aws

사용 설명서

Amazon Elastic File System

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

Amazon Elastic File System: 사용 설명서

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

Amazon의 상표 및 브랜드 디자인은 Amazon 외 제품 또는 서비스와 함께, Amazon 브랜드 이미지를 떨어뜨리거나 고객에게 혼동을 일으킬 수 있는 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 Elastic File System이란 무엇입니까? 1
Amazon EFS를 처음 사용하십니까? 2
작동 방식 3
Amazon EFS가 Amazon EC2와 함께 작동하는 방식 4
Amazon EFS와 AWS Direct Connect 및 AWS Managed VPN의 작동 방식 7
Amazon EFS의 작동 방식 AWS Backup 8
특성 9
인증 및 액세스 제어 9
Amazon EFS 데이터 일관성 9
가용성과 내구성 10
복제 16
시작하기 17
사전 조건 17
파일 시스템 생성 및 EC2 인스턴스 시작 17
EFS 파일 시스템으로 파일 전송 18
사전 조건 18
리소스 정리 19
NFS 클라이언트 설치 21
EFS 도구에 대한 종속성 21
지원되는 배포판 22
EFS 클라이언트 수동 설치 24
Amazon EC2 Linux 인스턴스에 Amazon EFS 클라이언트 설치 24
다른 Linux 배포판에 Amazon EFS 클라이언트 설치 25
EC2 Mac 인스턴스에 EFS 클라이언트 설치 25
EFS 클라이언트 자동 설치 26
Amazon EFS 클라이언트가 설치 중에 수행하는 작업 26
Systems Manager 지원 운영 체제 27
EFS 클라이언트를 설치 AWS Systems Manager 하도록 구성 28
`botocore` 설치 및 업그레이드 29
`stunnel` 업그레이드 30
stunnel 설치 관련 문제 해결 31
FIPS 모드 활성 33
리소스 생성 및 관리 35
구현 요약 36

리소스 ID ... 37
생성 토큰 및 멱등성 ... 37
파일 시스템 생성 ... 38
파일 시스템 생성에 필요한 IAM 권한 ... 39
파일 시스템의 구성 옵션 ... 39
파일 시스템 삭제 ... 54
파일 시스템 정책 생성 ... 55
액세스 포인트 생성 ... 57
액세스 포인트 삭제 ... 59
리소스에 태그 지정 ... 60
태그 기본 사항 ... 60
태그 제한 사항 ... 61
액세스 제어에 태그 사용 ... 61
리소스에 태그 지정 ... 62
자습서: 쓰기 가능한 사용자별 하위 디렉터리 생성 ... 63
파일 시스템 탑재 ... 65
Linux의 탑재 고려 사항 ... 65
EFS 탑재 도우미 사용 ... 67
EFS 탑재 도우미에서 사용하는 탑재 설정 ... 69
지원 로그 얻기 ... 69
사전 조건 ... 70
EC2 Linux에서 탑재 ... 72
EC2 Mac에 탑재 ... 73
다른 리전에서 탑재 ... 75
One Zone 파일 시스템 탑재하기 ... 76
IAM 권한 부여를 통한 탑재 ... 79
EFS 액세스 포인트를 사용한 탑재 ... 80
여러 EC2 인스턴스 탑재 ... 81
다른 계정 또는 VPC에서 탑재 ... 82
NFS 사용 ... 85
사전 조건 ... 85
NFS 지원 ... 86
NFS 클라이언트 설치 ... 87
권장 NFS 탑재 설정 ... 89
DNS를 사용하여 EC2에 탑재 ... 90
IP 주소를 사용하여 탑재 ... 93

파일 시스템 자동 탑재 94
새 EC2 Linux 인스턴스 95
기존 EC2 Linux 인스턴스 96
NFS를 사용하는 Linux 및 Mac 인스턴스 100
파일 시스템 탑재 해제 102
자습서:를 사용하여 파일 시스템 생성 및 탑재 AWS CLI 103
사전 조건 104
설정 AWS CLI 105
1단계: EC2 리소스 생성 106
2단계: EFS 리소스 생성 111
3단계: 파일 시스템 탑재 및 테스트 114
4단계: 정리 117
자습서: 온프레미스 클라이언트로 탑재 119
사전 조건 120
1단계: EFS 리소스 생성 122
2단계: NFS 클라이언트 설치 123
3단계: 온프레미스 클라이언트에 EFS 파일 시스템 탑재 124
4단계: 리소스 정리 및 AWS 계정 보호 125
선택 사항: 전송 중 데이터 암호화 126
자습서: 다른 VPC에서 파일 시스템 탑재하기 129
사전 조건 130
1단계: EFS 탑재 대상의 가용 영역 ID 확인 131
2단계: 탑재 대상 IP 주소 확인 131
3단계: 탑재 대상에 대한 호스트 항목 추가 132
4단계: EFS 탑재 도우미를 사용하여 파일 시스템 탑재 133
5단계: 리소스 정리 및 보호 AWS 계정 134
탑재 문제 해결 135
Windows 인스턴스에서 파일 시스템 탑재에 실패 135
서버에 의한 액세스 거부 135
자동 탑재 실패 및 인스턴스 무응답 136
/etc/fstab에 여러 Amazon EFS 파일 시스템을 탑재하는 데 실패 136
탑재 명령에 실패하고 "잘못된 fs 유형" 오류 메시지가 표시됨 137
탑재 명령에 실패하고 "잘못된 탑재 옵션" 오류 메시지가 표시됨 138
액세스 포인트를 사용한 탑재 실패 138
파일 시스템 생성 직후 파일 시스템 탑재에 실패함 139
파일 시스템 탑재가 중단된 후 실패하고 제한 시간 초과 오류가 표시됨 139

DNS 이름을 사용한 파일 시스템 탑재에 실패 139
파일 시스템 탑재에 실패하고 "nfs 응답 없음" 오류가 표시됨 140
탑재 대상 수명 주기가 특정 상태에서 멈춤 140
탑재 대상 수명 주기 상태에 오류가 표시됨 141
탑재가 응답하지 않음 141
탑재된 클라이언트의 연결이 끊깁니다. 142
새로 탑재한 파일 시스템에 대한 작업이 "잘못된 파일 핸들" 오류를 반환함 142
파일 시스템 탑재 해제 실패 142
데이터 전송 144
사용 AWS DataSync 144
사용 AWS Transfer Family 145
Amazon EFS AWS Transfer Family 에서를 사용하기 위한 사전 조건 145
에 대한 EFS 파일 시스템 구성 AWS Transfer Family 146
파일 시스템 관리 151
파일 시스템 상태에 대한 이해 151
탑재 대상 생성 152
탑재 대상 및 가용 영역 152
탑재 대상 생성 155
탑재 대상 삭제 161
탑재 대상 VPC 변경 162
탑재 대상 보안 그룹 변경 163
처리량 관리 164
스토리지 수명 주기 관리 166
수명 주기 관리를 위한 파일 시스템 작업 166
수명 주기 정책 구성 167
모니터링 170
모니터링 도구 171
자동 도구 171
수동 모니터링 도구 171
파일 시스템 측정 172
객체 측정 172
측정된 파일 시스템 크기 173
처리량 측정 175
스토리지 클래스 크기 보기 176
CloudWatch를 사용한 지표 모니터링 177
CloudWatch 지표 178

CloudWatch 지표 액세스 184
CloudWatch 지표 사용 185
CloudWatch 지표와 함께 지표 수식 사용 186
탑재 시도 성공 또는 실패 모니터링 192
경보 생성 194
CloudTrail을 사용하여 API 직접 호출 로깅 195
CloudTrail의 Amazon EFS 정보 195
Amazon EFS 로그 파일 항목 이해 196
파일 시스템 유휴 암호화용 Amazon EFS 로그 파일 항목 203
결제 및 사용 보고서 205
비용 할당 태그 사용 205
사용자 정의 비용 할당 태그 206
결제 보고서 206
사용 보고서 207
결제 및 사용 보고서 이해 208
사용 보고서의 추적 작업 213
성능 사양 214
성능 요약 214
스토리지 클래스 216
성능 모드 216
처리량 모드 217
처리량 모드 선택 217
탄력적 처리량 218
프로비저닝된 처리량 219
처리량 전환 및 프로비저닝량 변경에 대한 제한 사항 221
성능 팁 221
평균 I/O 크기 221
높은 처리량과 IOPS를 요구하는 워크로드 최적화 222
동시 연결 222
요청 모델 222
NFS 클라이언트 탑재 설정 223
소규모 파일 성능 최적화 223
디렉터리 성능 최적화 224
NFS read_ahead_kb 크기 최적화 224
성능 문제 해결 225
EFS 파일 시스템을 생성할 수 없음 226

NFS 파일 시스템에서 허용된 파일에 대한 액세스가 거부되었습니다. 227
Amazon EFS 콘솔에 액세스할 때 발생하는 오류 227
Amazon EC2 인스턴스 중단 227
대용량 데이터를 쓰는 애플리케이션이 중단됨 228
병렬로 많은 파일을 열 때 성능 불량 228
사용자 정의 NFS 설정으로 인해 쓰기 지연 발생 229
Oracle Recovery Manager로 백업을 생성하는 속도가 느림 229
AMI 및 커널 문제 해결 230
소유권을 변경할 수 없음 230
클라이언트 버그로 인해 파일 시스템이 계속해서 작업을 반복함 230
교착 상태에 빠진 클라이언트 231
큰 디렉터리의 파일을 나열하는 데 시간이 오래 걸림 231
데이터 백업 및 복제 232
파일 시스템 백업 232
에서 Amazon EFS를 AWS Backup 사용하는 방법 233
필수 IAM 권한 235
백업 성능 235
자동 백업 관리 236
파일 시스템 복제 237
비용 238
복제 성능 238
필수 IAM 권한 239
새 파일 시스템으로 복제 구성 240
기존 파일 시스템으로 복제 구성 244
AWS 계정 간 복제 249
애플리케이션 세부 정보 보기 255
복제 구성 삭제 259
복제본 사용 261
데이터 보안 262
데이터 보호 263
데이터 암호화 264
인터네트워크 개인 정보 보호 273
ID 및 액세스 관리 274
대상 275
ID를 통한 인증 275
정책을 사용하여 액세스 관리 276

Amazon Elastic File System과 IAM의 작동 방식 ... 278
ID 기반 정책 예시 ... 283
리소스 기반 정책 예제 ... 287
AWS 관리형 정책 ... 290
태그 사용 ... 295
서비스 연결 역할 사용 ... 298
문제 해결 ... 302
파일 시스템 데이터 액세스 제어 ... 304
기본 파일 시스템 정책 ... 304
클라이언트에 대한 EFS 작업 ... 305
클라이언트에 대한 EFS 조건 키 ... 305
파일 시스템 정책 예제 ... 306
규정 준수 확인 ... 306
복원성 ... 306
네트워크 액세스 제어 ... 307
VPC 보안 그룹 사용 ... 307
VPC 엔드포인트 작업 ... 310
NFS 수준 사용자, 그룹 및 권한 ... 311
파일 및 디렉터리 권한 ... 312
예제: EFS 파일 시스템 사용 사례 및 권한 ... 313
파일 시스템 내의 파일 및 디렉터리에 대한 사용자 및 그룹 ID 권한 ... 314
루트 스쿼싱 사용 안 함 ... 315
권한 캐싱 ... 318
파일 시스템 객체 소유권 변경 ... 318
EFS 액세스 포인트 ... 318
액세스 포인트 작업 ... 319
액세스 포인트는 탑재 대상과 함께 작동 ... 319
사용자 자격 증명 적용 ... 320
루트 디렉터리 적용 ... 321
IAM 정책에서 액세스 포인트 사용 ... 322
파일 시스템에 대한 퍼블릭 액세스 차단 ... 324
를 사용하여 퍼블릭 액세스 차단 AWS Transfer Family ... 324
"퍼블릭"의 의미 ... 325
네트워크 격리 ... 328
할당량 ... 329
늘릴 수 있는 Amazon EFS 할당량 ... 329

할당량 증가 요청 ... 332
변경할 수 없는 Amazon EFS 리소스 할당량 ... 332
NFS 클라이언트에 대한 할당량 ... 334
Amazon EFS 파일 시스템 할당량 ... 334
지원되지 않는 NFSv4.0 및 4.1 기능 ... 335
추가 고려 사항 ... 336
할당량과 관련된 파일 작업 오류 문제 해결 ... 337
열린 파일 및 파일 잠금 확인 ... 337
명령에 실패하고 "디스크 할당량이 초과됨" 오류가 표시됨 ... 338
명령에 실패하고 "I/O 오류"가 표시됨 ... 338
명령에 실패하고 "파일 이름이 너무 깊" 오류가 표시됨 ... 339
명령이 실패하고 "File not found(파일을 찾을 수 없음)" 오류가 표시됨 ... 339
명령에 실패하고 "링크가 너무 많음" 오류가 표시됨 ... 339
명령에 실패하고 "파일이 너무 큼" 오류가 표시됨 ... 340
Amazon EFS API ... 341
API 엔드포인트 ... 341
IPv4 엔드포인트 ... 342
이중 스택(IPv4 및 IPv6) 엔드포인트 ... 342
엔드포인트 지정 ... 343
API 버전 ... 343
관련 주제 ... 343
Amazon EFS의 쿼리 API 요청 속도 활용하기 ... 343
폴링 ... 344
재시도 또는 일괄 처리 ... 344
휴면 간격 계산 ... 344
작업 ... 344
CreateAccessPoint ... 346
CreateFileSystem ... 356
CreateMountTarget ... 373
CreateReplicationConfiguration ... 391
CreateTags ... 400
DeleteAccessPoint ... 404
DeleteFileSystem ... 407
DeleteFileSystemPolicy ... 412
DeleteMountTarget ... 416
DeleteReplicationConfiguration ... 421

DeleteTags 425
DescribeAccessPoints 429
DescribeAccountPreferences 434
DescribeBackupPolicy 437
DescribeFileSystemPolicy 441
DescribeFileSystems 447
DescribeLifecycleConfiguration 454
DescribeMountTargets 459
DescribeMountTargetSecurityGroups 466
DescribeReplicationConfigurations 471
DescribeTags 476
ListTagsForResource 482
ModifyMountTargetSecurityGroups 487
PutAccountPreferences 493
PutBackupPolicy 497
PutFileSystemPolicy 502
PutLifecycleConfiguration 509
TagResource 519
UntagResource 524
UpdateFileSystem 528
UpdateFileSystemProtection 538
데이터 타입 544
AccessPointDescription 545
BackupPolicy 548
CreationInfo 549
Destination 551
DestinationToCreate 554
FileSystemDescription 557
FileSystemProtectionDescription 562
FileSystemSize 563
LifecyclePolicy 565
MountTargetDescription 567
PosixUser 570
ReplicationConfigurationDescription 572
ResourceIdPreference 574
RootDirectory 575

Tag 577
문서 기록 578
..... dc

Amazon Elastic File System이란 무엇입니까?

Amazon Elastic File System(Amazon EFS)은 완전히 탄력적인 서버리스 파일 스토리지를 제공하므로 스토리지 용량과 성능을 프로비저닝하거나 관리하지 않고도 파일 데이터를 공유할 수 있습니다. Amazon EFS는 애플리케이션을 중단하지 않고 페타바이트급까지 온디맨드 규모로 확장할 수 있도록 구축되었으며 파일을 추가하고 제거할 때 확장 및 축소됩니다. Amazon EFS에는 간단한 웹 서비스 인터페이스가 있으므로 파일 시스템을 빠르고 쉽게 생성하고 구성할 수 있습니다. 모든 파일 스토리지 인프라를 관리하는 서비스이므로 사용자는 복잡한 파일 시스템 구성을 배포, 패치 및 유지 보수하는 번잡함에서 벗어날 수 있습니다.

Amazon EFS에서는 Network File System 버전 4(NFSv4.1 및 NFSv4.0) 프로토콜을 지원하므로 오늘날 사용하는 애플리케이션 및 도구도 Amazon EFS에서 원활하게 작동합니다. Amazon EFS는 Amazon EC2, Amazon ECS, Amazon EKS, AWS Lambda, AWS Fargate 등을 비롯한 대부분의 유형의 Amazon Web Services 컴퓨팅 인스턴스에서 액세스할 수 있습니다.

이 서비스는 확장성과 가용성이 높고 내구성이 높도록 설계되었습니다. Amazon EFS는 가용성 및 내구성 요구 사항을 충족하기 위해 다음과 같은 파일 시스템 유형을 제공합니다.

- 리전(권장) - 리전 파일 시스템(권장)은 동일한 AWS 리전 내에서 지리적으로 분리된 여러 가용 영역에 데이터를 중복 저장합니다. 여러 가용 영역에 데이터를 저장하면 AWS 리전에서 하나 이상의 가용 영역을 사용할 수 없는 경우에도 데이터를 지속적으로 사용할 수 있습니다.
- One Zone - One Zone 파일 시스템은 단일 가용 영역 내에 데이터를 저장합니다. 단일 가용 영역에 데이터를 저장하면 데이터를 지속적으로 사용할 수 있습니다. 하지만 가용 영역의 전체 또는 일부가 손실되거나 손상된 경우는 드물지만 이러한 유형의 파일 시스템에 저장된 데이터가 손실될 수 있습니다.

파일 시스템 유형에 대한 자세한 내용은 EFS 파일 시스템 유형 섹션을 참조하십시오.

Amazon EFS는 광범위한 워크로드에 필요한 처리량, IOPS 및 짧은 지연 시간을 제공하도록 설계되었습니다. EFS 파일 시스템은 페타바이트 규모까지 확장 가능하고, 높은 수준까지 처리량을 끌어 올리고, 컴퓨팅 인스턴스에서 데이터로 대량 병렬 엑세스를 허용합니다. 대부분의 워크로드에는 기본 모드인 범용 성능 모드와 탄력적 처리량 모드를 사용하는 것이 좋습니다.

- 범용 - 범용 성능 모드는 웹 서비스 환경, 콘텐츠 관리 시스템, 홈 디렉터리, 일반 파일 서비스 등 지연 시간에 민감한 애플리케이션에 이상적입니다.
- 탄력적 - 탄력적 처리량 모드는 워크로드 활동의 요구 사항에 맞게 처리량 성능을 자동으로 늘리거나 줄일 수 있도록 설계되었습니다.

EFS 성능 및 처리량 모드에 대한 자세한 내용은 Amazon EFS 성능 사양 단원을 참조하세요.

Amazon EFS는 강력한 데이터 일관성 및 파일 잠금 등 파일 시스템 액세스 의미 체계를 제공합니다. 자세한 내용은 Amazon EFS 데이터 일관성 섹션을 참조하세요. Amazon EFS는 또한 이동식 운영 체제 인터페이스(POSIX) 권한을 통해 파일 시스템에 대한 액세스를 제어할 수 있도록 지원합니다. 자세한 내용은 Amazon EFS에서 데이터 보안 섹션을 참조하세요.

Amazon EFS는 보안 및 규정 준수 요구 사항을 충족하는 데 도움이 되는 인증, 권한 부여 및 암호화 기능을 지원합니다. Amazon EFS는 전송 중 데이터 암호화와 유휴 데이터 암호화라는 두 가지 파일 시스템 암호화를 지원합니다. EFS 파일 시스템을 생성할 때 유휴 상태에서의 암호화를 활성화 할 수 있습니다. 이렇게 할 경우 모든 데이터와 메타데이터가 암호화됩니다. 파일 시스템을 탑재할 시 나중에 전송 중 암호화를 활성화할 수 있습니다. Amazon EFS에 대한 NFS 클라이언트 액세스는 AWS Identity and Access Management(IAM) 정책 및 보안 그룹과 같은 네트워크 보안 정책에 의해 제어됩니다. 자세한 내용은 Amazon EFS의 데이터 암호화, Amazon EFS의 자격 증명 및 액세스 관리, NFS 클라이언트용 EFS 파일 시스템에 대한 네트워크 액세스 제어 섹션을 참조하세요.

> **Note**
>
> Microsoft Windows에 기반을 둔 Amazon EC2 인스턴스에서는 Amazon EFS를 사용할 수 없습니다.

Amazon EFS를 처음 사용하십니까?

Amazon EFS를 처음 사용한다면, 다음 섹션을 순서대로 읽어보기를 권장합니다.

1. Amazon EFS 제품 및 요금에 대한 개요는 Amazon EFS를 참조하세요.
2. Amazon EFS 기술 개요는 Amazon EFS 작동 방식를 참조하세요.
3. 시작하기 연습을 시도해 봅니다.

Amazon EFS에 대해 보다 자세히 알고 싶으면 다음 단원을 참조하세요. 이 서비스에 대해 매우 자세한 설명이 나와 있습니다.

- EFS 리소스 생성 및 관리
- EFS 파일 시스템 관리
- Amazon EFS API

Amazon EFS 작동 방식

Amazon Elastic File System(EFS)은 설정하고 잊어버리는 간단한 서버리스 탄력적 파일 시스템을 제공합니다. Amazon EFS를 사용하면 파일 시스템을 생성하고, Amazon EC2 인스턴스 에 파일 시스템을 탑재한 후, 파일 시스템에 데이터를 작성하거나 파일 시스템에서 데이터를 읽을 수 있습니다. Network File System 버전 4.0 및 버전 4.1 프로토콜(NFSv4)을 통해 가상 프라이빗 클라우드(VPC)에 EFS 파일 시스템을 탑재할 수 있습니다. 최신 Amazon Linux, Amazon Linux 2, Red Hat, Ubuntu 및 macOS Big Sur AMI 등에 있는 최신 세대의 Linux NFSv4.1 클라이언트를 EFS 탑재 도우미와 함께 사용하는 것이 좋습니다. 지침은 Amazon EFS 클라이언트 수동 설치 섹션을 참조하세요.

이 프로토콜을 지원하는 Amazon EC2 Linux 및 macOS Amazon Machine Image(AMI) 목록은 NFS 지원 섹션을 참조하세요. 일부 AMI의 경우 Amazon EC2 인스턴스에 파일 시스템을 탑재하려면 NFS 클라이언트를 설치해야 합니다. 지침은 NFS 클라이언트 설치 섹션을 참조하세요.

여러 NFS 클라이언트에서 동시에 EFS 파일 시스템에 액세스할 수 있으므로 단일 연결 이상으로 확장되는 애플리케이션에서도 파일 시스템에 액세스할 수 있습니다. 동일한 내의 여러 가용 영역에서 실행되는 Amazon EC2 및 기타 AWS 컴퓨팅 인스턴스는 파일 시스템에 액세스할 AWS 리전 수 있으므로 많은 사용자가 공통 데이터 소스에 액세스하고 공유할 수 있습니다.

EFS 파일 시스템을 생성할 수 AWS 리전 있는 목록은 섹션을 참조하세요Amazon Web Services 일반 참조.

VPC에서 EFS 파일 시스템에 액세스하려면 VPC에서 탑재 대상을 하나 이상 만듭니다. 탑재 대상은 EFS 파일 시스템을 탑재할 수 있는 NFSv4 엔드포인트의 IP 주소를 제공합니다. EC2 인스턴스와 동일한 가용 영역에 있는 EFS 탑재 대상의 IP 주소로 해석되는 DNS(Domain Name Service) 이름을 사용하여 파일 시스템을 탑재합니다. AWS 리전의 각 가용 영역에 탑재 대상을 하나씩 만들 수 있습니다. VPC의 가용 영역에 서브넷이 여러 개 있는 경우 여러 서브넷 중 하나에만 탑재 대상을 만듭니다. 이렇게 하면 해당 가용 영역의 모든 EC2 인스턴스가 탑재 대상을 공유할 수 있습니다.

> **Note**
>
> EFS 파일 시스템은 한 번에 하나의 VPC에 있는 탑재 대상만 가질 수 있습니다.

탑재 대상 자체는 가용성이 매우 뛰어나게 설계되어 있습니다. 고가용성 및 다른 가용 영역으로의 장애 조치를 설계하는 경우 각 가용 영역에 있는 탑재 대상의 IP 주소와 DNS는 정적이며 여러 리소스가 지원하는 중복 구성 요소입니다. 탑재 대상에 대한 자세한 내용은 탑재 대상 생성 섹션을 참조하세요.

DNS 이름을 사용하여 파일 시스템을 탑재한 후 다른 파일 시스템과 마찬가지로 사용합니다. NFS 수준 권한 및 관련 고려 사항에 대한 자세한 내용은 NFS(Network File System) 수준 사용자, 그룹 및 권한 사용 단원을 참조하세요.

AWS Direct Connect 또는를 사용하여 Amazon VPC에 연결할 때 온프레미스 데이터 센터 서버에 EFS 파일 시스템을 탑재할 수 있습니다 Site-to-Site VPN. EFS 파일 시스템을 온프레미스 서버에 탑재하여 데이터세트를 EFS로 마이그레이션하거나, 클라우드 버스팅 시나리오를 활성화하거나, 온프레미스 데이터를 Amazon EFS에 백업할 수 있습니다.

아래에서 Amazon EFS가 다른 서비스와 연동되는 방식에 대한 설명을 확인할 수 있습니다.

주제

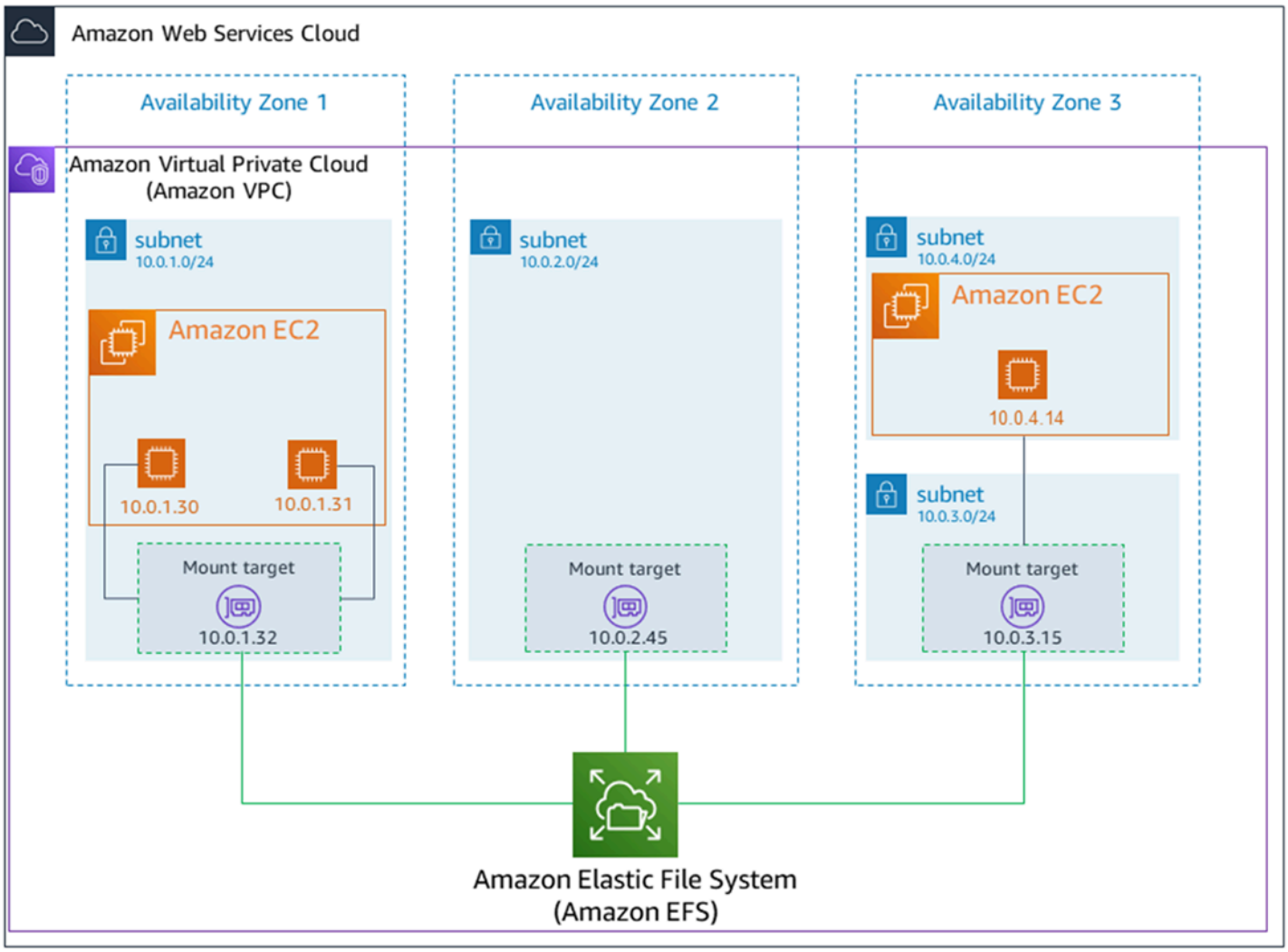
- Amazon EFS가 Amazon EC2와 함께 작동하는 방식
- Amazon EFS와 AWS Direct Connect 및 AWS Managed VPN의 작동 방식
- Amazon EFS의 작동 방식 AWS Backup

Amazon EFS가 Amazon EC2와 함께 작동하는 방식

이 섹션에서는 Amazon EFS Regional 및 One Zone 파일 시스템이 Amazon VPC의 EC2 인스턴스에 탑재되는 방법을 설명합니다.

