EBS 스냅샷의 형태로 백업을 생성할 가능성이 높습니다. 볼륨의 최신 스냅샷을 찾아 볼륨의 현재 데이터를 반영하는지 확인할 수 있습니다.

특정 볼륨의 스냅샷을 설명하려면

describe-snapshots 명령을 사용합니다. `--filters` 파라미터는 결과의 범위를 지정된 볼륨의 스냅샷으로 지정합니다. `--query` 파라미터는 출력에 지정된 필드만 표시합니다. 필요에 따라 추가 필드를 포함할 수 있습니다.

```
aws ec2 describe-snapshots \
    --filters Name=volume-id,Values=vol-00622ef8c2ac8b762 \
```

```
    --query Snapshots[*].[SnapshotId,CompletionTime,Description] --output table
```

다음은 예제 출력입니다. 열은 스냅샷 ID, 완료 날짜 및 볼륨의 암호화 여부입니다.

```
-------------------------------------------------------------------------
|                            DescribeSnapshots                          |
+-------------------------+-----------------------------------+--------+
|  snap-0ad439c50efabb47c |  2024-07-12T03:36:27.952000+00:00  |  False |
+-------------------------+-----------------------------------+--------+
```

EBS 스냅샷의 콘텐츠

EBS 스냅샷의 콘텐츠는 다른 형식으로 다운로드하거나 내보낼 수 없습니다. 그러나 EBS 다이렉트 API를 사용하거나 스냅샷에서 EBS 볼륨을 생성하고 파일을 전송하여 스냅샷의 데이터에 액세스할 수 있습니다.

중요한 이유

EBS 볼륨에 비즈니스 크리티컬 데이터가 있는 경우, EBS 스냅샷의 형태로 백업을 생성할 가능성이 높습니다. 이 데이터를 검색하여 해당 데이터를 기능적으로 동등한 스토리지에 복사할 수 있습니다.

옵션 1: 스냅샷에서 데이터를 읽으려면

EBS 다이렉트 API를 사용합니다.

옵션 2: 스냅샷에서 볼륨을 생성하고 데이터를 전송하려면

1. create-volume 명령을 사용하여 스냅샷에서 볼륨을 만듭니다.

   ```
   aws ec2 create-volume \
      --volume-type gp3 \
      --snapshot-id snap-0ad439c50efabb47c \
      --iops 6000 \
      --availability-zone us-east-2b
   ```

2. attach-volume 명령을 사용하여 볼륨을 인스턴스에 연결합니다.

   ```
   aws ec2 attach-volume \
      --device /dev/sdf \
      --instance-id i-0b1bf24fd4f297ab9 \
      --volume-id vol-1234567890abcdef0
   ```

3. 인스턴스에 연결하고 볼륨을 포맷 및 탑재합니다.
4. 사용자 컴퓨터로 볼륨의 파일을 전송합니다.

Amazon EBS 할당량

AWS 계정 에는 각각에 대해 이전에 제한이라고 하는 기본 할당량이 있습니다 AWS 서비스. 다르게 표시되지 않는 한 리전별로 각 할당량이 적용됩니다. 일부 할당량에 대한 증가를 요청할 수 있으며 다른 할당량은 늘릴 수 없습니다.

Amazon EBS 할당량을 보려면 Service Quotas 콘솔을 엽니다. 탐색 창에서 AWS 서비스를 선택하고 Amazon Elastic Block Store(Amazon EBS)를 선택합니다. 할당량 증가를 요청하려면 Service Quotas 사용 설명서의 할당량 증가 요청을 참조하세요.

AWS 계정 에는 Amazon EBS와 관련된 다음과 같은 할당량이 있습니다.

이름	기본값	조정 가능	설명
볼륨당 아카이빙된 스냅샷	지원되는 각 지역: 25	예	볼륨당 아카이빙된 스냅샷의 최대수입니다.
계정당 CompleteSnapshot 요청	지원되는 지역별: 초당 10개	아니요	계정당 허용되는 최대 CompleteSnapshot 요청의 최대수입니다.
목적지 지역당 동시 스냅샷 복사 수	지원되는 각 지역: 20	아니요	목적지 지역당 동시 스냅샷 복사의 최대수입니다.
콜드 HDD(sc1) 볼륨당 동시 스냅샷	각 지원되는 지역: 1	아니요	이 지역의 콜드 HDD(sc1) 볼륨당 동시 스냅샷의 최대수입니다.
범용 SSD(gp2) 볼륨당 동시 스냅샷	각 지원되는 지역: 5	아니요	이 지역의 범용 SSD(gp2) 볼륨당 동시 스냅샷의 최대수입니다.