리전 EFS 파일 시스템

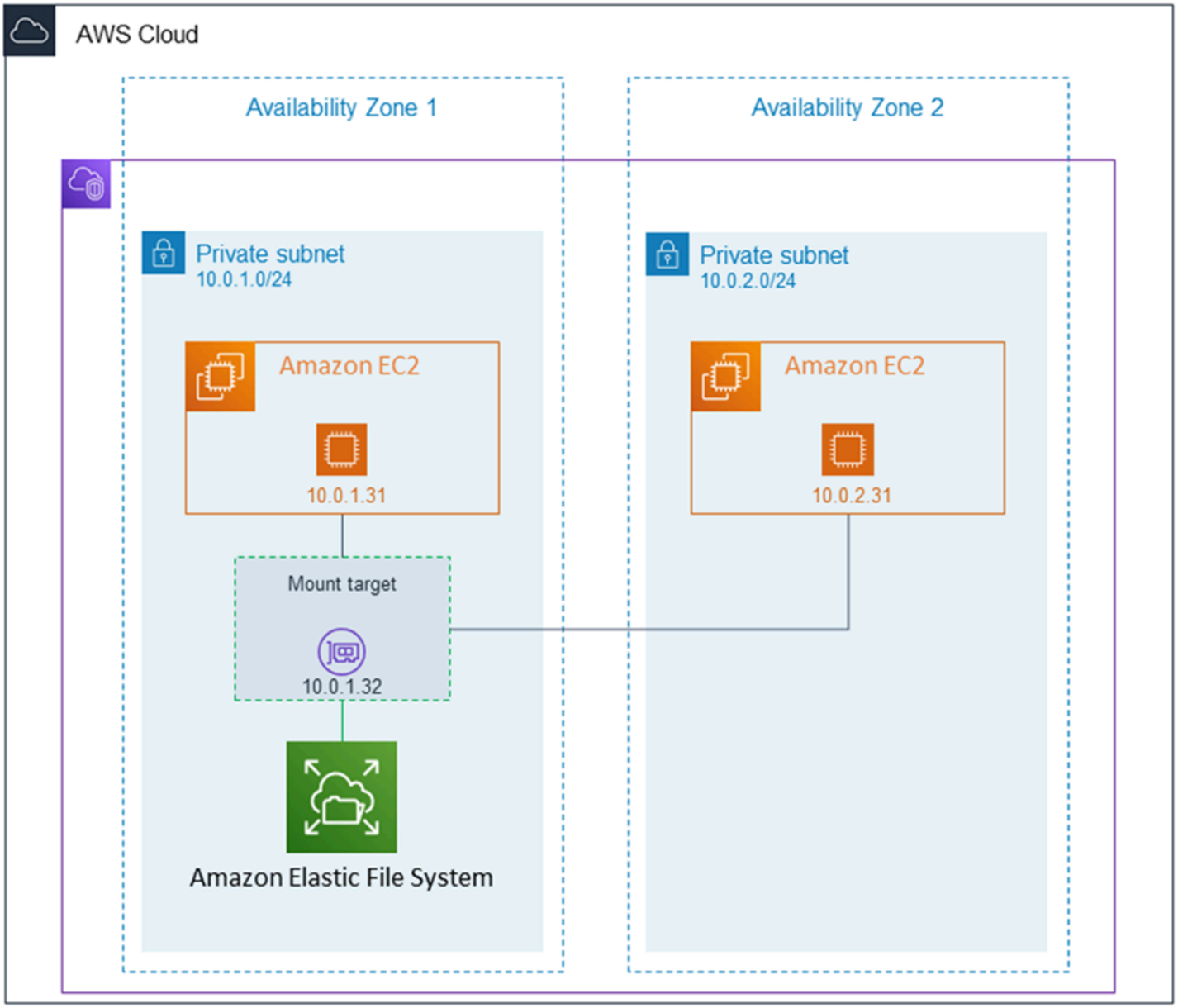
다음 그림은 AWS 리전의 여러 가용 영역에서 Amazon EFS 파일 시스템에 액세스하는 여러 EC2 인스턴스를 보여줍니다.



이 그림에서 Virtual Private Cloud(VPC)에는 가용 영역이 세 개 있습니다. 파일 시스템이 Regional이므로 각 가용 영역에 탑재 대상이 생성되었습니다. 성능 및 비용의 이유로, 동일한 가용 영역 내의 탑재 대상에서 파일 시스템에 액세스하는 것이 좋습니다. 가용 영역 중 하나에는 서브넷이 두 개 있습니다. 그러나 탑재 대상은 두 서브넷 중 하나에서만 생성됩니다. 자세한 내용은 EFS 탑재 도우미를 사용하여 EFS 파일 시스템 탑재 단원을 참조하십시오.

One Zone EFS 파일 시스템

다음 그림은 단일 AWS 리전의 여러 가용 영역에서 One Zone 파일 시스템에 액세스하는 여러 EC2 인스턴스를 보여줍니다.



이 그림에서 VPC에는 각각 서브넷이 하나씩 있는 두 개의 가용 영역이 있습니다. 파일 시스템 유형이 One Zone이기 때문에 탑재 대상을 하나만 가질 수 있습니다. 성능과 비용을 높이려면 파일을 탑재하려는 EC2 인스턴스와 동일한 가용 영역에 있는 탑재 대상에서 파일 시스템에 액세스하는 것이 좋습니다.

이 예시에서는 us-west-2c 가용 영역의 EC2 인스턴스가 다른 가용 영역에 있는 탑재 대상에 액세스하는 데 대해 EC2 데이터 액세스 요금을 지불합니다. 자세한 내용은 One Zone 파일 시스템 탑재하기 단원을 참조하십시오.

Amazon EFS와 AWS Direct Connect 및 AWS Managed VPN의 작동 방식

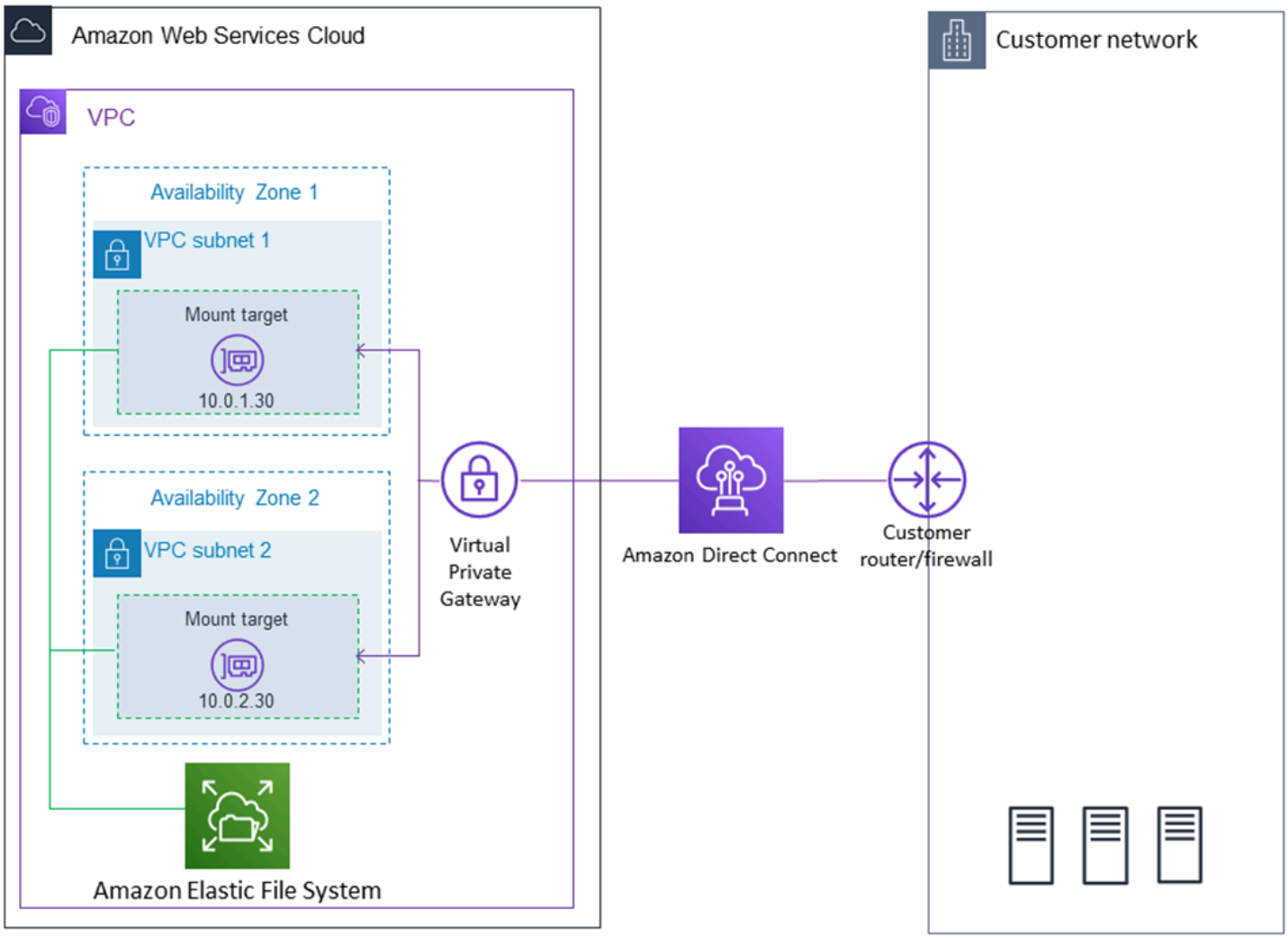
온프레미스 서버에 탑재된 Amazon EFS 파일 시스템을 사용하면 온프레미스 데이터를 Amazon EFS 파일 시스템에서 AWS 클라우드 호스팅되는 로 마이그레이션할 수 있습니다. 버스팅을 이용할 수도 있습니다. 즉, 온프레미스 서버의 데이터를 Amazon EFS로 이동하고 Amazon VPC의 Amazon EC2 인스턴스 플릿에서 데이터를 분석할 수 있습니다. 그런 다음 결과를 파일 시스템에 영구 저장하거나 결과를 온프레미스 서버로 다시 이동할 수 있습니다.

온프레미스 서버에서 Amazon EFS를 사용할 때는 다음 사항에 주의하세요.

- 온프레미스 서버에는 Linux 기반 운영 체제가 필요합니다. Linux 커널 버전 4.0 이상을 사용하는 것이 좋습니다.
- 간소화를 위해 DNS 이름 대신 탑재 대상 IP 주소를 사용하여 온프레미스 서버에서 Amazon EFS 파일 시스템을 탑재하는 것이 좋습니다.

Amazon EFS 파일 시스템에 대한 온프레미스 액세스에는 추가 비용이 들지 않습니다. Amazon VPC Direct Connect 연결에 대한 요금이 부과됩니다. 자세한 내용은 Direct Connect 요금을 참조하십시오.

다음 그림은 온프레미스(온프레미스 서버에는 탑재된 파일 시스템이 있음)에서 Amazon EFS 파일 시스템에 액세스하는 방법의 예가 나와 있습니다.



온프레미스 서버와 VPC 간의 Direct Connect 연결을 사용하여 탑재 대상의 서브넷에 도달할 수 있는 경우 VPC에서 탑재 대상을 사용할 수 있습니다. 온프레미스 서버에서 Amazon EFS에 액세스하려면 온프레미스 서버의 NFS 포트(2049)에 대한 인바운드 트래픽을 허용하는 규칙을 탑재 대상 보안 그룹에 추가합니다. 세부 절차를 포함한 자세한 내용은 사전 조건를 참조하세요.

Amazon EFS의 작동 방식 AWS Backup

파일 시스템에 대한 포괄적인 백업 구현을 위해 Amazon EFS를와 함께 사용할 수 있습니다 AWS Backup. AWS Backup 는 클라우드 및 온프레미스의 서비스 전반에서 데이터 백업을 쉽게 중앙 집중화하고 자동화할 수 있는 완전 관리형 백업 AWS 서비스입니다. AWS Backup를 사용하면 백업 정책을 중앙에서 구성하고 AWS 리소스의 백업 활동을 모니터링할 수 있습니다. Amazon EFS는 항상 백업 작업보다 파일 시스템 작업을 우선시합니다. 를 사용하여 EFS 파일 시스템을 백업하는 방법에 대한 자세한 내용은 섹션을 AWS Backup참조하세요EFS 파일 시스템 백업.

Amazon EFS의 기능

다음은 Amazon EFS의 기능입니다.

주제

- 인증 및 액세스 제어
- Amazon EFS 데이터 일관성
- EFS 파일 시스템의 가용성 및 내구성
- 복제

인증 및 액세스 제어

Amazon EFS 관리 콘솔을 사용하고 파일 시스템 생성과 같은 Amazon EFS API 요청을 수행하려면 유효한 자격 증명이 있어야 합니다. 또한 다른 EFS 및 AWS 리소스를 생성하거나 액세스할 수 있는 권한도 있어야 합니다.

AWS Identity and Access Management (IAM)에서 생성하는 사용자 및 역할에는 리소스를 생성하거나 액세스할 수 있는 권한이 부여되어야 합니다. 권한에 대한 자세한 내용은 Amazon EFS의 자격 증명 및 액세스 관리 섹션을 참조하세요.

NFS(Network File System) 클라이언트에 대한 IAM 권한 부여는 Amazon EFS의 추가 보안 옵션으로, IAM을 사용하여 NFS 클라이언트에 대한 대규모 액세스 관리를 간소화합니다. NFS 클라이언트에 대한 IAM 권한 부여를 사용하면 IAM을 사용하여 기본적으로 확장 가능한 방식으로 EFS 파일 시스템에 대한 액세스를 관리할 수 있습니다. NFS 클라이언트에 대한 IAM 권한 부여는 클라우드 환경에도 최적화되어 있습니다. NFS 클라이언트에 대한 IAM 권한 부여 사용에 대한 자세한 내용은 IAM을 사용하여 파일 시스템에 대한 액세스 제어 섹션을 참조하세요.

Amazon EFS 데이터 일관성

Amazon EFS에서는 애플리케이션이 NFS에서 기대하는 닫기 후 열기(close-to-open) 일관성 의미 체계를 제공합니다.

Amazon EFS에서는 다음과 같은 경우 Regional 파일 시스템의 쓰기 작업이 가용 영역 전체에서 영구적으로 저장됩니다.

- 애플리케이션이 동기식 쓰기 작업을 수행하는 경우(예: `O_DIRECT` 플래그가 지정된 `open` Linux 명령 또는 `fsync` Linux 명령 사용)

- 애플리케이션이 파일을 닫는 경우

Amazon EFS에서는 액세스 패턴에 따라 열기 후 닫기(close-to-open) 의미 체계보다 더 강력한 일관성 보장을 제공할 수 있습니다. 동기식 데이터 액세스를 수행하고 비추가 쓰기를 수행하는 애플리케이션은 데이터 액세스에 대해 읽기 후 쓰기(read-after-write) 일관성을 갖습니다.

파일 잠금

NFS 클라이언트 애플리케이션은 Amazon EFS 파일의 읽기 및 쓰기 작업에 NFS 버전 4 파일 잠금(바이트 범위 잠금 포함)을 사용할 수 있습니다.

Amazon EFS가 파일을 잠그는 방법에 대한 다음 사항을 기억하세요.

- Amazon EFS는 권고 잠금만 지원하므로 읽기/쓰기 작업은 실행 전에 충돌하는 잠금을 확인하지 않습니다. 예를 들어 원자 연산으로 인한 파일 동기화 문제를 방지하려면 애플리케이션이 NFS 시맨틱(예: 개방성에 가까운 일관성)을 알고 있어야 합니다.
- 연결된 모든 인스턴스와 파일에 액세스하는 사용자에 걸쳐 특정 파일 하나에 최대 512개의 잠금이 있을 수 있습니다.

EFS 파일 시스템의 가용성 및 내구성

이 섹션에서는 Amazon Elastic File System(Amazon EFS) 파일 시스템의 파일 시스템 유형 및 스토리지 클래스 옵션에 대해 설명합니다.

EFS 파일 시스템 유형

Amazon EFS는 Regional 및 One Zone 파일 시스템 유형을 제공합니다.

- 리전 - 리전 파일 시스템(권장)은 동일한 AWS 리전내에서 지리적으로 분리된 여러 가용 영역에 데이터를 중복 저장합니다. 여러 가용 영역에 데이터를 저장하면에서 하나 이상의 가용 영역을 사용할 수 없는 경우에도 데이터를 지속적으로 사용할 AWS 리전 수 있습니다.
- One Zone - One Zone 파일 시스템은 단일 가용 영역 내에 데이터를 저장합니다. 단일 가용 영역에 데이터를 저장하면 데이터를 지속적으로 사용할 수 있습니다. 하지만 가용 영역의 전체 또는 일부가 손실되거나 손상된 경우는 드물지만 이러한 유형의 파일 시스템에 저장된 데이터가 손실될 수 있습니다.

  드물지만 AWS 가용 영역의 전체 또는 일부가 손실되거나 손상된 경우 One Zone 스토리지 클래스의 데이터가 손실될 수 있습니다. 예를 들어 화재, 침수 등으로 데이터가 손실될 수 있습니다. 이러

한 유형의 이벤트 외에도 One Zone 스토리지 클래스는 Regional 스토리지 클래스와 유사한 엔지니어링 설계를 사용하여 독립적인 디스크, 호스트 및 랙 수준의 장애로부터 객체를 보호하며, 각각 99.999999999% 의 데이터 내구성을 제공하도록 설계되었습니다.

추가 데이터 보호를 위해 Amazon EFS는 One Zone 파일 시스템을 자동으로 백업합니다 AWS Backup. 파일 시스템 백업을 내의 모든 운영 가용 영역으로 복원 AWS 리전하거나 다른 로 복원할 수 있습니다 AWS 리전. 를 사용하여 생성 및 관리되는 EFS 파일 시스템 백업은 세 개의 가용 영역에 AWS Backup 복제되며 내구성을 위해 설계되었습니다. 자세한 내용은 Resilience in AWS Backup을 참조하세요.

> **Note**
>
> One Zone 파일 시스템은 특정 가용 영역에만 사용할 수 있습니다. One Zone 파일 시스템을 사용할 수 있는 가용 영역이 나열된 표를 보려면 One Zone 파일 시스템에서 지원되는 가용 영역 단원을 참조하세요.

다음 표에서는 가용성, 내구성, 기타 고려 사항 등으로 파일 시스템 유형을 비교합니다.

파일 시스템 유형	다음으로 설계됨	내구성(설계상)	가용성	가용 영역	기타 고려 사항
리전	최고의 내구성과 가용성을 필요로 하는 데이터입니다.	99.999999999%(11 9초)	99.99%	>=3	없음
One Zone	최고의 내구성과 가용성을 필요로 하지 않는 데이터입니다.	99.999999999%(11 9초)	99.99%	1	가용 영역의 손실에 대한 복원력이 없음

One Zone 파일 시스템에서 지원되는 가용 영역

One Zone 파일 시스템은 특정 가용 영역에만 사용할 수 있습니다. 다음 표에는 One Zone 파일 시스템을 사용할 수 있는 각 가용 영역의 AWS 리전 및 AZ IDs가 나열되어 있습니다. 계정의 가용 영역에 대한 AZ IDs의 매핑을 보려면 AWS Resource Access Manager 사용 설명서의 AWS 리소스의 가용 영역 IDs를 참조하세요.

One Zone 파일 시스템을 지원하는 가용 영역

AWS 리전 이름	AWS 리전 코드	지원되는 AZ ID
미국 동부(오하이오)	us-east-2	use2-az1, use2-az2, use2-az3
미국 동부(버지니아 북부)	us-east-1	use1-az1, use1-az2, use1-az4, use1-az5, use1-az6
미국 서부(캘리포니아 북부)	us-west-1	usw1-az1, usw1-az3
미국 서부(오리건)	us-west-2	usw2-az1, usw2-az2, usw2-az3, usw2-az4
Africa (Cape Town)	af-south-1	afs1-az1,afs1-az2,afs1-az3
Asia Pacific (Hong Kong)	ap-east-1	ape1-az1, ape1-az2,ape1-az3
아시아 태평양(뭄바이)	ap-south-1	aps1-az1, aps1-az2, aps1-az3
아시아 태평양(오사카)	ap-northeast-3	apne3-az1, apne3-az2, apne3-az3
아시아 태평양(서울)	ap-northeast-2	apne2-az1, apne2-az2, apne2-az3
아시아 태평양(싱가포르)	ap-southeast-1	apse1-az1, apse1-az2
아시아 태평양(시드니)	ap-southeast-2	apse2-az1, apse2-az2, apse2-az3
아시아 태평양(도쿄)	ap-northeast-1	apne1-az1,apne1-az4
캐나다(중부)	ca-central-1	cac1-az1, cac1-az2
중국(베이징)	cn-north-1	cnn1-az1, cnn1-az2
중국(닝샤)	cn-northwest-1	cnnw1-az1, cnnw1-az2, cnnw1-az3
유럽(프랑크푸르트)	eu-central-1	euc1-az1, euc1-az2, euc1-az3

AWS 리전 이름	AWS 리전 코드	지원되는 AZ ID
유럽(아일랜드)	eu-west-1	euw1-az1, euw1-az2, euw1-az3
Europe (London)	eu-west-2	euw2-az1, euw2-az2
Europe (Milan)	eu-south-1	eus1-az1, eus1-az2, eus1-az3
유럽(파리)	eu-west-3	euw3-az1, euw3-az3
유럽(스톡홀름)	eu-north-1	eun1-az1, eun1-az2, eun1-az3
중동(바레인)	me-south-1	mes1-az1, mes1-az2, mes1-az3
남아메리카(상파울루)	sa-east-1	sae1-az1, sae1-az2, sae1-az3
AWS GovCloud(미국 동부)	us-gov-east-1	usge1-az1, usge1-az2, usge1-az3
AWS GovCloud(미국 서부)	us-gov-west-1	usgw1-az1, usgw1-az2, usgw1-az3

EFS 스토리지 클래스

Amazon EFS의 스토리지 클래스는 사용 사례에 따라 가장 효과적인 스토리지를 제공하도록 설계되었습니다.

- EFS Standard – EFS Standard 스토리지 클래스에서 솔리드 스테이트 드라이브(SSD) 스토리지를 사용하여 자주 액세스하는 파일에 대해 최저 수준의 지연 시간을 제공합니다. 새 파일 시스템 데이터는 먼저 EFS Standard 스토리지 클래스에 기록된 후 수명 주기 관리를 사용하여 EFS Infrequent Access 및 EFS Archive 스토리지 클래스에 계층화할 수 있습니다.
- EFS Inrequent Access (IA) - 분기마다 몇 번만 액세스되는 데이터를 위한 비용 최적화된 스토리지 클래스입니다.
- EFS Archive - 매년 몇 번 또는 그 이하로 액세스되는 데이터를 위한 비용 최적화된 스토리지 클래스입니다.

EFS Archive 스토리지 클래스는 탄력적 처리량을 사용하는 EFS 파일 시스템에서 지원됩니다. 파일 시스템의 Archive 스토리지 클래스에 데이터가 있으면 파일 시스템의 처리량을 버스팅 또는 프로비저닝으로 업데이트할 수 없습니다.

스토리지 클래스 비교

다음 표는 스토리지 클래스를 비교합니다. 각 스토리지 클래스의 성능에 대한 자세한 내용은 Amazon EFS 성능 사양 섹션을 참조하십시오.

스토리지 클래스	다음으로 설계됨	첫 바이트 읽기 지연 시간	내구성(설계상)[1]	가용성 SLA	가용 영역	파일당 최소 청구 요금[2]	최소 스토리지 기간
EFS Standard	1밀리초 미만의 빠른 지연 시간 성능이 필요한 활성 데이터	1밀리초 미만	99.999999999% (11 9초)	99.99%(Regional) 99.9%(One Zone)	=>3 (Regional)	해당 사항 없음	해당 사항 없음
EFS Infrequent Access	분기마다 몇 번만 액세스되는 비활성 데이터입니다.	수십 밀리초			1 (One Zone)	128KiB	해당 사항 없음
EFS Archive	매년 몇 번 또는 그 이하로 액세스하는 비활성 데이터	수십 밀리초		99.9%(Regional)	=>3 (Regional)	128KiB	90일

[1]One Zone 파일 시스템은 단일 AWS 가용 영역에 데이터를 저장하므로 가용 영역 내의 모든 데이터 사본에 영향을 미치는 재해 또는 기타 장애가 발생하거나 가용 영역이 파괴되는 경우 이러한 유형의 파일 시스템에 저장된 데이터가 손실될 수 있습니다.

[2]2023년 11월 26일 오후 12시(PT) 이후 업데이트된 수명 주기 정책은 128KiB 미만의 파일을 IA 클래스로 계층화합니다. Amazon EFS에서 개별 파일 및 메타데이터를 측정하고 요금을 청구하는 방법에 대한 자세한 내용은 Amazon EFS에서 파일 시스템 및 객체 크기를 보고하는 방법 섹션을 참조하십시오.

스토리지 클래스 결제

각 스토리지 클래스의 데이터 양에 대해 요금이 청구됩니다. IA 또는 Archive 스토리지의 파일을 읽을 때 또는 수명 주기 관리를 사용하여 스토리지 클래스 간에 전환하는 데이터에 대해서도 데이터 액세스 요금이 청구됩니다. AWS 청구서에는 각 스토리지 클래스의 용량과 해당 파일 시스템의 스토리지 클래스에 대해 측정된 액세스가 표시됩니다. 자세한 내용은 Amazon EFS 요금을 참조하세요.

또한 Inrequent Access(IA) 및 Archive 스토리지 클래스에는 128KiB의 파일당 최소 청구 요금이 적용됩니다. 128KiB 미만의 파일에 대한 지원은 2023년 11월 26일 오후 12시(PT) 당일 또는 그 이후에 업데이트된 수명 주기 정책에만 사용할 수 있습니다. Amazon EFS에서 개별 파일 및 메타데이터를 측정하고 요금을 청구하는 방법에 대한 자세한 내용은 Amazon EFS에서 파일 시스템 및 객체 크기를 보고하는 방법 섹션을 참조하십시오.

처리량 모드에 따라 추가 결제 고려 사항이 적용됩니다.

- 탄력적 처리량을 사용하는 파일 시스템의 경우 스토리지 요금과 관계없이 파일 시스템에 대해 전송된 메타데이터 및 데이터의 총 월별 양에 대해 요금이 청구됩니다.
- 프로비저닝 처리량 모드를 사용하는 파일 시스템의 경우 EFS Standard 스토리지 클래스에 있는 데이터의 양을 기준으로 제공된 처리량을 초과하여 프로비저닝된 처리량에 대해 요금이 청구됩니다.
- 버스팅 처리량을 사용하는 파일 시스템의 경우 허용된 처리량은 Standard 스토리지 클래스에만 저장된 데이터의 양을 기준으로 결정됩니다.

EFS 처리량 모드에 대한 자세한 내용은 처리량 모드 단원을 참조하세요.

> **Note**
>
> 를 사용하여 수명 주기 관리가 활성화된 EFS 파일 시스템을 백업 AWS Backup 할 때는 데이터 액세스 요금이 발생하지 않습니다. Amazon EFS를 AWS Backup 사용하는에 대한 자세한 내용은 섹션을 참조하세요EFS 파일 시스템 백업.