이름	기본값	조정 가능	설명
범용 SSD(gp3) 볼륨당 동시 스냅샷	각 지원되는 지역: 5	아니요	이 지역의 범용 SSD(gp3) 볼륨당 동시 스냅샷의 최대수입니다.
마그네틱(표준) 볼륨당 동시 스냅샷	각 지원되는 지역: 5	아니요	이 지역의 마그네틱(표준) 볼륨당 동시 스냅샷의 최대수입니다.
프로비저닝된 IOPS SSD(io1) 볼륨당 동시 스냅샷	각 지원되는 지역: 5	아니요	이 지역의 프로비저닝된 IOPS SSD(io1) 볼륨당 동시 스냅샷의 최대수입니다.
프로비저닝된 IOPS SSD(io2) 볼륨당 동시 스냅샷	각 지원되는 지역: 5	아니요	이 지역의 프로비저닝된 IOPS SSD(io2) 볼륨당 동시 스냅샷의 최대수입니다.
처리량 최적화 HDD(st1) 볼륨당 동시 스냅샷	각 지원되는 지역: 1	아니요	이 지역의 처리량 최적화 HDD(st1) 볼륨당 동시 스냅샷의 최대수입니다.
빠른 스냅샷 복원	지원되는 각 리전: 5개	예	이 지역에서 빠른 스냅샷 복원에 대해 활성화할 수 있는 스냅샷의 최대수입니다.

이름	기본값	조정 가능	설명
계정당 GetSnapshotBlock 요청 수	us-east-1: 초당 5,000 us-east-2: 초당 5,000 us-west-2: 초당 5,000 ap-southeast-1: 초당 5,000 eu-west-1: 초당 5,000 각각의 지원되는 다른 리전: 초당 1,000	예	계정당 허용되는 GetSnapshotBlock 요청의 최대수입니다.
스냅샷당 GetSnapshotBlock 요청 수	지원되는 각 지역: 초당 1,000개	아니요	스냅샷당 허용되는 GetSnapshotBlock 요청의 최대수입니다.
프로비저닝된 IOPS SSD(io1)에 대한 IOPS 볼륨	지원되는 각 지역: 300,000	예	이 리전의 프로비저닝된 IOPS SSD(io1) 볼륨에 프로비저닝할 수 있는 IOPS의 최대 집계 수입니다.
프로비저닝된 IOPS SSD(io2)에 대한 IOPS 볼륨	지원되는 각 지역: 100,000	예	이 리전의 프로비저닝된 IOPS SSD(io2) 볼륨에 프로비저닝할 수 있는 IOPS의 최대 집계 수입니다.

이름	기본값	조정 가능	설명
프로비저닝된 IOPS SSD(io1)에 대한 IOPS 수정 볼륨	각각 지원되는 지역: 500,000	예	이 지역의 프로비저닝된 IOPS SSD(io1) 볼륨 전체에서 볼륨 수정 시 요청할 수 있는 IOPS의 최대 집계 수입니다.
프로비저닝된 IOPS SSD(io2)에 대한 IOPS 수정 볼륨	지원되는 각 지역: 100,000	예	이 지역의 프로비저닝된 IOPS SSD(io2) 볼륨 전체에서 볼륨 수정 요청의 최대 현재(시작) 및 요청된(끝) IOPS입니다.
계정당 진행 중인 스냅샷 아카이브 수	지원되는 각 지역: 25	예	계정당 진행 중인 스냅샷 아카이브의 최대수입니다.
계정당 아카이브에서 진행 중인 스냅샷 복원 수	지원되는 각 리전: 5개	예	계정당 아카이브에서 진행 중인 스냅샷 복원의 최대 수입니다.
계정당 ListChangedBlocks 요청 수	각각 지원되는 지역: 초당 50개	아니요	계정당 허용되는 ListChangedBlocks 요청의 최대수입니다.
계정당 ListSnapshotBlocks 요청 수	각각 지원되는 지역: 초당 50개	아니요	계정당 허용되는 ListSnapshotBlocks 요청의 최대수입니다.
계정당 최대 동시 복사 볼륨 작업 수	지원되는 각 리전: 5개	예	이 리전의 계정당 허용되는 최대 동시 복사 볼륨 작업 수.

이름	기본값	조정 가능	설명
리전당 동시 볼륨 생성 요청의 볼륨 초기화를 위한 프로비저닝 속도	각 지원되는 리전: 5,000	예	이 리전의 동시 볼륨 생성 요청에서 요청할 수 있는 MiB/s 단위의 최대 집계 볼륨 초기화를 위한 프로비저닝된 속도
계정당 PutSnapshotBlock 요청 수	us-east-1: 초당 5,000 us-east-2: 초당 5,000 us-west-2: 초당 5,000 ap-southeast-1: 초당 5,000 eu-west-1: 초당 5,000 각각의 지원되는 다른 리전: 초당 1,000	예	계정당 허용되는 PutSnapshotBlock 요청의 최대수입니다.
스냅샷당 PutSnapshotBlock 요청 수	지원되는 각 지역: 초당 1,000개	아니요	스냅샷당 허용되는 PutSnapshotBlock 요청의 최대수입니다.
지역당 스냅샷 수	지원되는 각 지역: 100,000	예	리전당 최대 스냅샷 수입니다.

이름	기본값	조정 가능	설명
계정당 StartSnapshot 보류 중인 스냅샷	지원되는 각 리전: 100	아니요	StartSnapshot API를 사용하여 생성할 수 있는 계정당 최대 보류 중 스냅샷 수.
계정당 StartSnapshot 요청 수	지원되는 지역별: 초당 10개	아니요	계정당 허용되는 StartSnapshot 요청의 최대수입니다.

이름	기본값	조정 가능	설명
콜드 HDD(sc1) 볼륨의 스토리지(단위: TiB)	us-east-1: 50 us-east-2: 50 us-west-1: 50 us-west-2: 50 ap-northeast-1: 50 ap-northeast-2: 50 ap-south-1: 50 ap-southeast-1: 50 ap-southeast-2: 50 ca-central-1: 50 eu-central-1: 50 eu-north-1: 50 eu-west-1: 50 eu-west-2: 50 eu-west-3: 50 sa-east-1: 50 각각의 지원되는 다른 리전: 300	예	이 지역의 콜드 HDD(sc1) 볼륨 전체에 프로비저닝할 수 있는 스토리지의 최대 집계량(단위: TiB)입니다.

이름	기본값	조정 가능	설명
범용 SSD(gp2) 볼륨을 위한 스토리지(TiB)	us-east-1: 50 us-east-2: 50 us-west-1: 50 us-west-2: 50 ap-northeast-1: 50 ap-northeast-2: 50 ap-south-1: 50 ap-southeast-1: 50 ap-southeast-2: 50 ca-central-1: 50 eu-central-1: 50 eu-north-1: 50 eu-west-1: 50 eu-west-2: 50 eu-west-3: 50 sa-east-1: 50 각각의 지원되는 다른 리전: 300	예	이 지역의 범용 SSD(gp2) 볼륨 전체에 프로비저닝할 수 있는 스토리지의 최대 집계량(단위: TiB)입니다.

이름	기본값	조정 가능	설명
범용 SSD(gp3) 볼륨을 위한 스토리지(TiB)	af-south-1: 300 ap-east-1: 300 ap-east-2: 300 ap-northeast-3: 300 ap-south-2: 300 ap-southeast-4: 300 ap-southeast-5: 300 ap-southeast-6: 300 ap-southeast-7: 300 ca-west-1: 300 eu-central-2: 300 eu-south-1: 300 eu-south-2: 300 il-central-1: 300 mx-central-1: 300	예	이 지역의 범용 SSD(gp3) 볼륨 전체에 프로비저닝할 수 있는 스토리지의 최대 집계량(단위: TiB)입니다.

이름	기본값	조정 가능	설명
	각각의 지원되는 다른 리전: 50		
마그네틱(표준) 볼륨의 스토리지(단위: TiB)	us-east-1: 50 us-east-2: 50 us-west-1: 50 us-west-2: 50 ap-northeast-1: 50 ap-northeast-2: 50 ap-south-1: 50 ap-southeast-1: 50 ap-southeast-2: 50 ca-central-1: 50 eu-central-1: 50 eu-north-1: 50 eu-west-1: 50 eu-west-2: 50 eu-west-3: 50 sa-east-1: 50 각각의 지원되는 다른 리전: 300	예	이 지역의 마그네틱(표준) 볼륨 전체에 프로비저닝할 수 있는 스토리지의 최대 집계량(단위: TiB)입니다.

이름	기본값	조정 가능	설명
프로비저닝된 IOPS SSD(io1) 볼륨의 스토리지(단위: TiB)	us-east-1: 50 us-east-2: 50 us-west-1: 50 us-west-2: 50 ap-northeast-1: 50 ap-northeast-2: 50 ap-south-1: 50 ap-southeast-1: 50 ap-southeast-2: 50 ca-central-1: 50 eu-central-1: 50 eu-north-1: 50 eu-west-1: 50 eu-west-2: 50 eu-west-3: 50 sa-east-1: 50 각각의 지원되는 다른 리전: 300	예	이 지역의 프로비저닝된 IOPS SSD(io1) 볼륨 전체에 프로비저닝할 수 있는 스토리지의 최대 집계량(단위: TiB)입니다.