수명 주기 관리

파일 시스템이 수명 주기 동안 비용 효율적으로 저장되도록 관리하려면 수명 주기 관리를 사용합니다. Lifecycle Management는 파일 시스템에 정의된 수명 주기 구성에 따라 스토리지 클래스 간에 데이터를 자동으로 전환합니다. 수명 주기 구성은 파일 시스템 데이터를 다른 스토리지 클래스로 전환할 시기를 정의하는 일련의 수명 주기 정책입니다. 자세한 내용은 스토리지 수명 주기 관리 단원을 참조하십시오.

복제

복제를 사용하여 기본 설정 AWS 리전 의에서 Amazon EFS 파일 시스템의 복제본을 생성할 수 있습니다. 복제는 EFS 파일 시스템의 데이터와 메타데이터를 선택한에서 생성된 새 대상 EFS 파일 시스템에 자동으로 투명하게 복제 AWS 리전 합니다. EFS는 소스 및 대상 파일 시스템을 자동으로 동기화된 상태로 유지합니다. Replication은 지속적이며 Recovery Point Objective(RPO) 및 Recovery Time Objective(RTO)를 제공하도록 설계되었습니다. 이러한 기능은 규정 준수 및 비즈니스 연속성 목표를 달성하는 데 도움이 됩니다. 자세한 내용은 EFS 파일 시스템 복제 단원을 참조하십시오.

Amazon EFS 시작하기

Amazon Elastic File System(Amazon EFS)을 처음 사용하는 경우 다음 단계에 따라 첫 번째 EFS 파일 시스템을 시작합니다.

1. 시작을 위한 사전 조건 검토
2. EFS 파일 시스템 생성 및 EC2 인스턴스 시작
3. 를 사용하여 EFS 파일 시스템으로 파일 전송 AWS DataSync
4. 리소스 정리 및 AWS 계정 보호

사전 조건

시작하기 단계를 완료하기 전에 다음 요구 사항이 있는지 확인합니다.

- Amazon EC2가 설정되어 있고 EC2 인스턴스 실행에 익숙합니다. 관리자 액세스 권한이 있는 AWS 계정사용자, 키 페어 및 보안 그룹이 필요합니다. 자세한 내용은 Amazon EC2 사용 설명서의 Amazon EC2 시작하기를 참조하세요.
- Amazon Virtual Private Cloud(Amazon VPC), EC2 및 EFS 리소스가 모두 동일한 AWS 리전 에 있으며 리전에 기본 VPC가 있습니다. 기본 VPC가 없거나 기존 또는 새 보안 그룹이 있는 새 VPC에서 파일 시스템을 탑재하려는 경우 VPC 보안 그룹 사용 단원을 참조하세요.
- 기본 보안 그룹에 대한 기본 인바운드 액세스 규칙을 변경하지 않았습니다.

AWS Command Line Interface (AWS CLI) 명령을 사용하여 유사한 시작하기 연습을 수행하여 EFS API를 호출할 수도 있습니다. 자세한 내용은 자습서:를 사용하여 EFS 파일 시스템을 생성하고 EC2 인스턴스에 탑재 AWS CLI 단원을 참조하십시오.

EFS 파일 시스템 생성 및 EC2 인스턴스 시작

이 시작하기 연습을 위한 사전 조건이 충족되었는지 확인한 후 EFS 파일 시스템을 생성하고 EC2 인스턴스를 시작할 수 있습니다. 첫 번째 EFS 파일 시스템을 시작하는 데 필요한 모든 단계를 완료하는 가장 빠른 방법은 인스턴스 시작 중에 EC2 새로운 시작 마법사를 사용하는 것입니다.

> **Note**
>
> Microsoft Windows 기반 EC2 인스턴스에는 Amazon EFS를 사용할 수 없습니다.

EC2 시작 마법사를 사용하여 EFS 파일 시스템을 생성하고 EC2 인스턴스를 시작하려면

EC2 인스턴스 시작을 생성할 때 EFS 파일 시스템을 생성하고 탑재하는 방법에 대한 지침은 Amazon EC2 사용 설명서의 Amazon EC2와 함께 Amazon EFS 사용을 참조하세요.

인스턴스 시작 중에 EFS 파일 시스템을 생성할 때 수행할 단계는 다음과 같습니다.

1. 선택한 키 페어와 네트워크 설정을 사용하여 Linux 운영 체제에서 실행되는 EC2 인스턴스를 생성합니다.
2. 권장 설정이 있고 EC2 인스턴스에 자동으로 탑재되는 공유 EFS 파일 시스템을 생성합니다.
3. EFS 파일 시스템을 파일 전송에 바로 사용할 수 있도록 EC2 인스턴스를 시작합니다.

또는 Amazon EFS 콘솔에서 권장 설정 또는 사용자 지정 설정을 사용하여 파일 시스템을 생성할 수 있습니다. AWS CLI 및 API를 사용하여 파일 시스템을 생성할 수도 있습니다. 파일 시스템 생성을 위한 모든 옵션에 대한 자세한 내용은 Amazon EFS 파일 시스템 생성 단원을 참조하세요.

를 사용하여 EFS 파일 시스템으로 파일 전송 AWS DataSync

EFS 파일 시스템을 생성한 후에는 AWS DataSync를 사용하여 기존 파일 시스템에서 해당 파일 시스템으로 파일을 전송할 수 있습니다. DataSync는 인터넷 또는 Direct Connect를 통해 온프레미스 스토리지 시스템과 AWS 스토리지 서비스 간 데이터 이동 및 복제를 단순화, 자동화 및 가속화하는 데이터 전송 서비스입니다. DataSync는 파일 데이터는 물론 소유권, 타임스탬프, 액세스 권한과 같은 메타데이터도 전송할 수 있습니다.

DataSync에 대한 자세한 내용은 AWS DataSync 단원을 참조하세요.

사전 조건

EFS 파일 시스템으로 파일을 전송하기 전에 다음 사항이 충족되어야 합니다.

- 파일을 전송할 수 있는 소스 NFS 파일 시스템입니다. 이 소스 시스템을 NFS 버전 3, 버전 4 또는 버전 4.1에서 액세스 할 수 있어야 합니다. 파일 시스템의 예로는 온프레미스 데이터 센터에 있는 파일 시스템, 자체 관리형 클라우드 내 파일 시스템 및 EFS 파일 시스템이 있습니다.
- DataSync를 사용하도록 설정되어 있습니다. 자세히 알아보려면 AWS DataSync 사용 설명서의 AWS DataSync시작하기를 참조하세요.

를 사용하여 EFS 파일 시스템으로 파일을 전송하려면 AWS DataSync

DataSync를 사용하여 EFS 파일 시스템으로 파일을 전송하는 방법에 대한 지침은 AWS DataSync 사용 설명서에서 Transferring your data with AWS DataSync를 참조하세요.

DataSync를 사용하여 EFS 파일 시스템으로 파일을 전송할 때 수행할 단계는 다음과 같습니다.

1. EC2 인스턴스에 연결합니다. 자세한 내용은 Amazon EC2 사용 설명서의 EC2 인스턴스에 연결을 참조하세요.
2. 사용자 환경에서 에이전트를 다운로드, 배포 및 활성화합니다.
3. 소스 및 대상 위치를 생성하고 구성합니다.
4. 작업을 생성하고 구성합니다.
5. 작업을 실행하여 소스에서 대상으로 파일 전송.

리소스 정리 및 AWS 계정 보호

이 시작하기 연습을 마치면 다음 단계를 수행하여 리소스를 정리하고 AWS 계정을 보호합니다.

리소스를 정리하고 AWS 계정을 보호하려면

1. EC2 인스턴스에 연결합니다. 자세한 내용은 Amazon EC2 사용 설명서의 EC2 인스턴스에 연결을 참조하세요.
2. 다음 명령을 사용하여 EFS 파일 시스템 탑재를 해제합니다.

   ```
   $ sudo umount efs
   ```

3. Amazon Elastic File System 콘솔(https://console.aws.amazon.com/efs/)을 엽니다.
4. 시작하기 연습의 첫 번째 단계에서 생성한 EFS 파일 시스템을 삭제합니다.

   a. 파일 시스템 목록에서 삭제하려는 EFS 파일 시스템을 선택합니다.
   b. 작업에서 파일 시스템 삭제를 선택합니다.
   c. 영구적으로 파일 시스템 삭제 대화 상자에서 삭제하려는 EFS 파일 시스템에 대한 파일 시스템 ID를 입력한 다음 파일 시스템 삭제를 선택합니다.

5. 이 시작하기 연습을 위해 시작한 EC2 인스턴스를 종료합니다. 자세한 내용은 AWS IAM Identity Center 사용 설명서에서 Amazon EC2 인스턴스 종료를 참조하세요.
6. 이 시작하기 연습을 위해 보안 그룹을 생성한 경우 보안 그룹을 삭제합니다. 자세한 내용은 AWS IAM Identity Center 사용 설명서에서 보안 그룹 삭제를 참조하세요.

> **Warning**
>
> VPC에 대한 기본 보안 그룹은 삭제할 수 없습니다.

Amazon EFS 클라이언트 수동 설치

Amazon EFS용 오픈 소스 도구 모음인 Amazon EFS 클라이언트(amazon-efs-utils)를 설치하는 것이 좋습니다. Amazon EFS 클라이언트에는 EFS 파일 시스템 탑재를 간소화하는 데 도움이 되는 프로그램인 탑재 도우미가 포함되어 있습니다. 또한 이 클라이언트는 Amazon CloudWatch를 사용하여 EFS 파일 시스템의 탑재 상태를 모니터링할 수 있도록 지원하며, Amazon EFS 파일 시스템에 대해 전송 중 데이터의 암호화를 쉽게 수행할 수 있는 도구를 제공합니다.

지원되는 배포판을 실행하는 Amazon EC2(EC2) 인스턴스에 Amazon EFS 클라이언트를 수동으로 설치할 수 있습니다. 지원되는 특정 운영 체제의 경우 패키지를 자동으로 설치하거나 업데이트 AWS Systems Manager 하도록를 구성할 수도 있습니다. 사용할 수 있는 배포 목록은 섹션을 AWS Systems Manager참조하세요Systems Manager 지원 운영 체제.

> **Important**
>
> 모든 기능에 액세스하려면 항상 최신 버전의 amazon-efs-utils를 사용하는 것이 좋습니다. 예를 들어 IPv6 주소를 사용한 탑재는 버전 2.3 이상에서 지원되지만 이전 버전에서는 지원되지 않습니다.

주제

- EFS 도구에 대한 종속성
- 지원되는 배포판
- Amazon EFS 클라이언트 수동 설치
- 를 사용하여 Amazon EFS 클라이언트 자동 설치 또는 업데이트 AWS Systems Manager
- botocore 설치 및 업그레이드
- stunnel 업그레이드
- FIPS 모드 활성

EFS 도구에 대한 종속성

amazon-efs-utils에는 다음 종속성이 있습니다. amazon-efs-utils 패키지를 설치할 때 설치됩니다.

- NFS 클라이언트

  - RHEL, CentOS, Amazon Linux 및 Fedora 배포판용 `nfs-utils`
  - Debian 및 Ubuntu 배포판용 `nfs-common`
- 네트워크 릴레이(stunnel 패키지 버전 4.56 이상)
- Python(버전 3.4 이상)
- OpenSSL 1.0.2 이상

> **Note**
>
> 기본적으로 Amazon EFS 탑재 도우미를 Transport Layer Security(TLS)와 함께 사용할 경우 탑재 도우미는 인증서 호스트 이름 확인을 강제로 수행합니다. EFS 탑재 도우미는 TLS 기능에 `stunnel` 프로그램을 사용합니다. 일부 Linux 버전에는 이 TLS 기능을 기본값으로 지원하는 `stunnel` 버전이 포함되지 않습니다. 이러한 Linux 버전 중 하나를 사용하는 경우 TLS를 사용해서 EFS 파일 시스템을 탑재할 수 없습니다.

`amazon-efs-utils` 패키지를 설치한 후에는 stunnel을 업그레이드합니다. `stunnel` 업그레이드을(를) 참조하세요.

AWS Systems Manager 를 사용하여 Amazon EFS 클라이언트를 관리하고 EC2 인스턴스에 amazon-efs-utils 패키지를 설치하거나 업데이트하는 데 필요한 작업을 자동화할 수 있습니다. 자세한 내용은 를 사용하여 Amazon EFS 클라이언트 자동 설치 또는 업데이트 AWS Systems Manager 단원을 참조하십시오.

암호화 관련 문제는 암호화 문제 해결을 참조하세요.

지원되는 배포판

Amazon EFS 클라이언트는 다음과 같은 Linux 및 Mac 배포판에 대해 검증되었습니다.

배포	패키지 유형	**`init`** 시스템
Amazon Linux 2023(AL2023)	RPM	`systemd`
Amazon Linux 2(AL2)	RPM	`systemd`
CentOS 8	RPM	`systemd`

배포	패키지 유형	**`init`** 시스템
Amazon Linux 1(AL1) 2,017.09 **Note** AL1 AMI는 2023년 12월 31일에 지원이 종료되어 2024년 4월 이후(버전 2.0 이상)에 릴리스된 `amazon-efs-utils` 패키지에는 지원되지 않습니다.	RPM	`upstart`
Debian 11	deb	`systemd`
Fedora 29 - 32	RPM	`systemd`
macOS Big Sur		`launchd`
macOS Monterey		`launchd`
macOS Ventura		`launchd`
macOS Sonoma		`launchd`
OpenSUSE Leap, Tumbleweed	RPM	`systemd`
Oracle8	RPM	`systemd`
Red Hat Enterprise Linux(RHEL) 8, 9	RPM	`systemd`
SUSE Linux Enterprise Server (SLES) 12, 15	RPM	`systemd`
Ubuntu 16.04 LTS, 18.04 LTS, 20.04 LTS, 22.04 LTS	deb	`systemd`

패키지가 검증된 지원되는 배포판의 전체 목록은 Github의 `amazon-efs-utils` README를 참조하세요.

Amazon EFS 클라이언트 수동 설치

Amazon EC2 Linux 인스턴스 및 macOS Big Sur, macOS Monterey 및 macOS Ventura를 실행하는 EC2(EC2) Mac 인스턴스에 Amazon EFS 클라이언트를 수동으로 설치할 수 있습니다. Amazon EFS 클라이언트를 지원하는 배포판 목록은 지원되는 배포판 단원을 참조하세요.

지원되는 운영 체제의 설치 절차는 다음 단원에 설명되어 있습니다.

주제

- Amazon EC2 Linux 인스턴스에 Amazon EFS 클라이언트 설치
- 다른 Linux 배포판에 Amazon EFS 클라이언트 설치
- macOS Big Sur, macOS Monterey 또는 macOS Ventura를 실행하는 EC2 Mac 인스턴스에 Amazon EFS 클라이언트 설치

Amazon EC2 Linux 인스턴스에 Amazon EFS 클라이언트 설치

다음 위치에서 Amazon EC2 Linux 인스턴스에 설치하기 위한 `amazon-efs-utils` 패키지입니다.

- Amazon Linux용 Amazon Machine Image(AMI) 패키지 리포지토리. 다음은 AMI 패키지 리포지토리에서 `amazon-efs-utils` 패키지를 설치하기 위한 지침입니다.
- AWS efs-utils GitHub 리포지토리. GitHub에서 `amazon-efs-utils` 패키지를 설치하는 자세한 방법은 다른 Linux 배포판에 Amazon EFS 클라이언트 설치 단원을 참조하세요.

> **Note**
>
> - 를 사용하는 경우에서 설치 지침을 찾을 Direct Connect수 있습니다사전 조건.
> - Amazon Linux 1(AL1) AMI는 2023년 12월 31일에 지원이 종료되어 2024년 4월 이후(버전 2.0 이상)에 릴리스된 `amazon-efs-utils` 패키지에는 지원되지 않습니다. 2028년까지 장기 지원이 포함된 Amazon Linux 2023(AL2023)으로 애플리케이션을 업그레이드하는 것이 좋습니다.

EC2 Linux 인스턴스에 AMI 패키지 리포지토리의 **`amazon-efs-utils`** 패키지를 설치하는 방법

1. AL2023 또는 Amazon Linux 2(AL2) EC2 인스턴스를 생성했는지 확인합니다. 이 작업을 수행하는 방법은 1단계: 인스턴스 시작을 참조하세요.

2. Secure Shell(SSH)를 통해 인스턴스 터미널에 액세스하고, 적절한 사용자 이름으로 로그인합니다. 자세한 내용은 Amazon EC2 사용 설명서의 EC2 인스턴스에 연결을 참조하세요.
3. 다음 명령을 실행하여 amazon-efs-utils 패키지를 설치합니다.

```
sudo yum install -y amazon-efs-utils
```

다른 Linux 배포판에 Amazon EFS 클라이언트 설치

Amazon Linux AMI 패키지 리포지토리에서 amazon-efs-utils 패키지를 가져오고 싶지 않다면, GitHub에서도 가져올 수 있습니다.

패키지를 복제한 후 Linux 배포판이 지원하는 패키지 유형에 따라 다음 방법 중 하나를 사용해 amazon-efs-utils를 빌드 및 설치할 수 있습니다.

- RPM - 이 패키지 유형은 AL2023, Amazon Linux 2, Red Hat Linux, CentOS 및 이와 유사한 운영체제에서 지원됩니다.
- DEB – Ubuntu, Debian 및 유사한 시스템이 지원하는 패키지 유형입니다.

다른 Linux 배포판용 amazon-efs-utils 패키지 설치에 대한 지침은 Github의 amazon-efs-utils README에서 On other Linux distributions를 참조하세요.

macOS Big Sur, macOS Monterey 또는 macOS Ventura를 실행하는 EC2 Mac 인스턴스에 Amazon EFS 클라이언트 설치

이 amazon-efs-utils 패키지는 macOS Big Sur, macOS Monterey 또는 macOS Ventura를 실행하는 EC2 Mac 인스턴스에 설치할 수 있습니다.

Mac 인스턴스에 amazon-efs-utils 패키지를 설치하는 방법에 대한 지침은 Github의 amazon-efs-utils README에서 On MacOS Big Sur, macOS Monterey, macOS Sonoma and macOS Ventura distribution을 참조하세요.

다음 단계

EC2 인스턴스에 amazon-efs-utils를 설치한 후 파일 시스템을 탑재하기 위한 다음 단계를 진행하세요.

- Amazon CloudWatch를 사용하여 파일 시스템의 탑재 상태를 모니터링할 수 있도록 botocore를 설치하세요.
- 전송 중 데이터를 암호화하려면 stunnel 최신 버전으로 업그레이드하세요.
- EFS 탑재 도우미를 사용하여 파일 시스템 탑재

를 사용하여 Amazon EFS 클라이언트 자동 설치 또는 업데이트 AWS Systems Manager

AWS Systems Manager 를 사용하여 Amazon EFS 클라이언트(amazon-efs-utils)의 관리를 간소화할 수 있습니다. AWS Systems Manager 는 인프라를 보고 제어하는 데 사용할 수 있는 AWS 서비스입니다 AWS. 를 AWS Systems Manager 사용하면 Amazon EC2(EC2) 인스턴스에 amazon-efs-utils 패키지를 설치하거나 업데이트하는 데 필요한 작업을 자동화할 수 있습니다. Distributor 및 상태 관리자와 같은 Systems Manager 기능을 사용하면 다음 프로세스를 자동화할 수 있습니다.

- Amazon EFS 클라이언트에 대한 버전 제어를 유지 관리합니다.
- Amazon EFS 클라이언트를 중앙 집중식으로 저장하고 Amazon EC2 인스턴스에 체계적으로 배포합니다.
- EC2 인스턴스를 정의된 상태로 유지하는 과정을 자동화합니다.

자세한 내용은 AWS Systems Manager 사용 설명서를 참조하십시오.

Amazon EFS 클라이언트가 설치 중에 수행하는 작업

Amazon EFS 클라이언트를 사용하여 파일 시스템 탑재 상태에 대한 Amazon CloudWatch Logs 모니터링을 자동화하고 stunnel을 선택한 Linux 배포판의 최신 버전으로 업그레이드할 수 있습니다. Systems Manager를 사용하여 Amazon EC2 인스턴스에 Amazon EFS 클라이언트를 설치한 경우 다음과 같은 작업이 수행됩니다.

- botocore 설치 및 업그레이드에 설명된 것과 동일한 단계를 사용하여 botocore 패키지를 설치합니다. Amazon EFS 클라이언트는 EFS 파일 시스템 탑재 상태를 모니터링하는 데 botocore를 사용합니다.
- efs-utils.conf 업데이트를 통해 CloudWatch 로그에서 EFS 파일 시스템 탑재 상태를 모니터링할 수 있습니다. 자세한 내용은 탑재 시도 성공 또는 실패 모니터링 단원을 참조하십시오.
- RHEL7 또는 CentOS7 를 실행하는 EC2 인스턴스의 경우 Amazon EFS 클라이언트가 stunnel 업그레이드에 설명된 대로 stunnel을 자동으로 업그레이드합니다. TLS를 사용하여 EFS 파일 시스

템을 성공적으로 탑재하려면 `stunnel` 업그레이드가 필요하며, 이 `stunnel` 버전은 RHEL7과 함께 제공되지만 CentOS7는 Amazon EFS 클라이언트(`amazon-efs-utils`)를 지원하지 않습니다.

Systems Manager 지원 운영 체제

를 사용하여 Amazon EFS 클라이언트를 AWS Systems Manager 자동으로 업데이트하거나 설치하려면 EC2 인스턴스에서 다음 운영 체제 중 하나를 실행해야 합니다.

플랫폼	플랫폼 버전	아키텍처
Amazon Linux 2023(AL2023)	AL2023	x86_64, arm64(Graviton2 이상 프로세서)
Amazon Linux 2(AL2)	2.0	x86_64, arm64(Amazon Linux 2, A1 인스턴스 유형)
Amazon Linux 1(AL1) **Note** Amazon Linux 1(AL1) AMI는 2023년 12월 31일에 지원이 종료되어 2024년 4월 이후(버전 2.0 이상)에 릴리스된 `amazon-efs-utils` 패키지에는 지원되지 않습니다. 2028년까지 장기 지원이 포함된 Amazon Linux 2023(AL2023)으로 애플리케이션을 업그레이드하는 것이 좋습니다.	2017.09, 2018.03	x86_64
CentOS	7, 8	x86_64

플랫폼	플랫폼 버전	아키텍처
Red Hat Enterprise Linux(RHEL)	8, 9	x86_64, arm64
SUSE Linux Enterprise Server(SLES)	12, 15	x86_64
Ubuntu 서버	16.04, 18.04, 20.04	x86_64, arm64(Ubuntu Server 16 이상, A1 인스턴스 유형)

EFS 클라이언트를 설치 AWS Systems Manager 하도록 구성

`amazon-efs-utils` 패키지를 자동으로 설치 또는 업데이트하도록 Systems Manager를 설정하려면 두 가지 일회성 구성이 필요합니다.

1. 필요한 권한으로 AWS Identity and Access Management (IAM) 인스턴스 프로파일을 구성합니다.
2. State Manager의 설치 또는 업데이트에 사용되는 연결(일정 포함)을 구성합니다.

1단계: 필요한 권한으로 IAM 인스턴스 프로파일 구성

기본적으로 AWS Systems Manager 에는 Amazon EFS 클라이언트를 관리하고 amazon-efs-utils 패키지를 설치하거나 업데이트할 수 있는 권한이 없습니다. AWS Identity and Access Management (IAM) 인스턴스 프로파일을 사용하여 Systems Manager 액세스 권한을 부여해야 합니다. 인스턴스 프로파일은 시작할 때 IAM 역할 정보를 Amazon EC2(EC2) 인스턴스에 전달하는 컨테이너입니다.

`AmazonElasticFileSystemsUtils` AWS 관리형 권한 정책을 사용하여 역할에 적절한 권한을 할당합니다. 인스턴스 프로파일의 역할을 새로 생성하거나 기존 역할에 `AmazonElasticFileSystemsUtils` 권한 정책을 추가할 수 있습니다. 그런 다음 이 인스턴스 프로파일을 사용하여 EC2 인스턴스를 시작해야 합니다. 자세한 내용은 Systems Manager에 필요한 인스턴스 권한 구성을 참조하세요.

2단계: State Manager에서 사용하는 연결 구성

이 `amazon-efs-utils` 패키지는 Distributor와 함께 제공되며 관리형 EC2 인스턴스에 바로 배포할 수 있습니다. 설치에 사용할 수 `amazon-efs-utils` 있는 최신 버전의를 보려면 AWS Systems Manager 콘솔 또는 원하는 AWS 명령줄 도구를 사용할 수 있습니다. Distributor에 액세스하려

면 https://console.aws.amazon.com/systems-manager/를 열고 왼쪽 탐색 창에서 Distributor를 선택합니다. 아마존 소유 섹션에서 AmazonEFSUtils를 찾으세요. 패키지 세부 정보를 보려면 AmazonEFSUtils를 선택하세요. 자세한 정보는 패키지 보기를 참조하세요.

State Manager를 사용하면 관리형 EC2 인스턴스에 `amazon-efs-utils` 패키지를 즉시 또는 일정에 따라 설치하거나 업데이트할 수 있습니다. 또한 `amazon-efs-utils`가 새 EC2 인스턴스에 자동으로 설치되도록 할 수 있습니다. Distributor 및 State Manager를 사용하여 패키지를 설치하거나 업데이트하는 방법에 대한 자세한 내용은 Distributor 사용을 참조하세요.

Systems Manager 콘솔을 사용하여 인스턴스에 amazon-efs-utils 패키지를 자동으로 설치 또는 업데이트하려면 패키지 설치 또는 업데이트 예약(콘솔)을 참조하세요. 그러면 인스턴스 세트에 적용할 상태를 정의하는 상태 관리자에 대한 연결을 생성하라는 메시지가 표시됩니다. 연결을 만들 때 다음 입력 내용을 사용하세요.

- 파라미터에 대해 작업 > 설치 및 설치 유형 > 현재 위치 업데이트를 선택합니다.
- 대상에 권장되는 설정은 모든 인스턴스를 선택입니다. 이 설정은 AmazonEFSutils를 자동으로 설치 또는 업데이트하기 위한 대상으로 모든 신규 및 기존 EC2 인스턴스를 등록합니다. 또는 인스턴스 태그를 지정하거나, 인스턴스를 수동으로 선택하거나, 리소스 그룹을 선택하여 인스턴스의 하위 집합에 연결을 적용할 수 있습니다. 인스턴스 태그를 지정하는 경우 태그를 사용하여 EC2 인스턴스를 시작해야 합니다. 그러면 AWS Systems Manager가 Amazon EFS 클라이언트를 자동으로 설치하거나 업데이트할 수 있습니다.
- 일정 지정의 경우 AmazonEFSUtils의 권장 설정은 30일마다입니다. 컨트롤을 사용하여 연결에 대한 cron 또는 rate 일정을 생성합니다.

AWS Systems Manager 를 사용하여 EFS 파일 시스템을 여러 EC2 인스턴스에 탑재하려면 섹션을 참조하세요여러 EC2 인스턴스에 EFS 탑재.

`botocore` 설치 및 업그레이드

Amazon EFS 클라이언트는 `botocore`를 사용하여 다른 AWS 서비스와 상호 작용합니다. CloudWatch Logs에서 EFS 파일 시스템에 대한 탑재 시도 성공 또는 실패를 모니터링하려는 경우에 필요합니다. 자세한 내용은 탑재 시도 성공 또는 실패 모니터링 단원을 참조하십시오.

`botocore` 설치 및 업그레이드에 대한 지침은 Github의 `amazon-efs-utils` README에서 Installing `botocore`를 참조하세요.

stunnel 업그레이드

EFS 탑재 도우미를 사용하여 전송 중 데이터를 암호화하려면 OpenSSL 버전 1.0.2 이상과 온라인 인증서 상태 프로토콜(OCSP) 및 인증서 호스트 이름 확인을 지원하는 stunnel 버전이 필요합니다. EFS 탑재 도우미는 TLS 기능에 stunnel 프로그램을 사용합니다. 일부 Linux 버전에는 이 TLS 기능을 기본값으로 지원하지 않는 stunnel 버전이 포함되어 있습니다. 이런 Linux 배포판 중 하나를 사용하는 경우 TLS를 사용해서 EFS 파일 시스템을 탑재할 수 없습니다.

EFS 탑재 도우미를 설치하면 다음 지침에 따라 시스템의 stunnel 버전을 업그레이드할 수 있습니다.