이름	기본값	조정 가능	설명
프로비저닝된 IOPS SSD(io2) 볼륨의 스토리지(단위: TiB)	각 지원되는 지역: 20	예	이 지역의 프로비저닝된 IOPS SSD(io2) 볼륨 전체에 프로비저닝할 수 있는 스토리지의 최대 집계량(단위: TiB)입니다.

이름	기본값	조정 가능	설명
처리량 최적화 HDD(st1) 볼륨의 스토리지(단위: TiB)	us-east-1: 50 us-east-2: 50 us-west-1: 50 us-west-2: 50 ap-northeast-1: 50 ap-northeast-2: 50 ap-south-1: 50 ap-southeast-1: 50 ap-southeast-2: 50 ca-central-1: 50 eu-central-1: 50 eu-north-1: 50 eu-west-1: 50 eu-west-2: 50 eu-west-3: 50 sa-east-1: 50 각각의 지원되는 다른 리전: 300	예	이 지역의 처리량 최적화 HDD(st1) 볼륨 전체에 프로비저닝할 수 있는 스토리지의 최대 집계량(단위: TiB)입니다.

이름	기본값	조정 가능	설명
콜드 HDD(sc1) 볼륨의 스토리지 수정(단위: TiB)	각 지원되는 지역: 500	예	이 지역의 콜드 HDD(sc1) 볼륨 전체에 볼륨 수정 시 요청할 수 있는 스토리지의 최대 집계량(단위: TiB)입니다.
범용 SSD(gp2) 볼륨의 스토리지 수정(단위: TiB)	각 지원되는 지역: 500	예	이 지역의 범용 SSD(gp2) 볼륨 전체에 볼륨 수정 시 요청할 수 있는 스토리지의 최대 집계량(단위: TiB)입니다.
범용 SSD(gp3) 볼륨의 스토리지 수정(단위: TiB)	각 지원되는 지역: 500	예	이 지역의 범용 SSD(gp3) 볼륨 전체에 볼륨 수정 시 요청할 수 있는 스토리지의 최대 집계량(단위: TiB)입니다.
마그네틱(표준) 볼륨의 스토리지 수정(단위: TiB)	각 지원되는 지역: 500	예	이 지역의 마그네틱(표준) 볼륨 전체에 볼륨 수정 시 요청할 수 있는 스토리지의 최대 집계량(단위: TiB)입니다.
프로비저닝된 IOPS SSD(io1) 볼륨의 스토리지 수정(단위: TiB)	각 지원되는 지역: 500	예	이 지역의 프로비저닝된 IOPS SSD(io1) 볼륨 전체에 볼륨 수정 시 요청할 수 있는 스토리지의 최대 집계량(단위: TiB)입니다.

이름	기본값	조정 가능	설명
프로비저닝된 IOPS SSD(io2) 볼륨의 스토리지 수정(단위: TiB)	각 지원되는 지역: 20	예	이 지역의 프로비저닝된 IOPS SSD(io2) 볼륨 전체에 볼륨 수정 시 요청할 수 있는 스토리지의 최대 집계량(단위: TiB)입니다.
처리량 최적화 HDD(st1) 볼륨의 스토리지 수정(단위: TiB)	각 지원되는 지역: 500	예	이 지역의 처리량 최적화 HDD(st1) 볼륨 전체에 볼륨 수정 시 요청할 수 있는 스토리지의 최대 집계량(단위: TiB)입니다.
대상 리전당 시간 기반 스냅샷 복사 처리량	지원되는 각 리전: 2,000	예	대상 리전당 시간 기반 스냅샷 복사 작업에 대한 MiB/s 단위의 최대 계정 수준 처리량.

고려 사항

- 할당량은 시간 경과에 따라 변경될 수 있습니다. Amazon EBS는 각 지역 내에서 프로비저닝된 스토리지 및 IOPS 사용량을 지속적으로 모니터링하며 사용량에 따라 지역별로 할당량을 자동으로 늘릴 수 있습니다. Amazon EBS에서는 사용량에 따라 자동으로 할당량이 증가하지만, 필요하면 할당량 증가를 요청할 수 있습니다. 예를 들어 미국 동부(버지니아 북부)에서 현재 할당량보다 많은 gp3 스토리지를 사용할 예정이라면 계획한 사용량에 도달하기 전에 해당 리전의 해당 볼륨 유형에 대한 할당량 증가를 요청할 수 있습니다.
- 지역당 동시 스냅샷 복사본의 할당량은 Service Quotas를 사용하여 조정할 수 없습니다. 그러나 AWS Support에 문의하여이 할당량 증가를 요청할 수 있습니다.
- IOPS 수정 및 스토리지 수정 할당량은 동시에 수정될 수 있는 볼륨의 집계된 현재 값(할당량에 따라 크기 또는 IOPS)에 적용됩니다. 현재 값(크기 또는 IOPS)을 할당량에 합친 볼륨에 대해 동시에 수정 요청을 할 수 있습니다. 예컨대, 프로비저닝된 IOPS SSD(io1) 볼륨의 IOPS 수정 할당량이 `50,000`인 경우, 현재 합산 IOPS가 같거나 `50,000`보다 작으면 `io1` 볼륨 수에 상관없이 동시 IOPS