Amazon Linux, Amazon Linux 2 및 기타 지원되는 Linux 배포판에서 **stunnel**을 업그레이드하려면 (SLES 12 제외)

1. 웹 브라우저에서 stunnel 다운로드 페이지 https://stunnel.org/downloads.html로 이동합니다.
2. tar.gz 형식으로 제공되는 최신 stunnel 버전을 찾습니다. 다음 단계에서 필요하므로 파일 이름을 기록합니다.
3. Linux 클라이언트에서 터미널을 열어 표시된 순서대로 다음 명령을 실행합니다.

   a. RPM의 경우:

   ```
   sudo yum install -y gcc openssl-devel tcp_wrappers-devel
   ```

   DEB의 경우:

   ```
   sudo apt-get install build-essential libwrap0-dev libssl-dev
   ```

   b. *latest-stunnel-version*을 이전 2단계에서 기록한 파일 이름으로 교체합니다.

   ```
   sudo curl -o latest-stunnel-version.tar.gz https://www.stunnel.org/
   downloads/latest-stunnel-version.tar.gz
   ```

   c.
   ```
   sudo tar xvfz latest-stunnel-version.tar.gz
   ```

   d.
   ```
   cd latest-stunnel-version/
   ```

   e.
   ```
   sudo ./configure
   ```

   f.
   ```
   sudo make
   ```

g. 현재 `stunnel` 패키지가 `bin/stunnel`에 설치되어 있습니다. 새 버전을 설치하려면 다음 명령으로 해당 디렉터리를 제거합니다.

```
sudo rm /bin/stunnel
```

h. 최신 버전 설치

```
sudo make install
```

i. 심링크 생성:

```
sudo ln -s /usr/local/bin/stunnel /bin/stunnel
```

macOS에서 stunnel을 업그레이드하려면

- EC2 Mac 인스턴스에서 터미널을 열고 다음 명령을 실행하여 최신 버전의 stunnel로 업그레이드합니다.

```
brew upgrade stunnel
```

SLES 12용 stunnel 업그레이드

- 다음 명령을 실행하고 zypper 패키지 관리자 지침에 따라 SLES12 실행 컴퓨팅 인스턴스에서 stunnel을 업그레이드하세요.

```
sudo zypper addrepo https://download.opensuse.org/repositories/security:Stunnel/
SLE_12_SP5/security:Stunnel.repo
sudo zypper refresh
sudo zypper install -y stunnel
```

필수 기능을 가진 stunnel 버전을 설치한 후에 Amazon EFS 권장 설정의 TLS를 사용해 파일 시스템을 탑재할 수 있습니다.

stunnel 설치 관련 문제 해결

stunnel을 설치할 수 없는 경우 인증서 호스트 이름 확인을 비활성화해 보세요. 또한 온라인 인증서 상태 프로토콜(OCSP)을 활성화하여 가능한 가장 강력한 보안을 제공하세요.

주제

- 인증서 호스트 이름 확인 비활성화
- 온라인 인증서 상태 프로토콜 활성화

인증서 호스트 이름 확인 비활성화

필수 종속성을 설치할 수 없는 경우, Amazon EFS 탑재 도우미 구성 내 인증서 호스트 이름 확인을 선택적으로 비활성화할 수 있습니다. 프로덕션 환경에서는 이 기능을 비활성화하지 않는 것이 좋습니다. 인증서 호스트 이름 확인을 비활성화하려면 다음을 수행하세요.

1. Secure Shell(SSH)를 통해 EC2 인스턴스의 터미널에 액세스하고, 적절한 사용자 이름으로 로그인합니다. 자세한 내용은 Amazon EC2 사용 설명서의 EC2 인스턴스에 연결을 참조하세요.
2. 선택한 텍스트 편집기를 사용해 `/etc/amazon/efs/efs-utils.conf` 파일을 엽니다.
3. `stunnel_check_cert_hostname` 값을 false로 설정합니다.
4. 파일 변경 사항을 저장하고 닫습니다.

전송 중 데이터 암호화 사용에 대한 자세한 내용은 EFS 파일 시스템 탑재 섹션을 참조하세요.

온라인 인증서 상태 프로토콜 활성화

VPC에서 CA에 연결할 수 없는 경우 파일 시스템 가용성을 극대화하기 위해 전송 중 데이터를 암호화하도록 선택하면 OCSP(온라인 인증서 상태 프로토콜)가 기본적으로 활성화되지 않습니다. Amazon EFS는 Amazon 인증 기관(CA)을 사용하여 TLS 인증서를 발급하고 서명하며, CA는 OCSP를 사용하여 해지된 인증서를 확인하도록 클라이언트에 지시합니다. 인증서의 상태를 확인하기 위해 OCSP 엔드포인트는 가상 사설 클라우드에서 인터넷을 통해 액세스할 수 있어야 합니다. 서비스 내에서 Amazon EFS는 인증서 상태를 지속적으로 모니터링하고 감지되는 취소된 인증서를 교체할 새로운 인증서를 발급합니다.

가장 강력한 보안을 제공하기 위해 Linux 클라이언트가 취소된 인증서를 확인할 수 있도록 OCSP를 활성화할 수 있습니다. VPC 내에서 이러한 일이 발생할 가능성은 없지만, OCSP는 해지된 인증서가 악의적으로 사용되지 않도록 보호합니다. EFS TLS 인증서가 취소되면 Amazon이 보안 공지를 게시하고 취소된 인증서를 거부하는 새로운 버전의 EFS 탑재 도우미를 출시합니다.

EFS와의 향후 모든 TLS 연결을 위해 Linux 클라이언트에서 OCSP 활성화

1. Linux 클라이언트에서 터미널을 엽니다.
2. 선택한 텍스트 편집기를 사용해 `/etc/amazon/efs/efs-utils.conf` 파일을 엽니다.

3. `stunnel_check_cert_validity` 값을 true로 설정합니다.
4. 파일 변경 사항을 저장하고 닫습니다.

`mount` 명령의 일부로 OCSP 활성화

- 파일 시스템을 탑재할 때 다음 탑재 명령을 사용하여 OCSP를 활성화합니다.

```
$ sudo mount -t efs -o tls,ocsp fs-12345678:/ /mnt/efs
```

FIPS 모드 활성

운영 체제가 파일 시스템을 탑재할 때 FIPS(Federal Information Processing Standards) 엔드포인트를 사용하는 경우 Amazon EFS 클라이언트에서 FIPS 모드를 활성화해야 합니다. FIPS 모드를 활성화하려면 운영 체제에서 `efs-utils.conf` 파일을 수정해야 합니다.

> **Note**
>
> FIPS 모드에서는 설치된 버전의 OpenSSL을 FIPS로 컴파일해야 합니다. FIPS로 OpenSSL을 구성하는 방법에 대한 자세한 내용은 OpenSSL FIPS README를 참조하세요.

Amazon EFS 클라이언트에서 FIPS 모드를 활성화하는 방법

1. Secure Shell(SSH)를 통해 Amazon EC2 인스턴스의 터미널에 액세스하고, 적절한 사용자 이름으로 로그인합니다. 자세한 내용은 Amazon Elastic Compute Cloud 사용 설명서의 EC2 인스턴스에 연결을 참조하세요.
2. 선택한 텍스트 편집기를 사용해 `/etc/amazon/efs/efs-utils.conf` 파일을 엽니다.
3. 다음 텍스트가 포함된 줄을 찾습니다.

```
"fips_mode_enabled = false"
```

4. 텍스트를 다음으로 변경합니다.

```
"fips_mode_enabled = true"
```

5. 변경 내용을 저장합니다.

EFS 리소스 생성 및 관리

Amazon EFS는 탄력적인 공유 파일 스토리지를 제공합니다. 생성한 파일 시스템은 여러 Amazon EC2 인스턴스에서의 동시 읽기 및 쓰기 액세스를 지원합니다. 파일 시스템은 생성된의 모든 가용 영역에서도 액세스할 AWS 리전 수 있습니다.

네트워크 파일 시스템 버전 4.0 및 버전 4.1 프로토콜(NFSv4)을 사용하여 Amazon VPC를 기반으로 Virtual Private Cloud(VPC)의 EC2 인스턴스에 Amazon EFS 파일 시스템을 탑재할 수 있습니다. 자세한 내용은 Amazon EFS 작동 방식 단원을 참조하십시오.

예를 들어, VPC에서 시작한 EC2 인스턴스가 한 개 이상 있다고 가정해 보겠습니다. 이제 이러한 인스턴스에서 파일 시스템을 만들어 사용하려고 합니다. 다음은 VPC에서 Amazon EFS 파일 시스템을 사용하기 위해 수행해야 하는 일반적인 단계입니다.

- Amazon EFS 파일 시스템 생성 - 파일 시스템을 생성할 때는 이름 태그를 사용하는 것이 좋습니다. 콘솔에 이름 태그 값이 표시되므로 파일 시스템을 쉽게 식별할 수 있습니다. 파일 시스템에 다른 선택적 태그를 추가할 수도 있습니다.
- 파일 시스템의 탑재 대상 생성 - VPC의 파일 시스템에 액세스하고 파일 시스템을 Amazon EC2 인스턴스에 탑재하려면 VPC 서브넷에 탑재 대상을 생성해야 합니다.
- 보안 그룹 생성 - Amazon EC2 인스턴스와 탑재 대상 모두 보안 그룹이 연결되어 있어야 합니다. 보안 그룹은 인스턴스와 탑재 대상 간의 트래픽을 제어하는 가상 방화벽의 역할을 합니다. 탑재 대상과 연결한 보안 그룹을 사용하여 파일 시스템으로의 인바운드 트래픽을 제어할 수 있습니다. 이렇게 하려면 탑재 대상 보안 그룹에 특정 EC2 인스턴스에서의 액세스를 허용하는 인바운드 규칙을 추가하세요. 그런 다음 해당 EC2 인스턴스에서만 파일 시스템을 탑재할 수 있습니다.

주제

- 구현 요약
- 리소스 ID
- 생성 토큰 및 멱등성
- Amazon EFS 파일 시스템 생성
- EFS 파일 시스템 삭제
- 파일 시스템 정책 생성
- 액세스 포인트 생성
- 액세스 포인트 삭제
- EFS 리소스에 태그 지정

- 자습서: 쓰기 가능한 사용자별 하위 디렉터리 생성

구현 요약

Amazon EFS에서는 파일 시스템이 기본 리소스입니다. 각 파일 시스템에는 ID, 생성 토큰, 생성 시간, 파일 시스템 크기(바이트), 파일 시스템에 대해 생성된 탑재 대상 수 및 파일 시스템 수명 주기 정책 등의 속성이 있습니다.

또한 Amazon EFS는 다른 리소스에서 기본 리소스를 구성하도록 지원합니다. 여기에는 탑재 대상 및 액세스 포인트가 포함됩니다.

- 탑재 대상 – 파일 시스템에 액세스하려면 VPC 내에서 탑재 대상을 만들어야 합니다. 각 탑재 대상에는 탑재 대상 ID, 탑재 대상이 생성된 서브넷 ID, 탑재 대상이 생성된 파일 시스템 ID, 파일 시스템을 탑재할 수 있는 IP 주소, VPC 보안 그룹 및 탑재 대상 상태와 같은 속성이 있습니다. `mount` 명령에 IP 주소 또는 DNS 이름을 사용할 수 있습니다.

  각 파일 시스템에는 다음과 같은 형식의 DNS 이름이 있습니다.

  ```
  file-system-id.efs.aws-region.amazonaws.com
  ```

  `mount` 명령에 이 DNS 이름을 지정하여 Amazon EFS 파일 시스템을 탑재할 수 있습니다. EC2 인스턴스 또는 온프레미스 서버의 홈 디렉터리에서 `efs-mount-point` 하위 디렉터리를 만든다고 가정해 보세요. 그런 다음 탑재 명령을 실행하여 파일 시스템을 탑재합니다. 예를 들어 Amazon Linux AMI에서는 다음 `mount` 명령을 사용할 수 있습니다.

  ```
  $ sudo mount -t nfs -o
   nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport file-system-DNS-name:/ ~/efs-mount-point
  ```

  자세한 내용은 탑재 대상 생성 단원을 참조하십시오.
- 액세스 포인트 – 액세스 포인트는 액세스 포인트를 사용하여 이루어진 모든 파일 시스템 요청에 운영 체제 사용자, 그룹 및 파일 시스템 경로를 적용합니다. 액세스 포인트의 운영 체제 사용자 및 그룹은 NFS 클라이언트에서 제공하는 모든 자격 증명 정보를 재정의합니다. 파일 시스템 경로는 액세스 포인트의 루트 디렉터리로 클라이언트에 노출됩니다. 이렇게 하면 공유 파일 기반 데이터 세트에 액세스할 때 각 애플리케이션이 항상 올바른 운영 체제 자격 증명과 올바른 디렉터리를 사용할 수 있습니다. 액세스 포인트를 사용하는 애플리케이션은 자체 디렉터리 및 해당 하위 디렉터리의 데이터에만 액세스할 수 있습니다. 자세한 내용은 액세스 포인트 작업 단원을 참조하십시오.

탑재 대상 및 태그는 파일 시스템과 연관된 하위 리소스입니다. 기존 파일 시스템의 컨텍스트 내에만 생성할 수 있습니다.

Amazon EFS에서는 이러한 리소스를 만들고 관리할 수 있는 API 작업을 제공합니다. 각 리소스에 대한 생성 및 삭제 작업 외에 Amazon EFS에서는 리소스 정보를 검색하기 위한 설명 작업도 가능합니다. 이러한 리소스의 생성 및 관리 옵션은 다음과 같습니다.

- Amazon EFS 콘솔 사용 – 시작하기에서 예제를 참조하세요.
- Amazon EFS 명령줄 인터페이스(CLI) 사용 – 자습서:를 사용하여 EFS 파일 시스템을 생성하고 EC2 인스턴스에 탑재 AWS CLI에서 예제를 참조하세요.
- 또한 다음과 같이 프로그래밍 방식으로 리소스를 관리할 수도 있습니다.
  - AWS SDKs - AWS SDKs는 기본 Amazon EFS API를 래핑하여 프로그래밍 작업을 간소화합니다. 또한 SDK 클라이언트는 사용자가 제공한 액세스 키를 사용하여 요청을 인증합니다. 자세한 내용은 샘플 코드 및 라이브러리 섹션을 참조하세요.
  - 애플리케이션에서 직접 Amazon EFS API 직접 호출 – 어떤 이유로 인해 SDK를 사용할 수 없는 경우, 애플리케이션에서 직접 Amazon EFS API 직접 호출을 할 수 있습니다. 그러나 이 옵션을 사용하는 경우 요청을 인증하려면 필수 코드를 작성해야 합니다. Amazon EFS API에 대한 자세한 내용은 Amazon EFS API 섹션을 참조하세요.

리소스 ID

Amazon EFS는 EFS 리소스를 생성할 때 모든 EFS 리소스에 고유한 리소스 식별자(ID)를 할당합니다. 모든 EFS 리소스 ID는 리소스 식별자와 숫자 0~9와 소문자 a~f의 조합으로 구성됩니다.

2021년 10월 이전에는 새로 생성된 파일 시스템 및 탑재 대상 리소스에 할당된 ID가 하이픈 뒤에 8자 예: `fs-12345678`)를 사용했습니다. 2021년 5월부터 2021년 10월까지는, 하이픈 뒤에 17자(예: `fs-1234567890abcdef0`)를 사용하도록 이러한 리소스 유형의 ID를 변경했습니다. 계정을 만든 시기에 따라 다음 리소스 유형의 리소스가 짧은 ID를 가질 수도 있는데, 이러한 유형의 새 리소스에는 더 긴 ID가 할당됩니다. 리소스 ID는 변경되지 않습니다.

생성 토큰 및 멱등성

멱등성은 API 요청이 한 번만 완료되도록 합니다. 멱등성 요청을 사용하면, 원래 요청이 성공적으로 완료되면 후속 요청이 추가 영향을 미치지 않습니다. 이는 Amazon EFS API와 상호 작용할 때 중복 작업이 생성되는 것을 방지하는 데 유용합니다.

Amazon EFS API는 클라이언트 요청 토큰을 통한 멱등성을 지원합니다. 클라이언트 요청 토큰은 작업 생성 요청을 할 때 지정하는 고유 문자열입니다.

클라이언트 요청 토큰은 최대 64자의 ASCII 문자를 포함하는 모든 문자열이 될 수 있습니다. 요청 성공 후 1분 이내에 클라이언트 요청 토큰을 재사용하는 경우 API는 원래 요청의 작업 세부 정보를 반환합니다.

콘솔을 사용하는 경우, 콘솔에서 토큰이 생성됩니다. 콘솔에서 Custom Create 흐름을 사용하는 경우 자동으로 생성되는 생성 토큰의 형식은 다음과 같습니다.

```
"CreationToken": "console-d215fa78-1f83-4651-b026-facafd8a7da7"
```

Quick Create를 사용하여 서비스 권장 설정으로 파일 시스템을 생성하는 경우 생성 토큰의 형식은 다음과 같습니다.

```
"CreationToken": "quickCreated-d7f56c5f-e433-41ca-8307-9d9c0f8a77a2"
```

Amazon EFS 파일 시스템 생성

시작하기 연습에 설명된 대로 새 EC2 시작 인스턴스를 생성할 때 파일 시스템을 생성할 수 있습니다. 그러나 Amazon EFS 콘솔, AWS Command Line Interface (AWS CLI) 또는 Amazon EFS API를 사용하여 파일 시스템을 생성할 수도 있습니다.

Amazon EFS 콘솔을 사용하여 파일 시스템을 생성할 때 파일 시스템에서 권장 설정을 사용할지 아니면 설정을 사용자 지정할지에 따라 다양한 옵션이 있습니다.

- 빠른 생성을 사용하여 다음과 같은 권장 설정으로 파일 시스템을 빠르게 생성할 수 있습니다.
  - 리전별 가용성
  - 30일 후 파일 시스템을 EFS Infrequent Access(IA) 스토리지로, 90일 후 EFS Archive 스토리지로 전환하고 EFS Standard 스토리지로 전환하지 않는 수명 주기 정책
  - 저장 데이터 암호화 활성화
  - 탄력적 처리량 모드
  - 범용 성능 모드
  - 지정된 서브넷에서 사용 가능한 IPv4를 사용하여 파일 시스템이 생성되는 AWS 리전 의 각 가용 영역에 구성된 대상 탑재
- 사용자 지정을 사용하여 선택한 설정으로 파일 시스템을 생성할 수 있습니다.

파일 시스템 설정과 권장 값을 나열하는 테이블은 파일 시스템의 구성 옵션 섹션을 참조하세요.

파일 시스템 생성에 필요한 IAM 권한

파일 시스템 및 액세스 포인트와 같은 EFS 리소스를 생성하려면 해당 API 작업 및 리소스에 대한 AWS Identity and Access Management (IAM) 권한이 있어야 합니다.

IAM 사용자를 생성하고 사용자 정책에 따라 Amazon EFS 작업에 대한 권한을 부여합니다. 또한 역할을 사용하여 교차 계정 권한을 부여할 수도 있습니다. Amazon Elastic File System은 AWS 서비스 사용자를 대신하여 다른를 호출하는 데 필요한 권한이 포함된 IAM 서비스 연결 역할도 사용합니다. API 작업 권한 관리에 대한 자세한 내용은 Amazon EFS의 자격 증명 및 액세스 관리 섹션을 참조하세요.

파일 시스템의 구성 옵션

EFS 파일 시스템은 다음 테이블에 나열된 설정으로 구성됩니다.

- 빠른 생성을 사용하여 파일 시스템을 생성하는 경우 설정에 대해 표시된 권장 값으로 파일 시스템이 생성됩니다.
- 사용자 지정을 사용하여 사용자 지정 파일 시스템을 생성하는 경우 설정에 대한 권장 값을 변경할 수 있습니다.

EFS 파일 시스템 설정

설정	설명
파일 시스템 유형	권장: 리전별 파일 시스템 유형에 따라 EFS 파일 시스템이 AWS 리전내 데이터를 저장하는 데 사용하는 가용성과 내구성이 결정됩니다. • 리전별 파일 시스템은 AWS 리전내 모든 가용 영역에 걸쳐 데이터와 메타데이터를 중복 저장합니다. AWS 리전내의 각 가용 영역에 탑재 대상을 하나씩 만들 수 있습니다. Regional은 최고 수준의 가용성과 내구성을 제공합니다. • One Zone 파일 시스템은 단일 가용 영역 내에서 데이터와 메타데이터를 중복 저장합니다.

설정	설명
	One Zone 파일 시스템은 단일 탑재 대상만 가질 수 있습니다. 이 탑재 대상은 파일 시스템이 생성된 가용 영역과 동일한 가용 영역에 있어야 합니다. **Note** One Zone 파일 시스템은 특정 가용 영역에만 사용할 수 있습니다. 파일 시스템 유형에 대한 자세한 내용은 EFS 파일 시스템의 가용성 및 내구성 섹션을 참조하십시오.

설정	설명
수명 주기 관리	권장 사항: 30일 후에 IA 스토리지로 전환하고 90일 후에 아카이브 스토리지로 전환합니다. 표준 스토리지로 전환하지 마세요. 수명 주기 관리는 수명 주기 정책을 사용하여 액세스 패턴에 따라 저렴한 Infrequent Access(IA) 스토리지 클래스 안팎으로 파일을 자동으로 이동합니다. 를 사용하여 파일 시스템을 생성하면 AWS Management Console파일 시스템의 수명 주기 정책이 다음과 같은 기본 설정으로 구성됩니다. • IA로의 전환은 마지막 액세스 이후 30일로 설정됩니다. • Archive로 전환은 마지막 액세스 이후 90일로 설정되어 있습니다. • Standard로 전환은 없음으로 설정되어 있습니다. AWS CLI, Amazon EFS API 또는 AWS SDKs를 사용하여 파일 시스템을 생성하는 경우 수명 주기 정책을 동시에 설정할 수 없습니다. 파일 시스템이 생성될 때까지 기다린 다음 PutLifecycleConfiguration API 작업을 사용하여 수명 주기 정책을 업데이트해야 합니다. 수명 주기 관리에 대한 자세한 내용은 스토리지 수명 주기 관리를 참조하세요.

설정	설명
저장된 데이터 암호화	권장: 활성화됨 Amazon EFS는 AWS Key Management Service (AWS KMS) EFS 서비스 키(`aws/elasticfilesystem`)를 사용하여 저장 데이터를 암호화합니다. 암호화를 사용하면 여기에 저장된 모든 데이터와 메타데이터가 암호화됩니다. EFS 파일 시스템을 생성한 후에는 암호화 설정을 변경할 수 없습니다. 암호화되지 않은 파일 시스템을 암호화된 것으로 변환하려면 새 파일 시스템을 생성해야 합니다. Amazon EFS의 암호화에 대한 자세한 내용은 Amazon EFS의 데이터 암호화 섹션을 참조하세요.

설정	설명
처리량 모드	권장: 탄력적 다음 처리량 모드 중에서 선택할 수 있습니다. • 탄력적 - 워크로드의 성능 요구 사항에 맞게 실시간으로 자동으로 확장 및 축소되는 처리량을 제공합니다. • 프로비저닝된 - 파일 시스템의 크기와 무관하게 원하는 처리량을 지정하십시오. • 버스팅 - 파일 시스템의 Standard 스토리지에 있는 데이터 양에 따라 조정되는 처리량을 제공합니다. **Note** 탄력적 처리량 및 프로비저닝 처리량 모드 사용 시 추가 요금이 발생합니다. 자세한 내용은 Amazon EFS 요금을 참조하세요. 처리량 모드에 대한 자세한 내용은 처리량 모드 단원을 참조하세요.

설정	설명
성능 모드	권장 사항: 범용 범용 성능 모드는 작업당 지연 시간이 가장 낮으며 모든 파일 시스템에 권장됩니다. 최대 I/O는 범용 모드보다 긴 지연 시간을 견딜 수 있는 고도로 병렬화된 워크로드용으로 설계된 이전 세대 성능 유형입니다. One Zone 파일 시스템 또는 탄력적 처리량을 사용하는 파일 시스템에서는 최대 I/O 모드가 지원되지 않습니다. **Important** 최대 I/O에서는 작업당 지연 시간이 길어지기 때문에 모든 파일 시스템에 범용 성능 모드를 사용하는 것이 좋습니다. 자세한 내용은 성능 모드 단원을 참조하십시오.

설정	설명
네트워크 액세스	권장 사항: 파일 시스템을 사용할 수 있는 각 가용 영역에서 생성된 탑재 대상 탑재 대상 설정은 다음과 같습니다. • VPC - 탑재 대상은 한 번에 VPC 하나의 파일 시스템에 대해 여러 개를 만들 수 있습니다. 기본적으로 EC2 인스턴스의 기본 서브넷이 선택됩니다. 다른 VPC에서 파일 시스템에 액세스해야 하는 경우 VPC를 변경합니다. 파일 시스템을 생성한 후 VPC를 변경하려면 먼저 현재 VPC에서 탑재 대상을 삭제해야 합니다. 자세한 내용은 탑재 대상 VPC 변경 단원을 참조하십시오. • 가용 영역 및 서브넷 ID - 탑재 대상의 가용 영역 또는 서브넷을 변경할 수 있습니다. • IP 주소 유형 - 기본적으로 Amazon EFS는 탑재 대상을 생성할 IPv4 주소를 사용합니다. 대안으로 IPv6 주소만 지원하려면 IPv6 전용을 선택하거나, IPv4 및 IPv6 주소를 모두 지원하려면 듀얼 스택을 선택할 수 있습니다. **Note** IP 주소 유형은 서브넷의 IP 유형과 일치해야 합니다. 또한 IP 주소 유형은 서브넷의 IP 주소 지정 속성을 재정의합니다. 예를 들어, IP 주소 유형이 IPv4 전용이고 서브넷에 대해 IPv6 주소 지정 속성이 활성화된 경우, 해당 서브넷에서 생성된 네트워크 인터페이스는 서브넷의 주소 범위에서 IPv4 주소를 할당받습니다. 자세한 내용은

설정	설명
	서브넷의 IP 주소 지정 속성 수정을 참조하세요. • IP 주소 - 기본적으로 Amazon EFS는 지정된 서브넷의 사용 가능한 주소에 탑재 대상을 생성합니다. 또는 탑재 대상의 IP 주소를 지정할 수 있습니다. 탑재 대상 생성 후 IP 주소를 변경할 수 없습니다. 탑재 대상을 삭제한 다음 새 주소로 탑재 대상을 새로 생성해야 합니다. • 보안 그룹 - VPC 대상에 할당된 보안 그룹이 기본적으로 사용됩니다. 보안 그룹을 추가하거나 제거할 수 있습니다. 보안 그룹에 대한 자세한 내용은 탑재 대상 보안 그룹 변경를 (을) 참조하세요. AWS CLI, Amazon EFS API 또는 AWS SDKs를 사용하여 파일 시스템을 생성하는 경우 동시에 탑재 대상을 생성할 수 없습니다. 파일 시스템이 생성될 때까지 기다린 후 CreateMountTarget API 작업을 사용하여 탑재 대상을 생성해야 합니다. 탑재 대상에 대한 자세한 내용은 탑재 대상 생성 섹션을 참조하세요.

콘솔을 사용하여 빠른 생성

Amazon EFS 콘솔을 사용하여 권장 설정이 있는 Amazon EFS 파일 시스템을 생성합니다. 사용자 지정 구성으로 파일 시스템을 생성하려는 경우 콘솔을 사용하여 사용자 지정 생성을 참조하세요.

권장 설정이 있는 Amazon EFS 파일 시스템을 빠르게 생성하려면

1. 에 로그인 AWS Management Console 하고 https://console.aws.amazon.com/efs/ Amazon EFS 콘솔을 엽니다.
2. 파일 시스템 생성을 선택해 파일 시스템 생성 마법사를 엽니다.
3. (선택 사항) 이름에 파일 시스템의 이름을 입력합니다.
4. Virtual Private Cloud(VPC)의 경우 VPC를 선택하거나 기본 VPC로 설정해 두세요.
5. 생성을 선택하여 다음과 같은 서비스 권장 설정을 사용하는 파일 시스템을 생성합니다.

   - 리전별 가용성
   - 30일 후 파일 시스템을 EFS Infrequent Access(IA) 스토리지로, 90일 후 EFS Archive 스토리지로 전환하고 EFS Standard 스토리지로 전환하지 않는 수명 주기 정책
   - 저장 데이터 암호화 활성화
   - 탄력적 처리량 모드
   - 범용 성능 모드
   - 지정된 서브넷에서 사용 가능한 IPv4를 사용하여 파일 시스템이 생성되는 AWS 리전 의 각 가용 영역에 구성된 대상 탑재

   파일 시스템 페이지는 상단에 생성한 파일 시스템의 상태를 보여주는 배너와 함께 나타납니다. 파일 시스템을 사용할 수 있게 되면 배너에 파일 시스템 세부 정보 페이지에 액세스할 수 있는 링크가 나타납니다.

   파일 시스템 상태에 대한 자세한 내용은 파일 시스템 상태에 대한 이해 섹션을 참조하세요.

콘솔을 사용하여 사용자 지정 생성

이 섹션에서는 Amazon EFS 콘솔을 사용하여 서비스 권장 설정을 사용하는 대신 사용자 지정 설정이 포함된 EFS 파일 시스템을 생성하는 프로세스를 설명합니다. 권장 설정을 사용하여 파일 시스템을 생성하는 자세한 방법은 콘솔을 사용하여 빠른 생성 단원을 참조하세요.

콘솔을 사용하여 사용자 지정 설정으로 EFS 파일 시스템을 생성하는 과정은 4단계로 구성됩니다.

- 1단계 - 스토리지 클래스 및 처리량 모드를 포함한 일반 파일 시스템 설정을 구성합니다.
- 2단계 - Virtual Private Cloud(VPC)와 탑재 대상을 포함한 파일 시스템 네트워크 설정을 구성합니다. 각 탑재 대상에 대해 가용 영역, 서브넷, IP 주소, 보안 그룹을 설정합니다.

- 3단계 - (선택 사항)파일 시스템에 대한 NFS 클라이언트 액세스를 제어하는 파일 시스템 정책을 생성합니다.
- 4단계 - 파일 시스템 설정을 검토하고 변경한 다음 파일 시스템을 생성합니다.

1단계: 파일 시스템 설정 구성

1. 에 로그인 AWS Management Console 하고 https://console.aws.amazon.com/efs/ Amazon EFS 콘솔을 엽니다.
2. 파일 시스템 생성을 선택해 파일 시스템 생성 마법사를 엽니다.
3. 권장 설정을 사용하여 파일 시스템을 만드는 대신 사용자 지정을 선택하면 사용자 지정 파일 시스템을 만들 수 있습니다. 파일 시스템 설정 페이지가 열립니다.
4. 네트워크 설정에서 다음을 수행합니다.

   a. (선택 사항)이름에 파일 시스템의 이름을 입력합니다.

   b. 파일 시스템 유형의 경우 기본적으로 리전별이 선택됩니다. 단일 가용 영역 내에 파일 시스템 데이터와 메타데이터를 중복 저장하는 파일 시스템을 생성하려면 One Zone을 선택합니다. One Zone을 선택하는 경우 파일 시스템을 생성할 가용 영역을 선택하거나 기본 값을 유지하십시오.

   c. Lifecycle Management의 경우 필요에 따라 수명 주기 정책을 변경하십시오.

      - IA로 전환 - Standard 스토리지에서 파일을 마지막으로 액세스한 이후 시간을 기준으로 파일을 Inrequent Access(IA) 스토리지 클래스로 전환할 시기를 선택합니다.
      - Archive로 전환 - Standard 스토리지에서 파일을 마지막으로 액세스한 이후 시간을 기준으로 파일을 Archive 스토리지 클래스로 전환할 시기를 선택합니다.
      - Standard로 전환 - 파일 시스템을 스토리지 클래스로 전환할지 여부를 선택합니다.

      수명 주기 정책에 대한 자세한 내용은 스토리지 수명 주기 관리 섹션을 참조하십시오.

   d. Amazon EFS는 AWS Key Management Service 기본적으로 (AWS KMS) EFS 서비스 키(`aws/elasticfilesystem`)를 사용하여 저장 데이터를 암호화합니다. 암호화에 사용할 다른 KMS 키를 선택하려면 암호화 설정 사용자 지정을 확장하고 목록에서 키를 선택합니다. 또는 사용하려는 KMS 키의 KMS 키 ID나 Amazon 리소스 이름(ARN)을 입력합니다.

      새 키를 생성해야 하는 경우 생성을 AWS KMS key 선택하여 AWS KMS 콘솔을 시작하고 새 키를 생성합니다.

      확인란을 선택 취소하여 저장 데이터의 암호화를 해제할 수 있습니다.

파일 시스템 생성 후에는 암호화 설정을 변경할 수 없습니다. 자세한 내용은 Amazon EFS의 데이터 암호화 단원을 참조하십시오.

5. 성능 설정에서 다음을 수행합니다.

   a. 처리량 모드의 경우 탄력적 모드가 기본적으로 선택됩니다.

      - 프로비저닝 처리량 모드를 사용하려면 프로비저닝 모드를 선택하고 프로비저닝 처리량(MiB/s)에서 파일 시스템 요청에 프로비저닝할 처리량을 입력합니다. 최대 읽기 처리량은 입력한 처리량의 3배로 표시됩니다.
      - 버스팅 처리량을 사용하려면 버스팅을 선택합니다.

      처리량 모드로 선택하면 파일 시스템의 월별 예상 비용이 표시됩니다. 파일 시스템을 사용할 수 있게 된 후에 처리량 모드를 변경할 수 있습니다.

      성능 요구 사항에 맞는 올바른 처리량 모드를 선택하는 방법에 대한 자세한 내용은 처리량 모드을 참조하세요.

   b. 성능 모드 선택에서 기본값은 범용입니다. 성능 모드를 변경하려면 추가 설정을 확장한 다음 최대 I/O를 선택합니다.

      파일 시스템을 사용할 수 있게 된 후에는 성능 모드를 변경할 수 없습니다. 자세한 내용은 성능 모드 단원을 참조하십시오.

      > **Important**
      >
      > 최대 I/O에서는 작업당 지연 시간이 길어지기 때문에 모든 파일 시스템에 범용 성능 모드를 사용하는 것이 좋습니다.

6. (선택 사항)파일 시스템에 태그 키-값 페어를 추가합니다.
7. 다음을 선택하여 파일 시스템에 대한 네트워크 액세스를 구성합니다.

2단계: 네트워크 액세스 구성

2단계에서는 VPC와 탑재 대상을 포함한 파일 시스템의 네트워크 설정을 구성합니다.

1. EC2 인스턴스를 파일 시스템에 연결할 Virtual Private Cloud(VPC)를 선택합니다. 자세한 내용은 탑재 대상 생성 단원을 참조하십시오.

2. 탑재 대상의 경우 파일 시스템에 대한 탑재 대상을 하나 이상 생성합니다. 각 탑재 대상에 대해 다음 속성을 설정하세요.

   - 가용 영역 - 기본적으로 탑재 대상은 AWS 리전의 각 가용 영역에 구성됩니다. 특정 가용 영역에 탑재 대상을 두지 않으려면 제거를 선택하여 해당 영역의 탑재 대상을 삭제하세요. 파일 시스템에 액세스하려는 모든 가용 영역에 탑재 대상을 생성합니다. 추가 비용은 없습니다.
   - 서브넷 ID - 가용 영역에서 사용 가능한 서브넷 중에서 선택합니다. 기본 서브넷은 미리 선택되어 있습니다.
   - IP 주소 유형 - IPv4 전용을 선택하면 IPv4 주소만 지원하고, IPv6 전용을 선택하면 IPv6 주소만 지원하며, 듀얼 스택을 선택하면 IPv4 및 IPv6 주소를 모두 지원합니다.
   - IPv4 또는 IPv6 주소 - 탑재 대상을 배치하려는 IP 주소를 알고 있는 경우 IP 주소 유형과 일치하는 IP 주소 상자에 해당 주소를 입력합니다. 값을 지정하지 않으면 Amazon EFS가 지정된 서브넷에서 사용되지 않은 IP 주소를 선택합니다.
   - 보안 그룹 - 기본적으로 Amazon EFS는 VPC의 기본 보안 그룹을 선택합니다. 보안 그룹을 변경하려면 할당된 그룹을 삭제한 다음 보안 그룹 선택 목록에서 그룹을 선택합니다. 탑재 대상에 하나 이상의 보안 그룹을 지정할 수 있습니다. 자세한 내용은 VPC 보안 그룹 사용 단원을 참조하십시오.
3. 탑재 대상이 없는 가용 영역에 탑재 대상을 만들려면 탑재 대상 추가를 선택합니다. 탑재 대상이 각 가용 영역에 구성된 경우 이 옵션을 사용할 수 없습니다.
4. 다음을 선택하여 파일 시스템 정책을 저장합니다.

3단계: 파일 시스템 정책 생성(선택 사항)

이 단계에서는 파일 시스템의 파일 시스템 정책을 생성합니다. EFS 파일 시스템 정책은 해당 파일 시스템에 대한 NFS 클라이언트 액세스를 제어하는 데 사용되는 IAM 리소스 정책입니다. 자세한 내용은 IAM을 사용하여 파일 시스템에 대한 액세스 제어 단원을 참조하십시오.

1. 정책 옵션에서 사전 구성된 파일 시스템 정책을 원하는 대로 조합하여 선택할 수 있습니다.

   - 기본적으로 루트 액세스를 차단 - 이 옵션은 허용된 EFS 작업 집합에서 `ClientRootAccess`를 제거합니다.
   - 기본적으로 읽기 전용 액세스를 적용 - 이 옵션은 허용된 EFS 작업 집합에서 `ClientWriteAccess`를 제거합니다.
   - 익명 액세스 방지 - 이 옵션은 허용된 EFS 작업 집합에서 `ClientMount`를 제거합니다.
   - 모든 클라이언트에 전송 중 암호화 적용 - 이 옵션은 암호화되지 않은 클라이언트에 대한 액세스를 거부합니다.

사전 구성된 정책을 선택하면 정책 JSON 개체가 정책 편집기 창에 표시됩니다.

2. 추가 권한 부여를 사용하여 파일 시스템 권한을 다른 IAM 보안 주체를 포함한 추가 IAM 보안 주체에 부여합니다 AWS 계정. 추가를 선택하고 권한을 부여하려는 엔티티의 보안 주체 ARN을 입력합니다. 부여할 권한을 선택합니다. 추가 권한은 정책 편집기에 표시됩니다.
3. 정책 편집기를 사용하여 사전 구성된 정책을 사용자 지정하거나 자체 파일 시스템 정책을 만들 수 있습니다. 편집기를 사용하면 사전 구성된 정책 옵션을 사용할 수 없게 됩니다. 현재 파일 시스템 정책을 지우고 새 정책을 만들려면 지우기를 선택합니다.
4. 다음을 선택하여 파일 시스템을 검토하고 생성합니다.

4단계: 검토 및 생성

1. 각 파일 시스템 구성 그룹을 검토하세요. 지금은 편집을 선택하여 각 그룹을 변경할 수 있습니다.
2. 파일 시스템 생성을 선택하여 파일 시스템을 생성하고 파일 시스템 페이지로 돌아갑니다.

   상단의 배너는 새 파일 시스템이 생성되고 있음을 나타냅니다. 파일 시스템을 사용할 수 있게 되면 배너에 새 파일 시스템 세부 정보 페이지에 액세스할 수 있는 링크가 나타납니다.

를 사용하여 생성 AWS CLI

를 사용할 때 이러한 리소스를 순서대로 AWS CLI생성합니다. 먼저, 파일 시스템을 만듭니다. 그런 다음 해당 AWS CLI 명령을 사용하여 파일 시스템에 대한 탑재 대상 및 추가 선택적 태그를 생성할 수 있습니다.

다음 예에서는 `adminuser`를 `--profile` 파라미터 값으로 사용합니다. 적절한 사용자 프로파일을 사용하여 보안 인증 정보를 입력해야 합니다. 자세한 내용은 AWS Command Line Interface 사용 설명서에서 AWS CLI를 사용하기 위한 전제 조건을 참조하세요.

- 자동 백업이 활성화된 상태에서 암호화된 파일 시스템을 생성하려면 다음과 같이 Amazon EFS `create-file-system` CLI 명령(해당 작업은 CreateFileSystem)을 사용합니다.

```
aws efs create-file-system \
--creation-token creation-token \
--encrypted \
--backup \
--performance-mode generalPurpose \
--throughput-mode elastic \
--region aws-region \
```

```
--tags Key=key,Value=value Key=key1,Value=value1 \
--profile adminuser
```

예를 들어 다음 `create-file-system` 명령은 `us-west-2`AWS 리전에서 탄력적 처리량을 사용하여 파일 시스템을 만듭니다. 이 명령은 `MyFirstFS`를 생성 토큰으로 지정합니다. Amazon EFS 파일 시스템을 생성할 수 AWS 리전 있는 목록은의 Amazon EFS 엔드포인트 및 할당량을 참조하세요Amazon Web Services 일반 참조.

```
aws efs create-file-system \
--creation-token MyFirstFS \
--backup \
--encrypted \
--performance-mode generalPurpose \
--throughput-mode elastic \
--region us-west-2 \
--tags Key=Name,Value="Test File System" Key=developer,Value=rhoward \
--profile adminuser
```

파일 시스템을 만든 후 Amazon EFS에서는 다음 예에서처럼 파일 시스템 설명을 JSON으로 반환합니다.

```
{
    "OwnerId": "123456789abcd",
    "CreationToken": "MyFirstFS",
    "Encrypted": true,
    "FileSystemId": "fs-c7a0456e",
    "CreationTime": 1422823614.0,
    "LifeCycleState": "creating",
    "Name": "Test File System",
    "NumberOfMountTargets": 0,
    "SizeInBytes": {
        "Value": 6144,
        "ValueInIA": 0,
        "ValueInStandard": 6144
        "ValueInArchive": 0
    },
    "PerformanceMode": "generalPurpose",
    "ThroughputMode": "elastic",
    "Tags": [
      {
         "Key": "Name",
         "Value": "Test File System"
```

```
      }
   ]
}
```

- 다음 예제는 availability-zone-name 속성을 사용하여 us-west-2a 가용 영역에 버스팅 처리량을 사용하는 파일 시스템을 생성합니다.

```
aws efs create-file-system \
--creation-token MyFirstFS \
--availability-zone-name us-west-2a \
--backup \
--encrypted \
--performance-mode generalPurpose \
--throughput-mode bursting \
--region us-west-2 \
--tags Key=Name,Value="Test File System" Key=developer,Value=rhoward \
--profile adminuser
```

파일 시스템을 만든 후 Amazon EFS에서는 다음 예에서처럼 파일 시스템 설명을 JSON으로 반환합니다.

```
{
    "AvailabilityZoneId": "usw-az1",
    "AvailabilityZoneName": "us-west-2a",
    "OwnerId": "123456789abcd",
    "CreationToken": "MyFirstFS",
    "Encrypted": true,
    "FileSystemId": "fs-c7a0456e",
    "CreationTime": 1422823614.0,
    "LifeCycleState": "creating",
    "Name": "Test File System",
    "NumberOfMountTargets": 0,
    "SizeInBytes": {
        "Value": 6144,
        "ValueInIA": 0,
        "ValueInStandard": 6144
        "ValueInArchive": 0
    },
    "PerformanceMode": "generalPurpose",
    "ThroughputMode": "bursting",
    "Tags": [
      {
```

```
            "Key": "Name",
            "Value": "Test File System"
        }
    ]
}
```

또한 Amazon EFS에서는 다음과 같이 계정에서 파일 시스템 목록을 검색하는 데 사용할 수 있는 `describe-file-systems` CLI 명령(해당 API 작업은 DescribeFileSystems)도 제공합니다.

```
aws efs describe-file-systems \
--region aws-region \
--profile adminuser
```

Amazon EFS는 지정된 리전에서 AWS 계정 생성된의 파일 시스템 목록을 반환합니다.

EFS 파일 시스템 삭제

파일 시스템 삭제는 실행 취소할 수 없는 안전하지 않은 작업입니다. 파일 시스템과 그 안에 저장된 모든 데이터가 손실됩니다. 파일 시스템에서 삭제한 데이터는 사라지며, 해당 데이터를 복원할 수 없습니다. 사용자가 파일 시스템에서 데이터를 삭제하면 해당 데이터는 즉시 사용할 수 없게 됩니다. EFS force가 마지막으로 데이터를 덮어씁니다.

> **Note**
>
> 복제 구성의 일부인 파일 시스템은 삭제할 수 없습니다. 먼저 복제 구성을 삭제해야 합니다. 자세한 내용은 복제 구성 삭제 단원을 참조하십시오.

> **Important**
>
> 파일 시스템을 삭제하기 전에는 항상 해당 파일 시스템의 탑재를 해제해야 합니다.

콘솔 사용

파일 시스템을 삭제하려면

1. Amazon Elastic File System 콘솔(https://console.aws.amazon.com/efs/)을 엽니다.

2. 파일 시스템 목록에서 삭제하려는 파일 시스템을 선택합니다.
3. 삭제를 선택합니다.
4. 파일 시스템 삭제 대화 상자에 표시된 파일 시스템 ID를 입력하고 확인을 선택하여 삭제를 확인합니다.

   콘솔은 간소화된 파일 삭제를 지원합니다. 먼저 연결된 탑재 대상을 삭제한 후 파일 시스템을 삭제합니다.

사용 AWS CLI

AWS CLI 명령을 사용하여 파일 시스템을 삭제하려면 먼저 파일 시스템에 대해 생성된 모든 탑재 대상과 액세스 포인트를 삭제해야 합니다.

예제 AWS CLI 명령은 단원을 참조하십시오4단계: 정리.

파일 시스템 정책 생성

Amazon EFS 콘솔 또는 AWS CLI를 사용하여 파일 시스템 정책을 생성할 수 있습니다. AWS SDKs 또는 Amazon EFS API를 직접 사용하여 프로그래밍 방식으로 파일 시스템 정책을 생성할 수도 있습니다. EFS 파일 시스템 정책은 20,000자로 제한됩니다. EFS 파일 시스템 정책 및 예제 사용에 대한 자세한 내용은 IAM을 사용하여 파일 시스템에 대한 액세스 제어 섹션을 참조하세요.

> **Note**
>
> Amazon EFS 파일 시스템 정책 변경 사항이 적용되는 데 몇 분 정도 걸릴 수 있습니다.

콘솔 사용

1. Amazon Elastic File System 콘솔(https://console.aws.amazon.com/efs/)을 엽니다.
2. 파일 시스템을 선택합니다.
3. 파일 시스템 페이지에서 파일 시스템 정책을 편집하거나 생성할 파일 시스템을 선택합니다.
4. 파일 시스템 정책을 선택한 다음 편집을 선택합니다.
5. 정책 옵션에서 사전 구성된 파일 시스템 정책을 원하는 대로 조합하여 선택할 수 있습니다.

   - 기본적으로 루트 액세스를 차단 - 이 옵션은 허용된 EFS 작업 집합에서 `ClientRootAccess`를 제거합니다.

- 기본적으로 읽기 전용 액세스를 적용 - 이 옵션은 허용된 EFS 작업 집합에서 `ClientWriteAccess`를 제거합니다.
- 익명 액세스 방지 - 이 옵션은 허용된 EFS 작업 집합에서 `ClientMount`를 제거합니다.
- 모든 클라이언트에 전송 중 암호화 적용 - 이 옵션은 암호화되지 않은 클라이언트에 대한 액세스를 거부합니다.

사전 구성된 정책을 선택하면 정책 JSON 개체가 정책 편집기 창에 표시됩니다.

6. 추가 권한 부여를 사용하여 파일 시스템 권한을 다른 IAM 보안 주체를 포함한 추가 IAM 보안 주체에 부여합니다 AWS 계정. 추가를 선택하고 권한을 부여하려는 엔티티의 보안 주체 ARN을 입력합니다. 부여할 권한을 선택합니다. 추가 권한은 정책 편집기에 표시됩니다.
7. 정책 편집기를 사용하여 사전 구성된 정책을 사용자 지정하거나 자체 파일 시스템 정책을 만들 수 있습니다. 편집기를 사용하면 사전 구성된 정책 옵션을 사용할 수 없게 됩니다. 현재 파일 시스템 정책을 지우고 새 정책을 만들려면 지우기를 선택합니다.

   편집기를 지우면 사전 구성된 정책을 다시 사용할 수 있게 됩니다.
8. 정책 편집을 완료한 후 저장을 선택합니다.

사용 AWS CLI

다음 예제에서 put-file-system-policy CLI 명령은 EFS 파일 시스템에 대한 지정된 AWS 계정 읽기 전용 액세스를 허용하는 파일 시스템 정책을 생성합니다. 동등한 API 명령은 PutFileSystemPolicy입니다.

```
aws efs put-file-system-policy --file-system-id fs-01234567 --policy '{
    "Id": "1",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "elasticfilesystem:ClientMount"
            ],
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:root"
            }
        }

    ]
}'
```

액세스 포인트 생성

AWS Management Console, AWS Command Line Interface (AWS CLI) 및 Amazon EFS API 및 SDK를 사용하여 Amazon EFS 액세스 포인트를 생성하고 삭제할 수 있습니다. SDKs 액세스 포인트를 생성한 후에는 수정할 수 없습니다. 파일 시스템은 증가를 요청하지 않는 한 최대 10,000개의 액세스 포인트를 가질 수 있습니다.

> **Note**
>
> 동일한 파일 시스템에 대한 액세스 포인트 생성 요청이 연속적으로 다수 전송되고, 해당 파일 시스템이 액세스 포인트 제한에 근접한 경우, 이러한 요청에 대해 스로틀링 응답이 발생할 수 있습니다. 이는 파일 시스템이 명시된 액세스 포인트 할당량을 초과하지 않도록 하기 위한 것입니다.

EFS 액세스 포인트에 대한 자세한 내용은 액세스 포인트 작업 섹션을 참조하세요.

콘솔 사용

1. Amazon Elastic File System 콘솔(https://console.aws.amazon.com/efs/)을 엽니다.
2. 액세스 포인트를 선택하여 액세스 포인트 창을 엽니다.
3. 액세스 포인트 생성 페이지에서 액세스 포인트 생성을 선택합니다.

   파일 시스템을 선택하여 액세스 포인트 생성 페이지를 열 수도 있습니다. 파일 시스템 이름 또는 파일 시스템 ID를 선택한 다음 액세스 포인트와 액세스 포인트 생성을 선택하여 해당 파일 시스템에 대한 액세스 포인트를 생성합니다.

   a. 세부 정보 창에서 다음 정보를 입력합니다.

      - 파일 시스템 - 파일 시스템 이름 또는 ID를 입력하고 일치하는 파일 시스템을 선택합니다. 입력 필드를 선택할 때 나타나는 목록에서 파일 시스템을 선택할 수도 있습니다.
      - (선택 사항)이름에 액세스 포인트의 이름을 입력합니다.
      - (선택 사항)루트 디렉터리 경로 - 액세스 포인트의 루트 디렉터리를 지정할 수 있습니다. 기본 액세스 포인트 루트는 /입니다. 루트 디렉터리 경로를 입력하려면 `/foo/bar` 형식을 사용합니다. 자세한 내용은 액세스 포인트를 사용하여 루트 디렉터리 적용 단원을 참조하십시오.

   b. (선택 사항)POSIX 사용자 창에서 액세스 포인트를 사용하는 NFS 클라이언트가 모든 파일 작업에 대해 사용자 및 그룹 정보를 적용하는 데 사용할 전체 POSIX 자격 증명을 지정할 수 있

습니다. 자세한 내용은 액세스 포인트를 사용하여 사용자 자격 증명 적용 단원을 참조하십시오.

- 사용자 ID - 사용자의 숫자 POSIX 사용자 ID입니다.
- 그룹 ID -사용자의 숫자 POSIX 그룹 ID입니다.
- 보조 그룹 ID - 쉼표로 구분된 보조 그룹 ID 목록을 선택적으로 입력합니다.

c. (선택 사항)루트 디렉터리 생성 권한의 경우, Amazon EFS가 루트 디렉터리 경로를 생성할 때 사용할 권한을 지정할 수 있습니다(지정되었으며 루트 디렉터리가 아직 존재하지 않는 경우). 자세한 내용은 액세스 포인트를 사용하여 루트 디렉터리 적용 단원을 참조하십시오.

> **Note**
>
> 루트 디렉터리 소유권 및 권한을 지정하지 않고 루트 디렉터리가 아직 없는 경우 EFS는 루트 디렉터리를 생성하지 않습니다. 액세스 포인트를 사용하여 파일 시스템을 탑재하려는 시도가 실패합니다.

- 소유자 사용자 ID - 루트 디렉터리 소유자로 사용할 숫자 POSIX 사용자 ID입니다.
- 소유자 그룹 ID – 루트 디렉터리 소유자 그룹으로 사용할 숫자 POSIX 그룹 ID입니다.
- Permissions - 디렉터리의 Unix 모드에 들어갑니다. 일반적인 구성은 755입니다. 탑재할 수 있도록 실행 비트가 액세스 포인트 사용자에 대해 설정되어 있는지 확인합니다.

4. 이 구성을 사용하여 액세스 포인트를 생성하려면 액세스 포인트 생성을 선택합니다.

사용 AWS CLI

다음 예제에서 `create-access-point` CLI 명령은 EFS 파일 시스템에 대한 액세스 포인트를 생성합니다. 동등한 API 명령은 CreateAccessPoint입니다.

```
aws efs create-access-point --file-system-id fs-abcdef0123456789a --client-token
 010102020-3 \
--root-directory “Path=/efs/mobileapp/
east,CreationInfo={OwnerUid=0,OwnerGid=11,Permissions=775}” \
--posix-user “Uid=22,Gid=4” \
--tags Key=Name,Value=east-users
```

요청이 성공하면 CLI는 액세스 포인트 설명으로 응답합니다.

```
{
    "ClientToken": "010102020-3",
    "Name": "east-users",
    "AccessPointId": "fsap-abcd1234ef5678901",
    "AccessPointArn": "arn:aws:elasticfilesystem:us-east-2:111122223333:access-point/
fsap-abcd1234ef5678901",
    "FileSystemId": "fs-01234567",
    "LifeCycleState": "creating",
    "OwnerId": "111122223333",
    "PosixUser": {
      "Gid": 4,
      "Uid": 22
    },
    "RootDirectory": {
    "CreationInfo": {
         "OwnerGid": 0,
         "OwnerUid": 11,
         "Permissions": "775"
      },
        "Path": "/efs/mobileapp/east",
    },
    "Tags": []
}
```

> **Note**
>
> 동일한 파일 시스템에 대한 액세스 포인트 생성 요청이 연속적으로 다수 전송되고, 해당 파일 시스템이 액세스 포인트 제한에 근접한 경우, 이러한 요청에 대해 스로틀링 응답이 발생할 수 있습니다. 이는 파일 시스템이 명시된 액세스 포인트 할당량을 초과하지 않도록 하기 위한 것입니다.

액세스 포인트 삭제

액세스 포인트를 삭제하면 액세스 포인트를 사용하는 모든 클라이언트가 액세스 포인트를 구성한 Amazon EFS 파일 시스템에 대한 액세스 권한을 잃게 됩니다.

콘솔 사용

1. Amazon Elastic File System 콘솔(https://console.aws.amazon.com/efs/)을 엽니다.

2. 왼쪽 탐색 창에서 액세스 포인트를 선택하여 액세스 포인트 페이지를 엽니다.
3. 삭제할 액세스 포인트를 선택합니다.
4. 삭제를 선택합니다.
5. [확인] 을 선택하여 작업을 확인하고 액세스 포인트를 삭제합니다.

사용 AWS CLI

다음 예제에서 `delete-access-point` CLI 명령은 지정된 액세스 포인트를 삭제합니다. 동등한 API 명령은 DeleteAccessPoint입니다. 명령이 성공하면 서비스가 비어있는 HTTP 본문과 함께 HTTP 204 응답을 다시 돌려 줍니다.

```
aws efs delete-access-point --access-point-id fsap-092e9f80b3fb5e6f3 --client-token
 010102020-3
```

EFS 리소스에 태그 지정

EFS 리소스를 관리하는 데 도움이 되도록 태그 형식으로 각 리소스에 고유한 메타데이터를 할당할 수 있습니다. 태그를 사용하면 용도, 소유자 또는 환경 등 다양한 방식으로 AWS 리소스를 분류할 수 있습니다. 이 분류는 동일 유형의 리소스가 많을 때 유용합니다. 지정한 태그에 따라 특정 리소스를 빠르게 식별할 수 있습니다. 이 주제에서는 태그를 설명하고 태그를 생성하는 방법을 보여줍니다.