수정 요청을 할 수 있습니다. 각각 20,000 IOPS로 프로비저닝된 io1 볼륨이 3개 있는 경우 두 볼륨에 대한 IOPS 수정을 동시에(20,000 * 2 < 50,000) 요청할 수 있습니다. 세 번째 볼륨에 대한 동시 IOPS 수정 요청을 제출하면 할당량을 초과하여 해당 요청이 실패합니다(20,000 * 3 > 50,000).

- Amazon EBS에는 인스턴스 시작 요청당 EBS 볼륨 수에 대해 다음과 같은 조정 불가능한 한도가 있습니다.
  - 2500 – us-east-1, us-west-2, eu-west-1, ap-northeast-1
  - 500 - 다른 모든 리전

  이 제한은 사용자가 수행하는 인스턴스 시작 요청과 사용자를 대신하여 Amazon EMR과 같은 AWS 서비스가 수행하는 인스턴스 시작 요청에 적용됩니다. 이 제한을 초과하여 인스턴스 시작 요청이 실패하는 경우 시작 요청에서 EBS 볼륨 구성을 조정하여 볼륨 수를 제한 미만으로 줄입니다. 또는 기술 계정 관리자(TAM)와 협력하여 클러스터를 시작하기 위한 다른 옵션을 살펴보세요.

Amazon EBS 사용 설명서에 대한 문서 이력

다음 표에서는 Amazon EBS에 대한 설명서 릴리스를 설명합니다.

변경 사항	설명	날짜
탄력적 볼륨 수정 업데이트	이제 이전 볼륨 수정이 완료되는 즉시 24시간 내에 최대 4회까지 새 탄력적 볼륨 수정을 시작할 수 있습니다.	2026년 1월 15일
EBS 볼륨용 휴지통	휴지통을 사용하면 실수로 삭제되지 않도록 Amazon EBS 볼륨을 보호할 수 있습니다.	2025년 11월 20일
EBS 볼륨 복사	동일한 가용 영역 내에서 Amazon EBS 볼륨에 대한 즉석 특정 시점 복사본을 생성할 수 있습니다.	2025년 10월 14일
gp3 볼륨 업데이트	이제 최대 64TiB의 크기, 최대 80,000 IOPS, 최대 2,000MiB/s의 처리량으로 범용 SSD(gp3) 볼륨을 프로비저닝할 수 있습니다.	2025년 9월 26일
io2 볼륨에 대한 지연 시간 지침 업데이트	io2 Block Express 볼륨은 16KiB I/O 작업에 500마이크로초 미만의 평균 지연 시간을 제공하도록 설계되었습니다. io2 또한 Block Express 볼륨은 범용 볼륨에 비해 더 나은 이상 지연 시간을 제공하여 800마이크로초를 초과하는 I/O의 빈도를 10배 이상 줄여 줍니다.	2025년 9월 26일
지연 시간 주입	AWS FIS 지연 시간 주입 작업을 사용하여 Amazon EBS 볼	2025년 9월 16일

	륨에서 늘어난 I/O 지연 시간을 시뮬레이션합니다.	
io2 리전 확장	이제 모든 상용 및 AWS GovCloud (US) 리전에서 프로비저닝된 IOPS SSD(io2) 볼륨을 사용할 수 있습니다.	2025년 7월 22일
로컬 영역의 로컬 스냅샷	이제 로컬 영역에서 로컬 스냅샷을 생성할 수 있습니다.	2025년 5월 15일
볼륨 초기화를 위한 Amazon EBS 프로비저닝 속도	이제 Amazon EBS 볼륨을 생성할 때 볼륨 초기화 속도를 지정하여 예측 가능한 시간 내에 볼륨이 완전히 초기화되도록 할 수 있습니다.	2025년 5월 6일
Amazon Data Lifecycle Manager VPC 엔드포인트	인터페이스 VPC 엔드포인트를 만들어 VPC와 Amazon Data Lifecycle Manager 간에 프라이빗 연결을 설정할 수 있습니다.	2025년 2월 28일
시간 기반 AMI 복사	이제 EBS 지원 AMI 복사 작업에 대한 완료 기간을 요청하여 AMI 복사본이 특정 기간 내에 완료되도록 할 수 있습니다.	2025년 2월 25일
전체 스냅샷 크기	이제 Amazon EC2 콘솔 및를 사용하여 Amazon EBS 스냅샷의 전체 크기를 볼 수 있습니다 AWS CLI.	2025년 2월 11일
Amazon Data Lifecycle Manager IPv6 지원	Amazon Data Lifecycle Manager는 이제 IPv4와 IPv6 트래픽을 모두 지원하는 듀얼 스택 엔드포인트를 제공합니다.	2025년 2월 7일