태그 기본 사항

태그는 AWS 리소스에 할당하는 레이블입니다. 각 태그는 사용자가 정의하는 키와 선택적 값으로 구성됩니다.

태그를 사용하면 용도, 소유자 또는 환경 등 다양한 방식으로 AWS 리소스를 분류할 수 있습니다. 예를 들어 계정의 Amazon EFS 파일 시스템에 각 파일 시스템의 소유자를 추적하는 데 도움이 되는 태그 집합을 정의할 수 있습니다.

각 리소스 유형에 대한 요건을 충족하는 태그 키 세트를 고안하는 것이 좋습니다. 일관된 태그 키 세트를 사용하면 리소스를 보다 쉽게 관리할 수 있습니다. 추가하는 태그에 따라 리소스를 검색하고 필터링할 수 있습니다.

태그는 Amazon EFS에는 의미가 없으며 엄격하게 문자열로 해석됩니다. 또한 태그는 리소스에 자동으로 배정되지 않습니다. 태그 키와 값을 편집할 수 있으며 언제든지 리소스에서 태그를 제거할 수 있습니다. 태그의 값을 빈 문자열로 설정할 수 있지만 태그의 값을 Null로 설정할 수는 없습니다. 해당 리소

스에 대해 키가 기존 태그와 동일한 태그를 추가하는 경우 새 값이 이전 값을 덮어씁니다. 리소스를 삭제하면 리소스에 대한 모든 태그도 삭제됩니다.

태그 제한 사항

태그에 적용되는 기본 제한 사항은 다음과 같습니다.

- 리소스당 최대 태그 수 – 50개
- 각 리소스에 대해 각 태그 키는 고유하며 하나의 값만 가질 수 있습니다.
- 최대 키 길이 - UTF-8 형식의 유니코드 문자 128자
- 최대 값 길이 - UTF-8 형식의 유니코드 문자 256자
- Amazon EFS는 태그에 모든 문자를 사용할 수 있지만, 다른 서비스에는 제한이 적용되기도 합니다. 서비스에서 허용되는 문자는 UTF-8로 표현할 수 있는 문자, 숫자 및 공백과 특수 문자 + - = . _ : / @ 입니다.
- 태그 키와 값은 대/소문자를 구분합니다.
- `aws:` 접두사는 AWS 사용을 위해 예약되어 있습니다. 태그에 이 접두사가 있는 태그 키가 있는 경우 태그의 키 또는 값을 편집하거나 삭제할 수 없습니다. `aws:` 접두사가 지정된 태그는 리소스당 태그 수 제한에 포함되지 않습니다.

태그에만 기초하여 리소스를 업데이트 또는 삭제할 수 없습니다. 리소스 식별자를 지정해야 합니다. 예를 들어 `DeleteMe`라는 태그 키로 태그를 지정한 파일 시스템을 삭제하려면 해당 파일 시스템 리소스 식별자(예: `fs-1234567890abcdef0`)를 지정하여 `DeleteFileSystem` 작업을 사용해야 합니다.

퍼블릭 또는 공유 리소스에 태그를 지정할 경우 할당하는 태그는 사용자의 AWS 계정에만 사용할 수 있습니다. 다른 AWS 계정 사용자는 이러한 태그에 액세스할 수 없습니다. 공유 리소스에 대한 태그 기반 액세스 제어를 위해 각는 리소스에 대한 액세스를 제어하기 위해 자체 태그 세트를 할당 AWS 계정 해야 합니다.

Amazon EFS 파일 시스템 및 액세스 포인트 리소스에 태그를 지정할 수 있습니다.

액세스 제어에 태그 사용

태그를 사용하여 Amazon EFS 리소스에 대한 액세스를 제어하고 ABAC(속성 기반 액세스 제어)를 구현할 수 있습니다.

> **Note**
>
> Replication은 ABAC(속성 기반 액세스 제어)에 대한 태그 사용을 지원하지 않습니다.

리소스에 태그 지정

계정에 이미 존재하는 Amazon EFS 파일 시스템 및 액세스 포인트 리소스에 태그를 지정할 수 있습니다.

콘솔 사용

- Amazon EFS 콘솔을 사용하면 리소스 세부 정보 화면의 태그 탭을 사용하여 기존 리소스에 태그를 적용할 수 있습니다. Amazon EFS 콘솔에서는 리소스를 생성할 때 리소스의 태그를 지정할 수 있습니다. 예를 들어 키가 `Name`인 태그는 지정한 값을 사용할 수 있습니다.

  대부분의 경우, 콘솔은 리소스 생성 직후(리소스 생성 중이 아니라) 태그를 적용합니다. 콘솔은 `Name` 태그에 따라 리소스를 조직할 수도 있지만 이 태그는 Amazon EFS 서비스에 대한 의미가 없습니다.

사용 AWS CLI

- Amazon EFS API, AWS CLI또는 AWS SDK를 사용하는 경우 `TagResource` EFS API 작업을 사용하여 기존 리소스에 태그를 적용할 수 있습니다. 또한 일부 리소스 생성 작업에서는 리소스 생성 시 리소스의 태그를 지정할 수 있습니다.

  태그를 관리하기 위한 AWS CLI 명령과 이에 상응하는 Amazon EFS API 작업이 다음 표에 나열되어 있습니다.

CLI 명령	설명	해당하는 API 작업
tag-resource	새 태그 추가 또는 기존 태그 업데이트	TagResource
list-tags-for-resource	기존 태그 검색	ListTagsForResource
untag-resource	기존 태그 삭제	UntagResource

자습서: 쓰기 가능한 사용자별 하위 디렉터리 생성

EFS 파일 시스템을 생성하여 Amazon EC2(EC2) 인스턴스에 로컬로 탑재하면 *## ### ##*라는 빈 디렉터리가 나타납니다. 이 파일 시스템 루트 디렉터리에 대한 일반적인 사용 사례 중 하나는 EC2 인스턴스에 생성하는 각 사용자에 대해 "쓰기 가능한" 하위 디렉터리를 생성하고 사용자의 홈 디렉터리에 하위 디렉터리를 탑재하는 것입니다. 그러면 사용자가 홈 디렉터리에 생성하는 모든 파일 및 하위 디렉터리가 EFS 파일 시스템에 생성됩니다.

> **Note**
>
> 시작하기 연습에 따라 EC2 인스턴스에 EFS 파일 시스템을 생성하고 탑재할 수 있습니다.

다음 단계에서는 사용자를 생성하고, 사용자를 위한 하위 디렉터리를 생성하고, 사용자를 하위 디렉터리의 소유자로 설정한 다음 사용자의 홈 디렉터리에 Amazon EFS 하위 디렉터리를 탑재합니다.

1. 사용자 mike 만들기:

   - EC2 인스턴스에 로그인합니다. 루트 권한(이 경우에는 sudo 명령 사용)을 사용하여 사용자를 생성하고 비밀번호를 할당합니다.

     예를 들어, 다음 명령은 사용자 mike를 생성합니다.

     ```
     $ sudo useradd -c "Mike Smith" mike
     $ sudo passwd mike
     ```

     사용자의 홈 디렉터리도 생성됩니다. 예를 들어 /home/mike입니다.

2. *EFSroot* 아래에 사용자를 위한 하위 디렉터리를 생성합니다.

   예를 들어 다음 명령은 *EFSroot* 아래에 하위 디렉터리 mike를 생성합니다.

   ```
   $  sudo mkdir /EFSroot/mike
   ```

   *EFSroot*를 로컬 디렉터리 이름으로 바꿔야 합니다.

3. 루트 사용자 및 루트 그룹은 하위 디렉터리의 소유자입니다. ls -l 명령을 사용하여 이를 확인할 수 있습니다. 이 하위 디렉터리에 대한 전체 권한을 사용자에게 부여하려면 사용자에게 디렉터리의 소유권을 부여합니다.

예제:

```
$ sudo chown mike:mike /EFSroot/mike
```

4. mount 명령을 사용하여 하위 디렉터리를 사용자의 홈 디렉터리에 탑재합니다.

   예제:

```
$  sudo mount -t nfs -o
 nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport mount-target-DNS:/mike  /home/mike
```

   mount-target-DNS 주소는 원격 EFS 파일 시스템 루트를 식별합니다.

이 탑재 대상을 탑재 해제하면 사용자는 다시 탑재하지 않고는 디렉터리에 액세스할 수 없으며, 다시 탑재하려면 루트 권한이 필요합니다.

EFS 파일 시스템 탑재

EFS 파일 시스템을 탑재하려면 EFS 탑재 도우미를 사용하는 것이 좋습니다. EFS 탑재 도우미는 지원되는 배포판을 실행하는 EC2 Linux 및 Mac 인스턴스에 EFS 파일 시스템을 탑재할 수 있도록 도와줍니다. 탑재 도우미는 Amazon EFS 클라이언트(`amazon-efs-utils`)를 설치할 때 설치되는 오픈 소스 도구 모음의 일부입니다. Amazon EFS 클라이언트 및 지원되는 배포판에 대한 자세한 내용은 Amazon EFS 클라이언트 수동 설치 단원을 참조하세요.

또는 표준 Linux NFS 클라이언트를 사용하여 EFS 파일 시스템을 수동으로 탑재할 수 있습니다. Amazon EC2 인스턴스에 파일 시스템을 탑재하는 경우, Amazon EFS에서는 네트워크 파일 시스템 버전 4.0 및 4.1(NFSv4) 프로토콜을 지원합니다.

또한 EFS 탑재 도우미 또는 NFS를 사용하여 인스턴스가 시작될 때 자동으로 EFS 파일 시스템을 탑재하도록 EC2 인스턴스를 구성할 수 있습니다.

주제

- Linux의 탑재 고려 사항
- EFS 탑재 도우미를 사용하여 EFS 파일 시스템 탑재
- Network File System을 사용하여 EFS 파일 시스템 탑재
- EFS 파일 시스템 자동 탑재
- 파일 시스템 탑재 해제
- 자습서:를 사용하여 EFS 파일 시스템을 생성하고 EC2 인스턴스에 탑재 AWS CLI
- 자습서: 온프레미스 Linux 클라이언트로 탑재
- 자습서: 다른 VPC에서 파일 시스템 탑재하기
- 탑재 문제 해결

Linux의 탑재 고려 사항

다음 Linux 탑재 옵션 값을 사용하는 것이 좋습니다.

- `rsize=1048576` – NFS 클라이언트가 각 네트워크 READ 요청에 대해 수신할 수 있는 최대 데이터 바이트 수를 설정합니다. 이 값은 EFS 파일 시스템의 파일에서 데이터를 읽을 때 적용됩니다. 성능이 저하되지 않도록 최대한 큰 크기(최대 `1048576`)를 사용하는 것이 좋습니다.

- wsize=1048576 – NFS 클라이언트가 각 네트워크 WRITE 요청에 대해 전송할 수 있는 최대 데이터 바이트 수를 설정합니다. 이 값은 EFS 파일 시스템의 파일에 데이터를 쓸 때 적용됩니다. 성능이 저하되지 않도록 최대한 큰 크기(최대 1048576)를 사용하는 것이 좋습니다.
- hard – NFS 요청 시간이 초과된 후에는 NFS 클라이언트의 복구 동작을 설정하여 서버가 응답할 때까지 NFS 요청을 무기한 재시도합니다. 데이터 무결성을 위하여 하드 탑재 옵션(hard)을 사용하는 것이 좋습니다. 그러나 soft 탑재를 사용하는 경우 timeo 파라미터를 150데시초(15초) 이상으로 설정해야 합니다. 이렇게 하면 소프트 탑재에 고유한 데이터 손상 위험을 최소화하는 데 도움이 됩니다.
- timeo=600 – NFS 클라이언트가 600데시초(60초) 후에 재시도하기 전에 NFS 클라이언트가 응답을 기다리는 데 사용하는 시간 초과 값을 설정합니다. 시간 제한 파라미터(timeo)를 변경해야 하는 경우 15초에 해당하는 150 이상의 값을 사용하는 것이 좋습니다. 이렇게 하면 원하지 않는 성능 저하를 방지할 수 있습니다.
- retrans=2 – NFS 클라이언트가 추가 복구 작업을 시도하기 전에 요청을 재시도하는 횟수를 2로 설정합니다.
- noresvport – 네트워크 연결이 다시 설정될 때 NFS 클라이언트가 새로운 권한이 없는 전송 제어 프로토콜(TCP) 소스 포트를 사용하도록 지시합니다. 이렇게 하면 네트워크 복구 이벤트 후에도 EFS 파일 시스템을 중단 없이 사용할 수 있습니다.
- _netdev – /etc/fstab에서 네트워크가 활성화될 때까지 클라이언트가 EFS 파일 시스템을 탑재하는 것을 방지합니다.

일반적으로 기본값과 다른 탑재 옵션을 설정하지 마세요. 성능 저하 및 기타 문제가 발생할 수 있습니다. 위의 기본값을 사용하지 않으면 다음 사항에 유의하세요.

- 읽기 또는 쓰기 버퍼 크기를 변경하거나 속성 캐싱을 비활성화하면 성능이 저하될 수 있습니다.
- Amazon EFS는 소스 포트를 무시합니다. Amazon EFS 소스 포트를 변경하더라도 어떠한 영향도 미치지 않습니다.
- Amazon EFS는 Kerberos 보안 변형을 지원하지 않습니다. 예를 들어 다음 탑재 명령은 실패합니다.

```
$ mount -t nfs4 -o krb5p <DNS_NAME>:/ /efs/
```

- DNS 이름을 사용하여 파일 시스템을 탑재하는 것이 좋습니다. Amazon EFS는 외부 리소스를 직접 호출하지 않고 Amazon EC2 인스턴스와 동일한 가용 영역에 있는 EFS 탑재 대상의 IP 주소로 이 이름을 확인합니다. EC2 인스턴스와 다른 가용 영역에 있는 탑재 대상을 사용하면 가용 영역을 통해 전송된 데이터에 대한 표준 EC2 요금이 부과됩니다. 또한 파일 시스템 작업의 대기 시간이 늘어날 수 있습니다.

- 추가 탑재 옵션과 기본값에 대한 자세한 설명은 Linux 설명서를 참조하세요.

> **Note**
>
> 탑재된 EFS 파일 시스템의 상태와 관계없이 EC2 인스턴스를 시작해야 하는 경우, `/etc/fstab` 파일의 파일 시스템 항목에 `nofail` 옵션을 추가합니다.

EFS 탑재 도우미를 사용하여 EFS 파일 시스템 탑재

Amazon EFS 클라이언트(`amazon-efs-utils`)를 설치한 후에는 EFS 탑재 도우미를 사용하여 지원되는 배포판을 실행하는 EC2 Linux 및 Mac 인스턴스에 EFS 파일 시스템을 탑재할 수 있습니다. Amazon EFS는 EC2 Windows 인스턴스에서의 탑재를 지원하지 않습니다.

> **Important**
>
> 성공적인 탑재를 위해 항상 최신 버전의 `amazon-efs-utils`를 사용하는 것이 좋습니다. 예를 들어 2.3 `amazon-efs-utils` 이전의 버전은 IPv6 주소를 사용한 탑재를 지원하지 않습니다.

파일 시스템을 탑재할 때 탑재 도우미는 Linux의 표준 `mount` 명령과 완전히 호환되는 `efs`라는 새로운 네트워크 파일 시스템 유형을 정의합니다. 또한 탑재 도우미는 EC2 Linux 인스턴스에 있는 `/etc/fstab` 구성 파일의 항목을 사용해 자동으로 인스턴스 부팅 시 인스턴스에 EFS 파일 시스템을 탑재하는 것을 지원합니다.

> **Warning**
>
> 파일 시스템을 자동으로 마운트하는 경우 네트워크 파일 시스템 식별에 사용하는 `_netdev` 옵션을 사용합니다. `_netdev`이 빠진 경우 EC2 인스턴스가 응답을 중지합니다. 컴퓨팅 인스턴스가 네트워킹을 시작한 후 네트워크 파일 시스템의 초기화를 완료해야 하기 때문입니다. 자세한 내용은 자동 탑재 실패 및 인스턴스 무응답 단원을 참조하십시오.

다음 속성 중 하나만 지정하여 파일 시스템을 탑재할 수 있습니다.

- 파일 시스템 DNS 이름 - 파일 시스템 DNS 이름을 사용하고 있는데 탑재 도우미가 이름을 확인할 수 없는 경우(예: 다른 VPC에 파일 시스템을 마운트하는 경우) 탑재 대상 IP 주소를 대신 사용합니다. 자세한 내용은 다른 AWS 계정 또는 VPC에서 EFS 파일 시스템 탑재 단원을 참조하십시오.
- 파일 시스템 ID - 파일 시스템 ID를 사용하는 경우 탑재 도우미는 외부 리소스를 직접 호출하지 않고 탑재 대상 탄력적 네트워크 인터페이스(ENI)의 로컬 IP 주소로 파일을 확인합니다.
- 탑재 대상 IP 주소 - 파일 시스템 탑재 대상 중 하나의 IP 주소를 사용할 수 있습니다.

Amazon EFS 콘솔에서 이러한 모든 속성의 값을 찾을 수 있습니다. 파일 시스템 DNS 이름은 연결 화면에서 찾을 수 있습니다.

전송 중 데이터 암호화를 EFS 파일 시스템의 탑재 옵션으로 선언하면, 탑재 도우미가 클라이언트 `stunnel` 프로세스와 `amazon-efs-mount-watchdog`이라는 관리자 프로세스를 초기화합니다. 이 `amazon-efs-mount-watchdog` 프로세스는 TLS 탑재의 상태를 모니터링하며, EFS 파일 시스템이 TLS를 통해 처음 탑재될 때 자동으로 시작됩니다. 클라이언트가 Linux에서 실행되는 경우 이 프로세스는 해당 Linux 배포판에 따라 `upstart` 또는 `systemd`에 의해 관리됩니다. 지원되는 macOS에서 실행되는 클라이언트의 경우 `launchd`에서 관리합니다.

`Stunnel`은 다목적 오픈 소스 네트워크 릴레이입니다. 클라이언트 `stunnel` 프로세스는 로컬 포트에서 인바운드 트래픽을 수신 대기하고, 탑재 도우미는 NFS 클라이언트 트래픽을 이 로컬 포트로 리디렉션합니다.

탑재 도우미는 TLS 버전 1.2를 사용해 파일 시스템과 통신합니다. TLS를 사용하기 위해서는 인증서가 필요하며, 신뢰할 수 있는 Amazon 인증 기관이 서명한 인증서여야 합니다. 암호화 작동 방식에 대한 자세한 내용은 Amazon EFS의 데이터 암호화 섹션을 참조하세요.

주제

- EFS 탑재 도우미에서 사용하는 탑재 설정
- 지원 로그 얻기
- EFS 탑재 도우미를 사용하기 위한 사전 조건
- EFS 탑재 도우미를 사용하여 EC2 Linux 인스턴스에 탑재
- EFS 탑재 도우미를 사용하여 EC2 Mac 인스턴스에 탑재
- 다른에서 EFS 파일 시스템 탑재 AWS 리전
- One Zone 파일 시스템 탑재하기
- IAM 권한 부여를 통한 탑재
- EFS 액세스 포인트를 사용한 탑재

- 여러 EC2 인스턴스에 EFS 탑재
- 다른 AWS 계정 또는 VPC에서 EFS 파일 시스템 탑재

EFS 탑재 도우미에서 사용하는 탑재 설정

Amazon EFS 탑재 도우미 클라이언트는 Amazon EFS에 최적화된 다음과 같은 탑재 옵션을 사용합니다.

- `nfsvers=4.1` - EC2 Linux 인스턴스에 탑재할 때 사용됩니다.

  `nfsvers=4.0` - macOS Big Sur, Monterey, Ventura를 실행하는 지원되는 EC2 Mac 인스턴스에 마운트할 때 사용됩니다.
- `rsize=1048576` – NFS 클라이언트가 원하지 않는 성능 저하를 방지하기 위해 가능한 가장 큰 1048576에 대한 각 네트워크 READ 요청에 대해 수신할 수 있는 최대 데이터 바이트 수를 설정합니다.
- `wsize=1048576` – NFS 클라이언트가 원하지 않는 성능 저하를 방지하기 위해 가능한 가장 큰 1048576에 대한 각 네트워크 WRITE 요청에 대해 송신할 수 있는 최대 데이터 바이트 수를 설정합니다.
- `hard` – NFS 요청 시간이 초과된 후에는 NFS 클라이언트의 복구 동작을 설정하여 데이터 무결성을 확인하기 위하여 서버가 응답할 때까지 NFS 요청을 무기한 재시도합니다.
- `timeo=600` – NFS 클라이언트가 600데시초(60초) 후에 재시도하기 전에 데이터 무결성을 확인하기 위하여 NFS 클라이언트가 응답을 기다리는 데 사용하는 시간 초과 값을 설정합니다.
- `retrans=2` – NFS 클라이언트가 추가 복구 작업을 시도하기 전에 요청을 재시도하는 횟수를 2로 설정합니다.
- `noresvport` – 네트워크 연결이 다시 설정될 때 NFS 클라이언트가 새로운 권한이 없는 전송 제어 프로토콜(TCP) 소스 포트를 사용하도록 지시합니다. 이 `noresvport` 옵션을 사용하면 재연결 또는 네트워크 복구 이벤트 후에도 EFS 파일 시스템을 중단 없이 사용할 수 있습니다.
- `mountport=2049` - macOS Big Sur, Monterey, Ventura를 실행하는 EC2 Mac 인스턴스에 마운트할 때만 사용됩니다.

지원 로그 얻기

EFS 탑재 도우미에는 EFS 파일 시스템에 대한 로깅 기능이 내장되어 있습니다. 문제 해결을 위해 이러한 로그를 AWS Support와 공유할 수 있습니다. EFS 탑재 도우미를 사용하여 클라이언트의 `/var/log/amazon/efs`에 저장된 로그를 찾을 수 있습니다. EFS 탑재 도우미, stunnel 프로세스(기본값으

로 비활성화됨), stunnel 프로세스를 모니터링하는 `amazon-efs-mount-watchdog` 프로세스용 로그입니다.

> **Note**
>
> `amazon-efs-mount-watchdog` 프로세스는 각 탑재의 stunnel 프로세스를 실행시키고, EFS 파일 시스템의 탑재가 해제되었을 때 stunnel을 중단합니다. 어떤 이유로 stunnel 프로세스가 예기치 않게 종료된 경우, 이 감시 프로세스가 stunnel 프로세스를 다시 시작합니다.

`/etc/amazon/efs/efs-utils.conf`의 로그 구성을 변경할 수 있습니다. 로그 변경 내용을 적용하려면 EFS 탑재 도우미를 사용하여 파일 시스템을 탑재 해제했다가 다시 탑재해야 합니다. 탑재 도우미와 감시 로그의 로그 용량은 20MiB로 제한되어 있습니다. Stunnel 프로세스의 로그는 기본값으로 비활성화되어 있습니다.

> **Important**
>
> Stunnel 프로세스 로그에 대한 로깅을 활성화 할 수 있습니다. 그러나 활성화된 stunnel 로그가 파일 시스템에서 상당한 공간을 사용할 수 있습니다.

EFS 탑재 도우미를 사용하기 위한 사전 조건

Amazon EFS 탑재 도우미를 사용해 Amazon EC2 인스턴스에 EFS 파일 시스템을 탑재할 수 있습니다. 탑재 도우미를 사용하려면 다음 절차를 따라야 합니다.

- 탑재할 파일 시스템의 파일 시스템 ID - EFS 탑재 도우미는 외부 리소스를 직접 호출하지 않고 탑재 대상 탄력적 네트워크 인터페이스(ENI)의 로컬 IP 주소로 파일 시스템 ID를 확인합니다.
- EFS 탑재 대상 - 가상 프라이빗 클라우드(VPC)에 탑재 대상을 생성합니다. 서비스 권장 설정을 사용하여 콘솔에서 파일 시스템을 생성하는 경우 파일 시스템이 AWS 리전 있는의 각 가용 영역에 탑재 대상이 생성됩니다. 탑재 대상을 만드는 방법에 대한 지침은 탑재 대상 생성 섹션을 참조하세요.

  > **Note**
  >
  > DNS를 통해 파일 시스템을 마운트하기 전에 새로 생성된 탑재 대상의 수명 주기 상태를 사용할 수 있게 된 후 60초 정도 기다리는 것이 좋습니다. 이 대기를 통해 DNS 레코드가 파일 시스템이 AWS 리전 있는에 완전히 전파됩니다.

EC2 인스턴스와 다른 가용 영역에 있는 탑재 대상을 사용하면 가용 영역을 통해 전송된 데이터에 대한 표준 EC2 요금이 부과됩니다. 또한 파일 시스템 작업의 대기 시간이 늘어날 수 있습니다.

- 다른 가용 영역의 One Zone 파일 시스템을 탑재하는 경우:
  - 파일 시스템의 가용 영역 이름 - EC2 인스턴스와 다른 가용 영역에 위치한 EFS One Zone 파일 시스템을 탑재하는 경우.
  - 탑재 대상 DNS 이름 - 또는 가용 영역 대신 탑재 대상의 DNS 이름을 지정할 수 있습니다.
- 지원되는 Linux 또는 macOS 배포를 실행하는 EC2 인스턴스 - 탑재 도우미를 사용하여 파일 시스템을 탑재하기 위해 지원되는 배포판은 다음과 같습니다.
  - Amazon Linux 2
  - Amazon Linux 2023
  - Amazon Linux 2017.09 이상
  - macOS Big Sur
  - Red Hat Enterprise Linux(CentOS 같은 계열 시스템 포함) 버전 7 이상
  - Ubuntu 16.04 LTS 이상

  > **Note**
  >
  > macOS Big Sur를 실행하는 EC2 Mac 인스턴스는 NFS 4.0만 지원합니다.

- EFS 탑재 도우미가 EC2 인스턴스에 설치됨 - 탑재 도우미는 유틸리티 `amazon-efs-utils` 패키지의 도구입니다. `amazon-efs-utils` 설치에 대한 자세한 내용은 Amazon EFS 클라이언트 수동 설치 섹션을 참조하세요.
- The EC2 인스턴스가 VPC에 있음 – 연결 EC2 인스턴스는 Amazon VPC 서비스를 기반으로 하는 Virtual Private Cloud(VPC)에 있어야 합니다. 또한에서 제공하는 DNS 서버를 사용하도록 구성해야 합니다 AWS. Amazon DNS 서버에 대한 자세한 정보는 Amazon VPC 사용 설명서에서 Amazon VPC의 DHCP 옵션 세트를 참조하세요.
- VPC에 활성화된 DNS 호스트 이름이 있음 – 연결 중인 EC2 인스턴스의 VPC에 DNS 호스트 이름이 활성화되어 있어야 합니다. 자세한 내용은 Amazon VPC 사용 설명서의 VPC에 대한 DNS 속성을 참조하세요.
- 다른의 EC2 인스턴스 및 파일 시스템의 경우 AWS 리전 - 탑재하려는 EC2 인스턴스와 파일 시스템이 다른에 있는 경우 `efs-utils.conf` 파일에서 `region` 속성을 편집 AWS 리전해야 합니다. 자세한 내용은 다른에서 EFS 파일 시스템 탑재 AWS 리전 단원을 참조하십시오.

EFS 탑재 도우미를 사용하여 EC2 Linux 인스턴스에 탑재

이 절차에는 다음이 필요합니다.

- Amazon EC2 인스턴스에 `amazon-efs-utils` 패키지를 설치해야 합니다. 자세한 내용은 Amazon EFS 클라이언트 수동 설치 단원을 참조하십시오.
- 파일 시스템에 탑재 대상을 만들었습니다. 자세한 내용은 탑재 대상 생성 단원을 참조하십시오.

EC2 Linux 인스턴스에 탑재 도우미를 사용하여 EFS 파일 시스템을 탑재하는 방법

1. Secure Shell(SSH)를 통해 EC2 인스턴스의 터미널 창을 열고, 적절한 사용자 이름으로 로그인합니다. 자세한 내용은 Amazon EC2 사용 설명서의 EC2 인스턴스에 연결을 참조하세요.
2. 다음 명령을 사용하여 파일 시스템 탑재 지점으로 사용할 디렉터리 `efs`를 생성합니다.

   ```
   sudo mkdir efs
   ```

3. 다음 명령 중 하나를 실행하여 파일 시스템을 탑재합니다.

   > **Note**
   >
   > 탑재하려는 EC2 인스턴스와 파일 시스템이 다른 AWS 리전에 위치한 경우 `efs-utils.conf` 파일의 `region` 속성을 편집하려면 다른에서 EFS 파일 시스템 탑재 AWS 리전을 참조하세요.

   - 파일 시스템 ID를 사용하여 탑재하려면:

     ```
     sudo mount -t efs file-system-id efs-mount-point/
     ```

     `efs-mount-point` 대신 *`file-system-id`* 및 `efs`의 위치에 탑재하려는 파일 시스템의 ID를 사용하세요.

     ```
     sudo mount -t efs fs-abcd123456789ef0 efs/
     ```

     또는 전송 중 데이터 암호화를 사용하려면 다음 명령을 사용하여 파일 시스템을 탑재할 수 있습니다.

```
sudo mount -t efs -o tls fs-abcd123456789ef0:/ efs/
```

- 파일 시스템 DNS 이름을 사용하여 탑재하려면:

```
sudo mount -t efs -o tls file-system-dns-name efs-mount-point/
```

```
sudo mount -t efs -o tls fs-abcd123456789ef0.efs.us-east-2.amazonaws.com efs/
```

- 탑재 대상 IP 주소를 사용하여 탑재하려면:

```
sudo mount -t efs -o tls,mounttargetip=mount-target-ip file-system-id efs-mount-point/
```

```
sudo mount -t efs -o tls,mounttargetip=192.0.2.0 fs-abcd123456789ef0 efs/
```

연결 대화 상자에서 파일 시스템을 탑재하기 위한 정확한 명령을 보고 복사할 수 있습니다.

a. Amazon EFS 콘솔에서 탑재하려는 파일 시스템을 선택하여 세부 정보 페이지를 표시합니다.

b. 이 파일 시스템에 사용할 탑재 명령을 표시하려면 오른쪽 상단에서 연결을 선택합니다.

   연결 화면에는 다음 방법들로 파일 시스템을 탑재하는 데 사용할 정확한 명령이 표시됩니다.

   - (DNS를 통해 탑재) EFS 탑재 도우미 또는 NFS 클라이언트에서 파일 시스템의 DNS 이름을 사용합니다.
   - (IP를 통해 탑재) 선택한 가용 영역의 탑재 대상 IP 주소를 NFS 클라이언트와 함께 사용합니다.

EFS 탑재 도우미를 사용하여 EC2 Mac 인스턴스에 탑재

이 절차에는 다음이 필요합니다.

- Amazon EC2 Mac 인스턴스에 `amazon-efs-utils` 패키지를 설치해야 합니다. 자세한 내용은 macOS Big Sur, macOS Monterey 또는 macOS Ventura를 실행하는 EC2 Mac 인스턴스에 Amazon EFS 클라이언트 설치 단원을 참조하십시오.
- 파일 시스템에 탑재 대상을 만들었습니다. 파일 시스템 생성 시 탑재 대상을 생성하여 기존 파일 시스템에 추가할 수 있습니다. 자세한 내용은 탑재 대상 생성 단원을 참조하십시오.

- macOS Big Sur, Monterey 또는 Ventura를 실행하는 EC2 Mac 인스턴스에 파일 시스템을 탑재하고 있습니다. 다른 macOS 버전은 지원되지 않습니다.

> **Note**
>
> macOS Big Sur, Monterey, Ventura를 실행하는 EC2 Mac 인스턴스만 지원됩니다. 다른 macOS 버전은 Amazon EFS와 함께 사용할 수 없습니다.

macOS Big Sur, Monterey, 또는 Ventura를 실행하는 EC2 Mac 인스턴스에 EFS 탑재 도우미를 사용하여 EFS 파일 시스템을 탑재하는 방법

1. Secure Shell(SSH)를 통해 EC2 Mac 인스턴스의 터미널 창을 열고, 적절한 사용자 이름으로 로그인합니다. 자세한 내용은 Amazon EC2 사용 설명서의 EC2 인스턴스에 연결을 참조하세요.
2. 다음 명령을 사용하여 파일 시스템 탑재 지점으로 사용할 디렉터리를 생성합니다.

   ```
   sudo mkdir efs
   ```

3. 다음 명령을 실행해 파일 시스템을 탑재합니다.

   > **Note**
   >
   > 기본적으로 EFS 탑재 도우미는 탑재 명령의 `tls` 옵션 사용 여부에 관계없이 EC2 Mac 인스턴스에 탑재할 때 전송 중 암호화를 사용합니다.

   ```
   sudo mount -t efs file-system-id efs-mount-point/
   ```

   ```
   sudo mount -t efs fs-abcd123456789ef0 efs/
   ```

   탑재할 때도 `tls` 옵션을 사용할 수 있습니다.

   ```
   sudo mount -t efs -o tls fs-abcd123456789ef0:/ efs
   ```

   전송 중 암호화를 사용하지 않고 EC2 Mac 인스턴스에 파일 시스템을 탑재하려면 다음 명령에 표시된 대로 `notls` 옵션을 사용합니다.

```
sudo mount -t efs -o notls file-system-id efs-mount-point/
```

다음과 같이 관리 콘솔의 연결 대화 상자에서 파일 시스템을 탑재하기 위한 정확한 명령을 보고 복사할 수 있습니다.

a. Amazon EFS 콘솔에서 탑재하려는 파일 시스템을 선택하여 세부 정보 페이지를 표시합니다.

b. 이 파일 시스템에 사용할 탑재 명령을 표시하려면 오른쪽 상단에서 연결을 선택합니다.

   연결 화면에는 다음 방법들로 파일 시스템을 탑재하는 데 사용할 정확한 명령이 표시됩니다.

   - (DNS를 통해 탑재) EFS 탑재 도우미 또는 NFS 클라이언트에서 파일 시스템의 DNS 이름을 사용합니다.
   - (IP를 통해 탑재) 선택한 가용 영역의 탑재 대상 IP 주소를 NFS 클라이언트와 함께 사용합니다.

다른에서 EFS 파일 시스템 탑재 AWS 리전

파일 시스템과 다른에 있는 EC2 인스턴스에서 EFS AWS 리전 파일 시스템을 탑재하려면 `efs-utils.conf` 파일에서 `region` 속성 값을 편집해야 합니다.

`efs-utils.conf`에서 `region` 속성을 편집하려면

1. Secure Shell(SSH)를 통해 EC2 인스턴스의 터미널에 액세스하고, 적절한 사용자 이름으로 로그인합니다. 자세한 내용은 Amazon EC2 사용 설명서의 EC2 인스턴스에 연결을 참조하세요.
2. 선호하는 편집기를 사용하여 `/etc/amazon/efs/efs-utils.conf` 파일을 찾아서 엽니다.
3. 다음 줄을 찾습니다.

   ```
   #region = us-east-1
   ```

   a. 해당 줄의 주석 처리를 제거합니다.

   b. 파일 시스템이 `us-east-1` 리전에 없는 경우 `us-east-1`을 파일 시스템이 위치한 리전의 ID로 교체하세요.

   c. 변경 사항을 저장합니다.
4. 리전 간 탑재를 위한 호스트 항목을 추가합니다. 이 작업을 수행하는 방법에 대한 자세한 내용은 3단계: 탑재 대상에 대한 호스트 항목 추가 섹션을 참조하세요.

5. Linux 또는 Mac 인스턴스용 EFS 탑재 도우미를 사용하여 파일 시스템을 탑재합니다.

One Zone 파일 시스템 탑재하기

EFS One Zone 파일 시스템은 파일 시스템과 동일한 가용 영역에 있는 탑재 대상을 하나만 지원합니다. 탑재 대상을 더 추가할 수 없습니다. 이 섹션에서는 One Zone 파일 시스템을 탑재할 때 고려할 사항을 설명합니다.

파일 시스템의 탑재 대상과 동일한 가용 영역에 있는 Amazon EC2 컴퓨팅 인스턴스를 사용하여 EFS 파일 시스템에 액세스하면 가용 영역 간 데이터 전송 요금을 피하고 성능을 높일 수 있습니다.

이 섹션에는 다음 절차가 포함됩니다.

- EC2 인스턴스에 `amazon-efs-utils package`를 설치했습니다. 자세한 내용은 Amazon EFS 클라이언트 수동 설치 단원을 참조하십시오.
- 파일 시스템에 대한 탑재 대상을 만들었습니다. 자세한 내용은 탑재 대상 생성 단원을 참조하십시오.

EC2의 One Zone 파일 시스템을 다른 가용 영역에 탑재

다른 가용 영역에 위치한 Amazon EC2 인스턴스에 One Zone 파일 시스템을 탑재하는 경우 탑재 도우미 탑재 명령에서 파일 시스템의 가용 영역 이름 또는 파일 시스템 탑재 대상의 DNS 이름을 지정해야 합니다.

다음 명령을 사용하여 파일 시스템 탑재 지점으로 사용할 `efs`라는 디렉터리를 생성합니다.

```
sudo mkdir efs
```

EFS 탑재 도우미를 사용하여 파일 시스템을 탑재하려면 다음 명령을 사용합니다. 명령은 파일 시스템의 가용 영역 이름을 지정합니다.

```
sudo mount -t efs -o az=availability-zone-name,tls file-system-id mount-point/
```

다음은 샘플 값이 포함된 명령입니다.

```
sudo mount -t efs -o az=us-east-1a,tls fs-abcd1234567890ef efs/
```

다음 명령은 파일 시스템 탑재 대상의 DNS 이름을 지정하여 파일 시스템을 탑재합니다.

```
sudo mount -t efs -o tls mount-target-dns-name mount-point/
```

다음은 예제 탑재 대상 DNS 이름이 포함된 명령입니다.

```
sudo mount -t efs -o tls us-east-1a.fs-abcd1234567890ef9.efs.us-east-1.amazonaws.com
 efs/
```

EFS 탑재 도우미를 사용하여 다른 가용 영역에 One Zone 파일 시스템을 자동으로 탑재합니다.

다른 가용 영역에 위치한 EC2 인스턴스에 One Zone 파일 시스템을 탑재하기 위하여 /etc/fstab를 사용하는 경우 /etc/fstab 항목에 파일 시스템의 가용 영역 이름 또는 파일 시스템 탑재 대상의 DNS 이름을 지정해야 합니다.

```
availability-zone-name.file-system-id.efs.aws-region.amazonaws.com:/ efs-mount-point
 efs defaults,_netdev,noresvport,tls 0 0
```

```
us-east-1a.fs-abc123def456a7890.efs.us-east-1.amazonaws.com:/ efs-one-zone efs
 defaults,_netdev,noresvport,tls 0 0
```

NFS로 One Zone 파일 시스템을 자동으로 탑재합니다.

다른 가용 영역에 위치한 EC2 인스턴스에 One Zone 스토리지를 사용하여 EFS 파일 시스템을 탑재하기 위하여 /etc/fstab을 사용하는 경우 /etc/fstab 항목에서 파일 시스템의 가용 영역 이름 또는 파일 시스템 탑재 대상의 DNS 이름을 지정해야 합니다.

```
availability-zone-name.file-system-id.efs.aws-region.amazonaws.com:/ efs-mount-point
 nfs4
 nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport,_netdev 0
 0
```

```
us-east-1a.fs-abc123def456a7890.efs.us-east-1.amazonaws.com:/ efs-one-zone nfs4
 nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport,_netdev 0
 0
```

/etc/fstab 파일을 편집하는 방법 및 이 명령에 사용된 값에 대한 자세한 내용은 EFS 파일 시스템 자동 탑재 섹션을 참조하세요.

다른 AWS 컴퓨팅 인스턴스에 One Zone 파일 시스템을 사용하여 파일 시스템 탑재

Amazon Elastic Container Service, Amazon Elastic Kubernetes Service 또는와 함께 One Zone 파일 시스템을 사용하는 AWS Lambda경우 EFS 파일 시스템이 있는 동일한 가용 영역을 사용하도록 서비스를 구성해야 하며, 다음 섹션에 설명되어 있습니다.

Amazon Elastic Container Service에서 연결

Amazon ECS에서 EFS 파일 시스템을 사용하여 컨테이너 인스턴스의 플릿 간에 파일 시스템 데이터를 공유할 수 있으므로 해당 작업이 차지한 인스턴스와 상관없이 동일한 영구 스토리지에 액세스할 수 있습니다. EFS One Zone 파일 시스템을 Amazon ECS와 함께 사용하려면 태스크를 시작할 때 파일 시스템과 동일한 가용 영역에 있는 서브넷만 선택해야 합니다. 자세한 내용을 알아보려면 Amazon Elastic Container Service 개발자 안내서의 Amazon EFS 볼륨을 참조하세요.

Amazon Elastic Kubernetes 서비스에서 연결

Amazon EKS의 One Zone 파일 시스템을 탑재하는 경우, EFS 액세스 포인트를 지원하는 Amazon EFS 컨테이너 스토리지 인터페이스(CSI) 드라이버를 사용하여 Amazon EKS 또는 자체 관리형 Kubernetes 클러스터의 여러 포드 간에 파일 시스템을 공유할 수 있습니다. 아마존 EFS CSI 드라이버는 Fargate 스택에 설치됩니다. EFS One Zone 파일 시스템과 함께 Amazon EFS CSI 드라이버를 사용

하는 경우, 포드를 시작할 때 `nodeSelector` 옵션을 사용하여 파일 시스템과 동일한 가용 영역 내에서 스케줄링되도록 할 수 있습니다.

에서 연결 AWS Lambda

와 함께 Amazon EFS AWS Lambda 를 사용하여 함수 호출 간에 데이터를 공유하고, 대규모 참조 데이터 파일을 읽고, 영구 및 공유 저장소에 함수 출력을 쓸 수 있습니다. Lambda는 함수 인스턴스를 동일한 가용 영역 및 서브넷에 있는 EFS 탑재 대상에 안전하게 연결합니다. One Zone 파일 시스템과 Lambda를 사용하는 경우, 파일 시스템과 동일한 가용 영역에 있는 서브넷에서만 간접 호출을 시작하도록 함수를 구성하십시오.

IAM 권한 부여를 통한 탑재

AWS Identity and Access Management (IAM) 권한 부여를 사용하여 Linux 인스턴스에 EFS 파일 시스템을 탑재하려면 EFS 탑재 도우미를 사용합니다. NFS 클라이언트에 대한 IAM 권한 부여에 대한 자세한 내용은 IAM을 사용하여 파일 시스템에 대한 액세스 제어 섹션을 참조하세요.

다음 단원에서 파일 시스템 탑재 지점으로 사용할 디렉터리를 생성해야 합니다. 다음 명령으로 탑재 지점 디렉터리 `efs`를 생성할 수 있습니다.

```
sudo mkdir efs
```

그런 다음 *`efs-mount-point`*의 인스턴스를 `efs`로 바꿀 수 있습니다.

EC2 인스턴스 프로파일을 사용하여 IAM 을 통해 탑재

인스턴스 프로파일이 있는 Amazon EC2 인스턴스에 IAM 권한 부여로 탑재하는 경우 다음과 같이 `tls` 및 `iam` 탑재 옵션을 사용합니다.

```
$ sudo mount -t efs -o tls,iam file-system-id efs-mount-point/
```

인스턴스 프로파일이 있는 EC2 인스턴스에 IAM 권한을 부여하여 자동으로 탑재하려면 EC2 인스턴스의 `/etc/fstab` 파일에 다음 줄을 추가합니다.

```
file-system-id:/ efs-mount-point efs _netdev,tls,iam 0 0
```

명명된 프로파일을 사용하여 IAM을 통해 탑재

자격 증명 파일 `~/.aws/credentials`또는 AWS CLI 구성 파일에 있는 IAM AWS CLI 자격 증명을 사용하여 IAM 권한 부여를 통해 탑재할 수 있습니다`~/.aws/config`. `"awsprofile"`이 지정되어 있지 않으면 “기본” 프로파일이 사용됩니다.

보안 인증 파일을 사용하여 Linux 인스턴스에 IAM 권한을 부여하여 탑재하려면 다음과 같이 `tls`, `awsprofile` 및 `iam` 탑재 옵션을 사용합니다.

```
$ sudo mount -t efs -o tls,iam,awsprofile=namedprofile file-system-id efs-mount-point/
```

보안 인증 파일을 사용하여 Linux 인스턴스에 IAM 권한을 부여하여 자동으로 탑재하려면 EC2 인스턴스의 `/etc/fstab` 파일에 다음 줄을 추가합니다.

```
file-system-id:/ efs-mount-point efs _netdev,tls,iam,awsprofile=namedprofile 0 0
```

EFS 액세스 포인트를 사용한 탑재

EFS 액세스 포인트만 사용하여 EFS 파일 시스템을 탑재하려면 EFS 탑재 도우미를 사용해야 합니다.

> **Note**
>
> EFS 액세스 포인트를 사용하여 파일 시스템을 탑재할 때는 파일 시스템에 대한 탑재 대상을 하나 이상 구성해야 합니다.

액세스 포인트를 사용하여 파일 시스템을 탑재하는 경우 탑재 명령에 일반 탑재 옵션 외에 `access-point-id` 및 `tls` 탑재 옵션이 포함됩니다. 예를 들면 다음과 같습니다.

```
$ sudo mount -t efs -o tls,accesspoint=access-point-id file-system-id efs-mount-point
```

액세스 포인트를 사용하여 파일 시스템을 자동으로 탑재하려면 EC2 인스턴스의 `/etc/fstab` 파일에 다음 줄을 추가합니다.

```
file-system-id efs-mount-point efs _netdev,tls,accesspoint=access-point-id 0 0
```

EFS 액세스 포인트에 대한 자세한 내용은 액세스 포인트 작업 섹션을 참조하세요.

여러 EC2 인스턴스에 EFS 탑재

AWS Systems Manager 실행 명령을 사용하여 인스턴스에 로그인하지 않고도 EFS 파일 시스템을 여러 Amazon EC2 인스턴스에 원격으로 안전하게 탑재할 수 있습니다. AWS Systems Manager Run Command에 대한 자세한 내용은 AWS Systems Manager 사용 설명서의 AWS Systems Manager Run Command를 참조하세요. 이 방법을 사용하여 EFS 파일 시스템을 탑재하려면 다음 사전 조건이 필요합니다.

1. EC2 인스턴스는 `AmazonElasticFileSystemsUtils` 권한 정책이 포함된 인스턴스 프로파일과 함께 시작됩니다. 자세한 내용은 1단계: 필요한 권한으로 IAM 인스턴스 프로파일 구성 단원을 참조하십시오.
2. Amazon EFS 클라이언트(amazon-efs-utils 패키지) 버전 1.28.1 이상이 EC2 인스턴스에 설치되어 있습니다. AWS Systems Manager를 사용하여 인스턴스에 패키지를 자동으로 설치할 수 있습니다. 자세한 내용은 2단계: State Manager에서 사용하는 연결 구성 단원을 참조하십시오.

콘솔을 사용하여 여러 EFS 파일 시스템을 여러 EC2 인스턴스에 탑재하려면

1. https://console.aws.amazon.com/systems-manager/ AWS Systems Manager 콘솔을 엽니다.
2. 탐색 창에서 명령 실행을 선택합니다.
3. 명령 실행을 선택합니다.
4. Commands 검색 필드에 **`AWS-RunShellScript`**를 입력합니다.
5. AWS-RunShellScript를 선택합니다.
6. 명령 파라미터에 탑재하려는 각 EFS 파일 시스템에 사용할 탑재 명령을 입력합니다. 예제:

   ```
   sudo mount -t efs -o tls fs-12345678:/ /mnt/efs
   sudo mount -t efs -o tls,accesspoint=fsap-12345678 fs-01233210 /mnt/efs
   ```

   Amazon EFS 클라이언트를 사용하는 EFS 탑재 명령에 대한 자세한 내용은 EFS 탑재 도우미를 사용하여 EC2 Linux 인스턴스에 탑재 또는 EFS 탑재 도우미를 사용하여 EC2 Mac 인스턴스에 탑재를 참조하세요.
7. 명령을 실행할 대상 AWS Systems Manager 관리형 EC2 인스턴스를 선택합니다.
8. 기타 원하는 추가 설정을 지정하세요. 그런 다음 실행을 선택하여 명령을 실행하고 명령에 지정된 EFS 파일 시스템을 탑재합니다.

   명령을 실행하면 명령 기록에서 상태를 볼 수 있습니다.

다른 AWS 계정 또는 VPC에서 EFS 파일 시스템 탑재

EFS 탑재 도우미를 사용하여 NFS 클라이언트 및 EFS 액세스 포인트에 대한 IAM 권한 부여를 사용하여 EFS 파일 시스템을 탑재할 수 있습니다. 기본적으로 EFS 탑재 도우미는 도메인 이름 서비스(DNS)를 사용하여 EFS 탑재 대상의 IP 주소를 확인합니다. 다른 계정이나 Virtual Private Cloud(VPC)에서 파일 시스템을 탑재하는 경우, EFS 탑재 대상을 수동으로 확인해야 합니다.

그런 다음, NFS 클라이언트에 사용할 올바른 EFS 탑재 대상 IP 주소를 결정하기 위한 지침을 찾을 수 있습니다. 또한 해당 IP 주소를 사용하여 EFS 파일 시스템을 탑재하도록 클라이언트를 구성하기 위한 지침을 찾을 수 있습니다.

주제

- 다른에서 EFS 파일 시스템 탑재 AWS 계정
- 다른 VPC에서 EFS 파일 시스템 탑재

다른에서 EFS 파일 시스템 탑재 AWS 계정

공유 VPC를 사용하면 다른 AWS 계정에서 소유한 Amazon EC2 인스턴스의 한 AWS 계정 에서 소유한 EFS 파일 시스템을 탑재할 수 있습니다. 공유 VPC 설정에 대한 자세한 내용은 Amazon VPC 피어링 설명서에서 다른 계정과 VPC 공유를 참조하세요.

VPC 공유를 설정하면 EC2 인스턴스가 DNS(Domain Name System) 이름 확인 또는 EFS 탑재 도우미를 사용하여 EFS 파일 시스템을 탑재할 수 있습니다. EFS 탑재 도우미를 사용하여 EFS 파일 시스템을 탑재하는 것이 좋습니다.

다른 VPC에서 EFS 파일 시스템 탑재

VPC 피어링 연결 또는 전송 게이트웨이를 사용하여 VPC를 연결하면 VPC가 다른 계정에 속해 있더라도 하나의 VPC에 있는 Amazon EC2 인스턴스가 다른 VPC의 EFS 파일 시스템에 액세스할 수 있습니다.

다른 VPC에서 EFS 탑재 지점을 확인할 때는 DNS 이름을 사용할 수 없습니다. EFS 파일 시스템을 탑재하기 위해 해당 가용 영역에서 탑재 지점의 IP 주소를 사용합니다.

또는 DNS 서비스로 Amazon Route 53을 사용할 수도 있습니다. Route 53에서 프라이빗 호스팅 영역과 리소스 레코드 세트를 생성하여 다른 VPC에서 EFS 탑재 대상 IP 주소를 확인할 수 있습니다. 확인하는 방법에 대한 자세한 내용은 Amazon Route 53 개발자 안내서의 프라이빗 호스팅 영역 작업을 참조하세요.

사전 조건

절차를 사용하기 전에 다음 단계를 수행하세요.

- EFS 파일 시스템을 탑재하려는 컴퓨팅 인스턴스에 유틸리티 `amazon-efs-utils` 세트의 일부인 Amazon EFS 클라이언트를 설치합니다. `amazon-efs-utils`에 포함된 EFS 탑재 도우미를 사용하여 파일 시스템을 탑재합니다. `amazon-efs-utils`를 설치하는 방법에 대한 지침은 Amazon EFS 클라이언트 수동 설치 섹션을 참조하세요.
- 인스턴스에 연결한 IAM 역할에 대해 IAM 정책에서 `ec2:DescribeAvailabilityZones` 작업을 허용하세요. AWS 관리형 정책을 IAM 엔`AmazonElasticFileSystemsUtils`터티에 연결하여 엔터티에 필요한 권한을 제공하는 것이 좋습니다.
- 다른에서 탑재할 때 파일 시스템 리소스 정책을 AWS 계정업데이트하여 다른의 보안 주체 ARN에 대한 `elasticfilesystem:DescribeMountTarget` 작업을 허용합니다 AWS 계정. 예제:

```
{
    "Id": "access-point-example03",
    "Statement": [
        {
            "Sid": "access-point-statement-example03",
            "Effect": "Allow",
            "Principal": {"AWS": "arn:aws:iam::555555555555:root"},
            "Action": "elasticfilesystem:DescribeMountTargets",
            "Resource": "arn:aws:elasticfilesystem:us-east-2:111122223333:file-
system/fs-12345678"
        }
    ]
}
```

  시스템 리소스 정책에 대한 자세한 내용은 Amazon EFS 내의 리소스 기반 정책 섹션을 참조하세요.
- botocore를 설치합니다. EFS 클라이언트는 다른 VPC에 파일 시스템을 탑재할 때 파일 시스템 DNS 이름을 확인할 수 없는 경우 botocore를 사용하여 탑재 대상 IP 주소를 검색합니다. 자세한 내용은 `amazon-efs-utils` README 파일에서 botocore 설치를 참조하세요.
- VPC 피어링 연결 또는 VPC 전송 게이트웨이를 설정합니다.

  VPC 피어링 연결 또는 VPC 전송 게이트웨이를 사용하여 클라이언트의 VPC와 EFS 파일 시스템의 VPC를 연결합니다. VPC 피어링 연결 또는 전송 게이트웨이를 사용하여 VPC를 연결하면 VPC가 다른 계정에 속해 있더라도 하나의 VPC에 있는 Amazon EC2 인스턴스가 다른 VPC의 EFS 파일 시스템에 액세스할 수 있습니다.

전송 게이트웨이는 VPC와 온프레미스 네트워크를 상호 연결하는 데 사용할 수 있는 네트워크 전송 허브입니다. 자세한 내용은 Amazon VPC 전송 게이트웨이 가이드의 전송 게이트웨이 시작하기를 참조하세요.

VPC 피어링 연결은 두 VPC 간의 네트워킹 연결입니다. 이러한 유형의 연결을 사용하면 프라이빗 Internet Protocol version 4(IPv4) 또는 Internet Protocol version 6(IPv6) 주소를 사용하여 이들 간의 트래픽을 라우팅할 수 있습니다. VPC 피어링을 사용하여 동일하거나 AWS 리전 그 사이에 VPCs를 연결할 수 있습니다 AWS 리전. VPC 피어링에 대한 자세한 내용은 Amazon VPC 피어링 가이드의 VPC 피어링이란?을 참조하세요.

파일 시스템의 고가용성을 보장하려면 항상 NFS 클라이언트와 동일한 가용 영역(AZ)에 있는 EFS 탑재 대상 IP 주소를 사용하는 것이 좋습니다. 다른 계정에 있는 EFS 파일 시스템을 탑재하는 경우, NFS 클라이언트와 EFS 탑재 대상이 동일한 가용 영역 ID에 있는지 확인하십시오. AZ 이름은 계정마다 다를 수 있으므로 이 요구 사항이 적용됩니다.

IAM 또는 액세스 포인트를 사용하여 다른 VPC에 EFS 파일 시스템 탑재

1. EC2 인스턴스에 연결합니다. 자세한 내용은 Amazon EC2 사용 설명서의 EC2 인스턴스에 연결을 참조하세요.
2. 다음 명령을 사용하여 파일 시스템을 탑재할 디렉터리를 생성합니다.

   ```
   $ sudo mkdir /mnt/efs
   ```

3. IAM 권한 부여를 사용하여 파일 시스템을 탑재하려면 다음 명령을 사용합니다.

   ```
   $ sudo mount -t efs -o tls,iam file-system-dns-name /mnt/efs/
   ```

   EFS에서 IAM 권한 부여를 사용하는 방법에 대한 자세한 내용은 IAM을 사용하여 파일 시스템에 대한 액세스 제어 섹션을 참조하세요.

   EFS 액세스 포인트를 사용하여 파일 시스템을 탑재하려면 다음 명령을 사용합니다.

   ```
   $ sudo mount -t efs -o tls,accesspoint=access-point-id file-system-dns-name /mnt/
   efs/
   ```

   EFS 액세스 포인트에 대한 자세한 내용은 액세스 포인트 작업 섹션을 참조하세요.

다른에서 EFS 파일 시스템 탑재 AWS 리전

파일 시스템과 다른에 있는 다른 VPC에서 EFS AWS 리전 파일 시스템을 탑재하는 경우 `efs-utils.conf` 파일을 편집해야 합니다. `/dist/efs-utils.conf`에서 다음 줄을 찾습니다.

```
#region = us-east-1
```

해당 줄의 주석 처리를 제거하고 파일 시스템이 위치한 리전의 ID 값을 바꾸세요(`us-east-1`에 없는 경우).