휴지통 IPv6 지원	휴지통은 이제 IPv4 트래픽과 IPv6 트래픽을 모두 지원합니다.	2024년 12월 19일
전용 로컬 영역의 로컬 스냅샷	이제 전용 로컬 영역에서 로컬 스냅샷을 생성할 수 있습니다.	2024년 12월 16일
AWSDataLifecycleManagerServiceRole AWS 관리형 정책 업데이트	AWSDataLifecycleManagerServiceRole AWS 관리형 정책이 `ec2:DescribeAvailabilityZones` 작업에 대한 권한을 포함하도록 업데이트되었습니다.	2024년 12월 16일
EBS 스냅샷의 퍼블릭 액세스 차단에 대한 선언적 정책	이제 선언적 정책을 사용하여 여러 리전과 계정에서 스냅샷의 퍼블릭 액세스 차단에 대한 계정 수준 설정을 동시에 적용할 수 있습니다. 자세한 내용은 AWS Organizations 사용 설명서의 선언적 정책을 참조하세요.	2024년 12월 1일
시간 기반 스냅샷 복사본	이제 스냅샷 복사 작업에 대한 완료 기간을 요청하여 스냅샷 복사가 특정 기간 내에 완료되도록 할 수 있습니다.	2024년 11월 26일
휴지통에 대한 제외 태그	이제 리전 수준 보존 규칙에 제외 태그를 추가하여 특정 태그가 있는 리소스를 제외할 수 있습니다.	2024년 11월 19일
CloudFormation 휴지통 지원	이제를 사용하여 휴지통 보존 규칙을 생성하고 관리할 수 있습니다 AWS CloudFormation.	2024년 11월 18일

Amazon EBS 세부 성능 통계	Amazon EBS NVMe 블록 디바이스는 Nitro 기반 Amazon EC2 인스턴스에 연결된 Amazon EBS 볼륨에 대한 실시간의 고해상도 I/O 성능 통계를 제공합니다.	2024년 11월 12일
Amazon EBS 볼륨에 대한 새로운 CloudWatch 지표	이제 볼륨 성능을 모니터링하기 위해 `VolumeAvgReadLatency` , `VolumeAvgWriteLatency` , `VolumeIOPSExceededCheck` 및 `VolumeThroughputExceededCheck` Amazon CloudWatch 지표를 사용할 수 있습니다.	2024년 10월 30일
여러 계정에서 Amazon Data Lifecycle Manager 기본 정책 활성화	CloudFormation StackSets를 사용하여 AWS 조직 전체 또는 특정 AWS 계정에서 Amazon Data Lifecycle Manager 기본 정책을 활성화할 수 있습니다.	2024년 4월 26일
AWSDataLifecycleManagerSSMFullAccess AWS 관리형 정책	`AWSSystemsManagerSAP-CreateDLMSnapshotForSAPHANA` SSM 문서를 사용하여 SAP HANA에 대해 애플리케이션에 일관되게 적용되는 스냅샷을 지원하도록 정책을 업데이트했습니다.	2023년 11월 17일
VolumeStalledIOCheck 지표	`VolumeStalledIOCheck` 지표를 사용하여 볼륨이 마지막 1분 동안 멈춘 IO 검사를 통과했는지 아니면 실패했는지 확인할 수 있습니다.	2023년 11월 16일

Amazon Data Lifecycle Manager 기본 정책	이제 EBS 스냅샷 및 EBS 지원 AMI에 대한 Amazon Data Lifecycle Manager 기본 정책을 생성하여 리전의 모든 볼륨과 인스턴스를 백업할 수 있습니다.	2023년 11월 16일
Amazon EBS 스냅샷 잠금	Amazon EBS 스냅샷을 잠가서 우발적이거나 악의적인 삭제로부터 스냅샷을 보호하거나 특정 기간 동안 WORM 형식으로 저장할 수 있습니다.	2023년 11월 15일
스냅샷에 대한 퍼블릭 액세스 차단	이제 스냅샷에 대한 퍼블릭 액세스 차단을 사용하여 스냅샷의 퍼블릭 공유를 방지할 수 있습니다.	2023년 11월 9일
Amazon Data Lifecycle Manager 사전 및 사후 스크립트	이제 Amazon Data Lifecycle Manager 스냅샷 정책에서 사전 및 사후 스크립트를 사용하여 애플리케이션에 일관되게 적용되는 스냅샷의 수명 주기를 자동화할 수 있습니다.	2023년 11월 7일
NVMe 예약	다중 연결 지원 `io2` 볼륨은 업계 표준 스토리지 펜싱 프로토콜 세트인 NVMe 예약을 지원합니다.	2023년 9월 18일
Amazon EBS에서 오류 테스트	워크로드가 I/O 중단을 처리하는 방법을 테스트 AWS FIS 하기 위해 EBS 볼륨과 연결된 인스턴스 간에 I/O를 일시적으로 중지하는 데 사용합니다.	2023년 1월 27일

휴지통 보존 규칙 잠금	보존 규칙을 잠그면 실수로 인한 또는 악의적인 수정과 삭제를 방지할 수 있습니다.	2022년 11월 23일
휴지통에 사용되는 조건 키	`rbin:Request/ResourceType` 및 `rbin:Attribute/ResourceType` 조건 키를 사용하여 휴지통 요청에 대한 액세스를 필터링할 수 있습니다.	2022년 6월 14일
io2 Block Express 볼륨	io2 Block Express 볼륨의 크기와 프로비저닝된 IOPS를 수정하고 빠른 스냅샷 복원을 위해 활성화할 수 있습니다.	2022년 5월 31일
AMI용 휴지통	휴지통을 사용하면 실수로 삭제된 AMI를 복원할 수 있습니다.	2022년 2월 3일
Amazon EBS 스냅샷용 휴지통	Amazon EBS 스냅샷용 휴지통은 실수로 삭제된 스냅샷을 복원할 수 있는 스냅샷 복구 기능입니다.	2021년 11월 29일
Amazon EBS 스냅샷 아카이브	Amazon EBS 스냅샷 아카이브는 자주 액세스하지 않는 스냅샷을 저렴한 장기 스토리지에 사용할 수 있는 새로운 스토리지 계층입니다.	2021년 11월 29일

Amazon Data Lifecycle Manager의 AMI 사용 중단 지원	Amazon Data Lifecycle Manager EBS 지원 AMI 정책은 AMI를 사용 중단할 수 있습니다. AWSDataLifecycleManagerServiceRoleForAMIManagement AWS 관리형 정책이 이 기능을 지원하도록 업데이트되었습니다.	2021년 8월 23일
Amazon Data Lifecycle Manager에 사용되는 CloudWatch 지표	Amazon CloudWatch를 사용하여 Amazon Data Lifecycle Manager 정책을 모니터링할 수 있습니다.	2021년 7월 28일
EBS 다이렉트 API에 대한 CloudTrail 데이터 이벤트	ListSnapshotBlocks, ListChangedBlocks, GetSnapshotBlock 및 PutSnapshotBlock API는 CloudTrail에서 데이터 이벤트를 기록할 수 있습니다.	2021년 7월 27일
io2 Block Express 볼륨	이제 io2 Block Express 볼륨을 일반적으로 사용할 수 있습니다.	2021년 7월 19일
Outposts의 Amazon EBS 로컬 스냅샷	이제 Outposts의 Amazon EBS 로컬 스냅샷을 사용하여 Outpost의 볼륨 스냅샷을 Outpost 자체의 Amazon S3에 로컬로 저장할 수 있습니다.	2021년 2월 4일
io2 볼륨에 대한 다중 연결 지원	이제 Amazon EBS 다중 연결에 프로비저닝된 IOPS SSD(io2) 볼륨을 사용할 수 있습니다.	2020년 12월 18일

Amazon Data Lifecycle Manager	Amazon Data Lifecycle Manager를 사용하여 스냅샷을 공유하고 AWS 계정 간에 복사하는 프로세스를 자동화합니다.	2020년 12월 17일
gp3 볼륨	새로운 Amazon EBS 범용 SSD 볼륨 유형입니다. 볼륨을 생성하거나 수정할 때 프로비저닝된 IOPS 및 처리량을 지정할 수 있습니다.	2020년 12월 1일
처리량 최적화 HDD 및 콜드 HDD 볼륨 크기	처리량 최적화 HDD(st1) 및 콜드 HDD(sc1) 볼륨의 크기는 125GiB에서 16TiB까지 다양합니다.	2020년 11월 30일
Amazon Data Lifecycle Manager	Amazon Data Lifecycle Manager를 사용하여 EBS-backed AMI의 생성, 보존 및 삭제를 자동화할 수 있습니다.	2020년 11월 9일
Amazon Data Lifecycle Manager	최대 4개의 일정으로 Amazon Data Lifecycle Manager 정책을 구성할 수 있습니다.	2020년 9월 17일
Amazon EBS의 프로비저닝된 IOPS SSD(io2) 볼륨	프로비저닝된 IOPS SSD(io2) 볼륨은 99.999%의 볼륨 내구성을 제공하도록 설계되었으며 AFR은 0.001% 이하입니다.	2020년 8월 24일
빠른 스냅샷 복원	자신에게 공유된 스냅샷에 대해 빠른 스냅샷 복구를 활성화할 수 있습니다.	2020년 7월 21일

Amazon EBS 다중 연결	이제 단일 프로비저닝된 IOPS SSD(IO1) 볼륨을 동일한 가용 영역에 있는 최대 16개의 Nitro 기반 인스턴스에 연결할 수 있습니다.	2020년 2월 14일
Amazon EBS 빠른 스냅샷 복원	EBS 스냅샷에서 빠른 스냅샷 복원을 활성화하여 스냅샷에서 생성된 EBS 볼륨이 생성 시 완전히 초기화되고 프로비저닝된 모든 성능을 즉시 제공할 수 있습니다.	2019년 11월 20일
Amazon EBS 다중 볼륨 스냅샷	EC2 인스턴스에 연결된 여러 EBS 볼륨에서 정확한 특정 시점, 데이터 조정 및 충돌 일치 스냅샷을 생성할 수 있습니다.	2019년 5월 29일
Amazon EBS 암호화 기본 제공	리전에서 기본적으로 암호화를 사용하도록 설정하면 해당 리전에서 생성한 모든 새 EBS 볼륨이 EBS 암호화용 기본 KMS 키를 사용하여 암호화됩니다.	2019년 5월 23일
스냅샷 수명 주기 자동화	Amazon Data Lifecycle Manager를 사용하여 EBS 볼륨의 스냅샷 생성 및 삭제를 자동화할 수 있습니다.	2018년 7월 12일
연결된 EBS 볼륨에 대한 수정 수행	대부분의 EC2 인스턴스에 연결된 대다수 EBS의 경우, 볼륨을 분리하거나 인스턴스를 중지하지 않고도 볼륨 크기, 유형, IOPS를 수정할 수 있습니다.	2017년 2월 13일
간에 암호화된 Amazon EBS 스냅샷 복사 AWS 계정	이제 AWS 계정간에 암호화된 EBS 스냅샷을 복사할 수 있습니다.	2016년 6월 21일

처리량 최적화 HDD 및 콜드 HDD 볼륨 유형	이제 처리량에 최적화된 HDD(st1) 및 콜드 HDD(sc1) 볼륨을 생성할 수 있습니다.	2016년 4월 19일
범용 SSD 볼륨 유형	범용 SSD 볼륨은 광범위한 작업에서 이상적으로 사용될 수 있는 비용 효과적인 스토리지를 제공합니다. 이러한 볼륨은 10밀리초 미만의 지연 시간, 연장된 기간 동안 3,000 IOPS로 버스트할 수 있는 기능 및 3 IOPS/GiB의 기본 성능을 제공합니다. 범용 SSD 볼륨 크기는 1GiB~1TiB입니다.	2014년 6월 16일
Amazon EBS 암호화	Amazon EBS 암호화에서는 EBS 데이터 볼륨 및 스냅샷에 완벽한 암호를 제공하므로 보안 키 관리 인프라를 구축하고 유지 관리할 필요가 없습니다. EBS 암호화는 AWS 관리형 키를 사용하여 데이터를 암호화하여 상주 데이터에 대한 보안을 활성화합니다. EC2 인스턴스를 호스트하는 서버에서 암호화가 이루어지기 때문에 EC2 인스턴스와 EBS 스토리지 사이를 이동하는 데이터도 암호화됩니다.	2014년 5월 21일
증분형 스냅샷 복사본	이제 증분형 스냅샷 사본을 사용할 수 있습니다.	2013년 6월 11일
EBS 스냅샷 복사본	스냅샷 복사본으로 데이터 백업, 새 Amazon EBS 볼륨 또는 Amazon Machine Image(AMI)를 생성할 수 있습니다.	2012년 12월 17일

기계 번역으로 제공되는 번역입니다. 제공된 번역과 원본 영어의 내용이 상충하는 경우에는 영어 버전이 우선합니다.