Network File System을 사용하여 EFS 파일 시스템 탑재

다음에서는 NFS(Network File System) 클라이언트를 설치하는 방법과 Amazon EC2 인스턴스에 Amazon EFS 파일 시스템을 탑재하는 방법에 대해 알아봅니다. 또한 `mount` 명령에 대한 설명과 `mount` 명령에서 파일 시스템의 도메인 이름 시스템(DNS) 이름을 지정하는 데 사용할 수 있는 옵션에 대한 설명도 찾을 수 있습니다. 또한 파일 `fstab`을 사용하여 시스템을 다시 시작한 후 파일 시스템을 자동으로 다시 탑재하는 방법에 대해서도 알아볼 수 있습니다.

> **Note**
>
> 이 섹션에서는 amazon-efs-utils 패키지 없이 Amazon EFS 파일 시스템을 탑재하는 방법을 알아봅니다. 파일 시스템에서 전송 중 데이터 암호화를 사용하려면 전송 계층 보안(TLS)을 사용해 파일 시스템을 탑재해야 합니다. 이렇게 하려면 amazon-efs-utils 패키지를 사용하는 것이 좋습니다. 자세한 내용은 Amazon EFS 클라이언트 수동 설치 단원을 참조하십시오.

주제

- 사전 조건
- NFS 지원
- NFS 클라이언트 설치
- 권장 NFS 탑재 설정
- DNS 이름을 사용하여 Amazon EC2에 탑재
- IP 주소를 사용하여 탑재

사전 조건

파일 시스템을 탑재하려면 먼저 다음 요구 사항을 충족해야 합니다.

- 관련 AWS 리소스를 생성, 구성 및 시작합니다. 지침은 Amazon EFS 시작하기 섹션을 참조하세요.
- Amazon EC2 인스턴스와 탑재 대상에 필요한 인바운드 및 아웃바운드 액세스를 포함하도록 VPC 보안 그룹을 생성합니다. 자세한 내용은 VPC 보안 그룹 사용 단원을 참조하십시오.

NFS 지원

Amazon EC2 인스턴스에 파일 시스템을 탑재하는 경우, Amazon EFS에서는 네트워크 파일 시스템 버전 4.0 및 4.1(NFSv4) 프로토콜을 지원합니다. NFSv4.0도 계속 지원되지만 NFSv4.1 사용을 권장합니다. Amazon EC2 인스턴스에 Amazon EFS 파일 시스템을 탑재하려면 선택한 NFSv4 프로토콜을 지원하는 NFS 클라이언트도 필요합니다. macOS Big Sur를 실행하는 Amazon EC2 Mac 인스턴스는 NFS v4.0만 지원합니다.

Amazon EFS는 `nconnect` 탑재 옵션을 지원하지 않습니다.

> **Note**
>
> Linux 커널 버전 5.4.*의 경우, 리눅스 NFS 클라이언트는 `read_ahead_kb` 기본값인 128KB를 사용합니다. 이 값을 15MB로 늘리는 것이 좋습니다. 자세한 내용은 NFS read_ahead_kb 크기 최적화 단원을 참조하십시오.

최적의 성능을 구현하고 다양한 유형의 알려진 NFS 클라이언트 버그를 피하기 위해 최신 Linux 커널을 사용하는 것이 좋습니다. 엔터프라이즈 Linux 배포판을 사용하는 경우 다음을 권장합니다.

- Amazon Linux 2
- Amazon Linux 2017.09 이상
- Red Hat Enterprise Linux(및 CentOS와 같은 파생 버전) 버전 8 이상
- Ubuntu 16.04 LTS 이상
- SLES 12 Sp2 이상

다른 배포판이나 사용자 지정 커널을 사용하고 있는 경우 커널 버전 4.3 이상을 권장합니다. EC2 인스턴스에서 Amazon EFS를 사용하는 경우 특정 AMI 또는 커널 버전과 관련된 문제를 해결하려면 AMI 및 커널 문제 해결 섹션을 참조하세요.

> **Note**
>
> Microsoft Windows를 실행하는 Amazon EC2 인스턴스에 EFS 파일 시스템을 탑재하는 것은 지원되지 않습니다.

NFS 클라이언트 설치

Amazon EC2 인스턴스에 EFS 파일 시스템을 탑재하려면 먼저 NFS 클라이언트를 설치해야 합니다. EC2 인스턴스에 연결하고 NFS 클라이언트를 설치하려면 EC2 인스턴스의 퍼블릭 DNS 이름과 로그인한 사용자 이름이 필요합니다. 인스턴스의 사용자 이름은 일반적으로 `ec2-user`입니다.

EC2 인스턴스 연결 및 NFS 클라이언트 설치

1. EC2 인스턴스에 연결합니다. 자세한 내용은 Amazon EC2 사용 설명서의 EC2 인스턴스에 연결을 참조하세요.

   SSH가 작동하려면 키가 공개적으로 표시되지 않아야 합니다. chmod 400 ***filename***.pem 명령을 사용하여 이 권한을 설정할 수 있습니다. 자세한 내용은 Amazon EC2 인스턴스에 대한 키 페어 생성을 참조하세요.
2. (선택 사항)업데이트를 가져오고 재부팅합니다.

   ```
   $ sudo yum -y update
              $  sudo reboot
   ```
3. 재부팅 후 EC2 인스턴스에 다시 연결합니다.
4. NFS 클라이언트를 설치합니다.

   Amazon Linux AMI 또는 Red Hat Linux AMI를 사용 중인 경우 다음 명령으로 NFS 클라이언트를 설치합니다.

   ```
   $ sudo yum -y install nfs-utils
   ```

   Ubuntu Amazon EC2 AMI를 사용 중인 경우 다음 명령으로 NFS 클라이언트를 설치합니다.

   ```
   $ sudo apt-get -y install nfs-common
   ```
5. 다음 명령을 사용하여 NFS 서비스를 시작합니다.

```
$ sudo service nfs-server start
```

6. 다음과 같이 NFS 서비스가 시작되었는지 확인합니다.

```
$ sudo service nfs-server status
Redirecting to /bin/systemctl status nfs.service
# nfs-server.service - NFS server and services
   Loaded: loaded (/usr/lib/systemd/system/nfs-server.service; disabled; vendor
 preset: disabled)
   Active: active (exited) since Wed 2019-10-30 16:13:44 UTC; 5s ago
  Process: 29446 ExecStart=/usr/sbin/rpc.nfsd $RPCNFSDARGS (code=exited, status=0/
SUCCESS)
  Process: 29441 ExecStartPre=/bin/sh -c /bin/kill -HUP `cat /run/gssproxy.pid`
 (code=exited, status=0/SUCCESS)
  Process: 29439 ExecStartPre=/usr/sbin/exportfs -r (code=exited, status=0/SUCCESS)
 Main PID: 29446 (code=exited, status=0/SUCCESS)
   CGroup: /system.slice/nfs-server.service
```

사용자 지정 커널을 사용하는 경우(사용자 지정 AMI를 빌드한 경우), 최소한 NFSv4.1 클라이언트 커널 모듈과 적절한 NFS4 사용자 공간 탑재 도우미를 포함해야 합니다.

> **Note**
>
> Amazon EC2 인스턴스를 시작할 때 Amazon Linux AMI 2016.03.0 또는 Amazon Linux AMI 2016.09.0을 선택하면 기본적으로 AMI에 이미 포함되어 있으므로 `nfs-utils`를 설치할 필요가 없습니다.

다음: 파일 시스템 탑재

다음 절차 중 하나에 따라 파일 시스템을 탑재합니다.

- DNS 이름을 사용하여 Amazon EC2에 탑재
- IP 주소를 사용하여 탑재
- EFS 파일 시스템 자동 탑재

권장 NFS 탑재 설정

다음 Linux 탑재 옵션 값을 사용하는 것이 좋습니다.

- `noresvport` – 네트워크 연결이 다시 설정될 때 NFS 클라이언트가 새로운 권한이 없는 전송 제어 프로토콜(TCP) 소스 포트를 사용하도록 지시합니다. 이전 버전의 Linux 커널(버전 v5.4 이하)에 포함된 NFS 클라이언트 소프트웨어에는 NFS 클라이언트가 연결이 끊기면 동일한 TCP 소스 포트에서 재연결을 시도하는 동작이 포함되어 있습니다. 이 동작은 TCP RFC를 준수하지 않으므로 이러한 클라이언트가 EFS 파일 시스템에 빠르게 다시 연결하지 못할 수 있습니다.

  `noresvport` 옵션을 사용하면 NFS 클라이언트가 EFS 파일 시스템에 투명하게 다시 연결되므로 네트워크 복구 이벤트 후 다시 연결할 때 중단 없는 가용성을 유지할 수 있습니다.

  > **Important**
  >
  > 재연결 또는 네트워크 복구 이벤트 후에도 EFS 파일 시스템의 가용성이 중단되지 않도록 하려면 `noresvport` 탑재 옵션을 사용하는 것이 좋습니다.
  > EFS 탑재 도우미를 사용하여 파일 시스템을 탑재하는 것을 고려해 보세요. EFS 탑재 도우미는 Amazon EFS 파일 시스템에 최적화된 NFS 탑재 옵션을 사용합니다.

- `rsize=1048576` – NFS 클라이언트가 각 네트워크 READ 요청에 대해 수신할 수 있는 최대 데이터 바이트 수를 설정합니다. 이 값은 EFS 파일 시스템의 파일에서 데이터를 읽을 때 적용됩니다. 성능이 저하되지 않도록 최대한 큰 크기(최대 `1048576`)를 사용하는 것이 좋습니다.
- `wsize=1048576` – NFS 클라이언트가 각 네트워크 WRITE 요청에 대해 전송할 수 있는 최대 데이터 바이트 수를 설정합니다. 이 값은 EFS 파일 시스템의 파일에 데이터를 쓸 때 적용됩니다. 성능이 저하되지 않도록 최대한 큰 크기(최대 `1048576`)를 사용하는 것이 좋습니다.
- `hard` – NFS 요청 시간이 초과된 후에는 NFS 클라이언트의 복구 동작을 설정하여 서버가 응답할 때까지 NFS 요청을 무기한 재시도합니다. 데이터 무결성을 위하여 하드 탑재 옵션(`hard`)을 사용하는 것이 좋습니다. 그러나 `soft` 탑재를 사용하는 경우 `timeo` 파라미터를 `150`데시초(15초) 이상으로 설정해야 합니다. 이렇게 하면 소프트 탑재에 고유한 데이터 손상 위험을 최소화하는 데 도움이 됩니다.
- `timeo=600` – NFS 클라이언트가 600데시초(60초) 후에 재시도하기 전에 NFS 클라이언트가 응답을 기다리는 데 사용하는 시간 초과 값을 설정합니다. 시간 제한 파라미터(`timeo`)를 변경해야 하는 경우 15초에 해당하는 `150` 이상의 값을 사용하는 것이 좋습니다. 이렇게 하면 원하지 않는 성능 저하를 방지할 수 있습니다.
- `retrans=2` – NFS 클라이언트가 추가 복구 작업을 시도하기 전에 요청을 재시도하는 횟수를 2로 설정합니다.

- `_netdev` – `/etc/fstab`에서 네트워크가 활성화될 때까지 클라이언트가 EFS 파일 시스템을 탑재하는 것을 방지합니다.
- `nofail` – 탑재된 EFS 파일 시스템의 상태와 관계없이 EC2 인스턴스를 시작해야 하는 경우, `/etc/fstab` 파일의 파일 시스템 항목에 `nofail` 옵션을 추가합니다.

위의 기본값을 사용하지 않는 경우 다음 사항에 유의하세요.

- 일반적으로 기본값과 다른 탑재 옵션을 설정하지 마세요. 성능 저하 및 기타 문제가 발생할 수 있습니다. 예를 들어, 읽기 또는 쓰기 버퍼 크기를 변경하거나 속성 캐싱을 비활성화하면 성능이 저하될 수 있습니다.
- Amazon EFS는 소스 포트를 무시합니다. Amazon EFS 소스 포트를 변경하더라도 어떠한 영향도 미치지 않습니다.
- Amazon EFS는 `nconnect` 탑재 옵션을 지원하지 않습니다.
- Amazon EFS는 Kerberos 보안 변형을 지원하지 않습니다. 예를 들어 다음 탑재 명령은 실패합니다.

```
$ mount -t nfs4 -o krb5p <DNS_NAME>:/ /efs/
```

- DNS 이름을 사용하여 파일 시스템을 탑재하는 것이 좋습니다. 이 이름은 Amazon EC2 인스턴스와 동일한 가용 영역에서 Amazon EFS 탑재 대상의 IP 주소로 해석됩니다. Amazon EC2 인스턴스와 다른 가용 영역에 있는 탑재 대상을 사용하면 가용 영역을 통해 전송된 데이터에 대한 표준 EC2 요금이 부과됩니다. 또한 파일 시스템 작업의 대기 시간이 늘어날 수 있습니다.
- 추가 탑재 옵션과 기본값에 대한 자세한 설명은 Linux 설명서를 참조하세요.

DNS 이름을 사용하여 Amazon EC2에 탑재

> **Note**
>
> 파일 시스템을 탑재하기 전에, EC2 보안 그룹에서 포트 2049를 통한 인바운드 NFS 접근을 허용하는 규칙을 탑재 대상 보안 그룹에 추가해야 합니다. 자세한 내용은 VPC 보안 그룹 사용 단원을 참조하십시오.

- 파일 시스템 DNS 이름 - 파일 시스템의 DNS 이름을 사용하는 것은 가장 간단한 탑재 옵션입니다. 파일 시스템 DNS 이름은 연결 중인 Amazon EC2 인스턴스의 가용 영역에서 탑재 대상의 IP 주소를 자동으로 확인합니다. DNS 이름은 콘솔에서 확인할 수 있으며, 파일 시스템 ID가 있는 경우에는 다음 규칙에 따라 파일 시스템 ID를 구성할 수 있습니다.

```
file-system-id.efs.aws-region.amazonaws.com
```

> **Note**
>
> 파일 시스템 DNS 이름을 위한 DNS 확인의 경우 클라이언트 인스턴스와 동일한 가용 영역에 있는 탑재 대상이 Amazon EFS 파일 시스템에 있어야 합니다.

- 파일 시스템 DNS 이름을 사용하면 다음 명령으로 Amazon EC2 Linux 인스턴스에 파일 시스템을 탑재할 수 있습니다.

```
sudo mount -t nfs -o
 nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport file-
system-id.efs.aws-region.amazonaws.com:/ /efs-mount-point
```

- 파일 시스템 DNS 이름을 사용하면 다음 명령으로 지원되는 macOS 버전(Big Sur, Monterey, Ventura)을 실행하는 Amazon EC2 Mac 인스턴스에 파일 시스템을 탑재할 수 있습니다.

```
sudo mount -t nfs -o
 nfsvers=4.0,rsize=65536,wsize=65536,hard,timeo=600,retrans=2,noresvport,mountport=2049 fil
system-id.efs.aws-region.amazonaws.com:/ /efs
```

> **Important**
>
> 지원되는 macOS 버전을 실행하는 EC2 Mac 인스턴스에 마운트할 때 EFS 파일 시스템에 성공적으로 연결하려면 `mountport=2049`를 사용해야 합니다.

- 탑재 대상 DNS 이름 - 2016년 12월에 파일 시스템 DNS 이름을 도입했습니다. 이전 버전과의 호환성을 위해 각 가용 영역 탑재 대상에 대한 DNS 이름을 계속해서 제공하고 있습니다. 탑재 대상 DNS 이름의 일반 형식은 다음과 같습니다.

```
availability-zone.file-system-id.efs.aws-region.amazonaws.com
```

> **Note**
>
> 가용 영역 전반의 탑재 대상 DNS 이름 확인이 지원됩니다.

일부 경우, 탑재 대상을 삭제한 후 동일한 가용 영역에 새 탑재 대상을 생성할 수 있습니다. 이 경우, 가용 영역에 새로 만든 탑재 대상의 DNS 이름은 기존 탑재 대상의 DNS 이름과 동일합니다.

연결 대화 상자에서 파일 시스템을 탑재하기 위한 정확한 명령을 보고 복사할 수 있습니다.

파일 시스템의 마운트 명령을 보려면

1. Amazon EFS 콘솔에서 탑재하려는 파일 시스템을 선택하여 세부 정보 페이지를 표시합니다.
2. 이 파일 시스템에 사용할 탑재 명령을 표시하려면 오른쪽 상단에서 연결을 선택합니다.

   연결 화면에는 파일 시스템을 탑재하는 데 사용할 정확한 명령이 표시됩니다.
3. 기본 DNS를 통한 탑재 보기에는 EFS 탑재 도우미 또는 NFS 클라이언트를 사용하여 탑재할 때 파일 시스템의 DNS 이름을 사용하여 파일 시스템을 탑재하는 명령이 표시됩니다.

Amazon EFS를 지원하는 목록은 AWS 리전의 Amazon Elastic File System을 참조하세요 AWS 일반 참조.

`mount` 명령에서 DNS 이름을 사용할 수 있으려면 다음에 해당해야 합니다.

- 연결 중인 EC2 인스턴스가 VPC 내에 있어야 하고, Amazon에서 제공하는 DNS 서버를 사용하도록 구성되어 있어야 합니다. Amazon DNS 서버에 대한 자세한 정보는 Amazon VPC 사용 설명서에서 Amazon VPC의 DHCP 옵션 세트를 참조하세요.
- 연결 중인 EC2 인스턴스의 VPC에는 DNS 확인 및 DNS 호스트 이름이 활성화되어 있어야 합니다. 자세한 내용은 Amazon VPC 사용 설명서의 VPC에 대한 DNS 속성을 참조하세요.
- 연결 중인 EC2 인스턴스는 EFS 파일 시스템과 동일한 VPC 내에 있어야 합니다. 다른 위치 또는 다른 VPC에서 파일 시스템에 액세스하고 탑재하는 방법에 대한 자세한 내용은 사전 조건 및 자습서: 다른 VPC에서 파일 시스템 탑재하기 섹션을 참조하세요.

> **Note**
>
> 탑재 대상을 만든 후 파일 시스템을 탑재하기 전에 90초 가량 기다리는 것이 좋습니다. 이 대기를 통해 DNS 레코드가 파일 시스템이 AWS 리전 있는에 완전히 전파됩니다.

IP 주소를 사용하여 탑재

DNS 이름을 사용하여 Amazon EFS 파일 시스템을 탑재하는 방법의 대안으로 Amazon EC2 인스턴스에서 탑재 대상의 IP 주소를 사용하여 파일 시스템을 탑재할 수 있습니다. IP 주소를 통한 탑재는 DNS 호스트 이름이 비활성화된 VPC와 같이 DNS가 비활성화된 환경에서 작동합니다.

탑재 대상 IP 주소를 사용하여 파일 시스템 탑재는 기본적으로 DNS 이름을 사용하여 파일 시스템을 탑재하도록 구성된 애플리케이션의 폴백 옵션으로 구성할 수도 있습니다. 탑재 대상 IP 주소에 연결할 때 EC2 인스턴스는 연결 중인 인스턴스와 동일한 가용 영역에서 탑재 대상 IP 주소를 사용하여 탑재해야 합니다.

연결 대화 상자에서 파일 시스템을 탑재하기 위한 정확한 명령을 보고 복사할 수 있습니다.

> **Note**
>
> 파일 시스템을 탑재하기 전에, EC2 보안 그룹에서 포트 2049를 통한 인바운드 NFS 액세스를 허용하도록 탑재 대상 보안 그룹에 규칙을 추가해야 합니다. 자세한 내용은 VPC 보안 그룹 사용 단원을 참조하십시오.

탑재 대상 IP 주소를 사용하여 EFS 파일 시스템을 탑재하기 위한 정확한 명령을 보고 복사하려면

1. Amazon Elastic File System 콘솔(https://console.aws.amazon.com/efs/)을 엽니다.
2. Amazon EFS 콘솔에서 탑재하려는 파일 시스템을 선택하여 세부 정보 페이지를 표시합니다.
3. 이 파일 시스템에 사용할 탑재 명령을 표시하려면 오른쪽 상단에서 연결을 선택합니다.
4. 연결 화면에는 파일 시스템을 탑재하는 데 사용할 정확한 명령이 표시됩니다.

   선택한 가용 영역에 NFS 클라이언트가 있는 탑재 대상 IP 주소를 사용하여 파일 시스템을 탑재하는 명령을 표시하려면 IP를 통한 탑재(Mount via IP)를 선택합니다.

- `mount` 명령에서 탑재 대상의 IP 주소를 사용하여 다음 명령으로 Amazon EC2 Linux 인스턴스에 파일 시스템을 탑재할 수 있습니다.

```
sudo mount -t nfs -o
 nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport mount-target-IP:/   /efs
```

- `mount` 명령에서 탑재 대상의 IP 주소를 사용하여 다음 명령으로 macOS Big Sur를 실행하는 Amazon EC2 Mac 인스턴스에 파일 시스템을 탑재할 수 있습니다.

```
sudo mount -t nfs -o
 nfsvers=4.0,rsize=65536,wsize=65536,hard,timeo=600,retrans=2,noresvport,mountport=2049 mount
target-IP:/ /efs
```

> **Important**
>
> macOS Big Sur를 실행하는 EC2 Mac 인스턴스에 마운트할 때 EFS 파일 시스템에 성공적으로 연결하려면 `mountport=2049`를 사용해야 합니다.

에서 IP 주소를 사용하여 탑재 AWS CloudFormation

CloudFormation 템플릿의 IP 주소를 사용하여 파일 시스템을 탑재할 수도 있습니다. 자세한 내용은 GitHub의 커뮤니티 제공 구성 파일에 대한 awsdocs/elastic-beanstalk-samples 리포지토리의 storage-efs-mountfilesystem-ip-addr.config를 참조하세요.

EFS 파일 시스템 자동 탑재

EFS 탑재 도우미 또는 NFS를 사용하여 인스턴스가 시작될 때 자동으로 EFS 파일 시스템을 탑재하도록 Amazon EC2 인스턴스를 구성할 수 있습니다.

- EFS 탑재 도우미 사용:
  - EC2 런치 인스턴스 마법사를 사용하여 새 EC2 Linux 인스턴스를 생성할 때 EFS 파일 시스템을 연결합니다.
  - EFS 파일 시스템에 대한 항목으로 EC2 `/etc/fstab` 파일을 업데이트합니다.
- EC2 Linux 및 Mac 인스턴스의 경우, EFS 탑재 도우미 없이 NFS를 사용하여 EC2 `/etc/fstab` 파일을 업데이트합니다.

> **Note**
>
> EFS 탑재 도우미는 macOS Big Sur 또는 Monterey를 실행하는 EC2 Mac 인스턴스에서의 자동 탑재를 지원하지 않습니다. 대신 NFS를 사용하여 EC2 Mac 인스턴스의 /etc/fstab 파일을 구성하여 EFS 파일 시스템을 자동으로 탑재할 수 있습니다.

주제

- 새 EC2 Linux 인스턴스에서 자동 탑재 활성화
- 기존 EC2 Linux 인스턴스에서 자동 탑재 활성화
- NFS를 사용하여 EC2 Linux 또는 Mac 인스턴스에서 자동 탑재 활성화

새 EC2 Linux 인스턴스에서 자동 탑재 활성화

Amazon EC2 인스턴스 시작 마법사를 사용하여 새 EC2 Linux 인스턴스를 생성할 때 Amazon EFS 파일 시스템을 자동으로 탑재하도록 구성할 수 있습니다. EC2 인스턴스는 인스턴스가 처음 시작될 때와 다시 시작될 때마다 파일 시스템을 자동으로 탑재합니다.

이 방법은 EFS 탑재 도우미를 사용하여 EC2 인스턴스에 파일 시스템 업데이트 /etc/fstab 파일을 탑재합니다. 탑재 도우미는 `amazon-efs-utils` 도구 세트의 일부입니다.

> **Note**
>
> EFS 파일 시스템은 인스턴스 시작 시 macOS Big Sur 또는 Monterey를 실행하는 EC2 Mac 인스턴스에 탑재하는 것을 지원하지 않습니다.

> **Note**
>
> Microsoft Windows 기반 EC2 인스턴스에는 Amazon EFS를 사용할 수 없습니다.

EC2 인스턴스를 시작하고 연결하려면 먼저 키 페어를 생성해야 합니다. 키 페어를 생성하는 방법에 대한 자세한 내용은 Amazon EC2 사용 설명서에서 Amazon EC2 키 페어 및 Amazon EC2 인스턴스를 참조하세요.

EC2 인스턴스가 시작 시 자동으로 EFS 파일 시스템을 탑재하도록 구성

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 인스턴스 시작을 선택합니다.
3. 1단계: Amazon Machine Image(AMI) 선택)에서 목록 맨 위의 Amazon Linux AMI를 찾아 선택을 선택합니다.
4. 2단계: 인스턴스 유형 선택에서 다음: 인스턴스 세부 정보 구성을 선택합니다.
5. 3단계: 인스턴스 세부 정보 구성에서 다음 정보를 제공합니다.

- 네트워크의 경우 탑재하려는 EFS 파일 시스템과 동일한 VPC에 대한 항목을 선택합니다.
- 서브넷의 경우 모든 가용 영역의 기본 서브넷을 선택합니다.
- 파일 시스템의 경우 탑재하려는 EFS 파일 시스템을 선택합니다. 파일 시스템 ID 옆에 표시된 경로는 EC2 인스턴스가 사용하게 될 탑재 지점입니다. 이 경로는 사용자가 변경할 수 있습니다.
- 고급 세부 정보에서, 사용자 데이터를 자동으로 생성할 수 있으며 파일 시스템에서 지정한 EFS 파일 시스템을 탑재하는 데 필요한 명령도 확인할 수 있습니다.

6. 다음: 스토리지 추가를 선택합니다.
7. 다음: 태그 추가를 선택합니다.
8. 인스턴스 이름을 지정한 후 다음: 보안 그룹 구성을 선택합니다.
9. 6단계: 보안 그룹 구성에서 보안 그룹 할당을 기존 보안 그룹 선택으로 설정합니다. 기본 보안 그룹을 선택하고 EFS 파일 시스템에 액세스할 수 있는지 확인합니다.

   이 보안 그룹을 사용하여 Secure Shell(SSH)로 EC2 인스턴스에 액세스할 수 없습니다. SSH 액세스의 경우, 나중에 기본 보안을 편집하고 SSH를 허용하는 규칙을 추가하거나 SSH를 허용하는 새 보안 그룹을 추가할 수 있습니다. 다음 설정을 사용할 수 있습니다.

   - 유형: SSH
   - 프로토콜: TCP
   - 포트 범위: 22
   - 소스: 위치 무관 0.0.0.0/0

10. 검토 및 시작을 선택합니다.
11. 시작을 선택합니다.
12. 생성한 키 페어의 확인란을 선택한 다음 인스턴스 시작을 선택합니다.

이제 EC2 인스턴스가 시작될 때와 재부팅될 때마다 EFS 파일 시스템을 탑재하도록 구성되었습니다.

기존 EC2 Linux 인스턴스에서 자동 탑재 활성화

`/etc/fstab` 파일에는 파일 시스템에 대한 정보가 들어 있습니다. 인스턴스 시작 중에 실행되는 `mount -a` 명령은 `/etc/fstab`에 나열된 파일 시스템을 탑재합니다. 이 절차에서는 인스턴스가 다시 시작될 때 EFS 탑재 도우미를 사용하여 EFS 파일 시스템을 자동으로 다시 탑재하도록 Amazon EC2 Linux 인스턴스에서 `/etc/fstab`를 수동으로 업데이트합니다.

> **Note**
>
> EFS 파일 시스템은 macOS Big Sur 또는 Monterey를 실행하는 EC2 Mac 인스턴스에서 EFS 마운트 도우미와 함께 `/etc/fstab`을 사용하는 자동 탑재를 지원하지 않습니다. 대신 NFS와 `/etc/fstab`을 사용하여 macOS Big Sur 및 Monterey를 실행하는 EC2 Mac 인스턴스에 파일 시스템을 자동으로 탑재할 수 있습니다.

이 두 방법 모두 EFS 탑재 도우미를 사용하여 파일 시스템을 탑재합니다. 탑재 도우미는 `amazon-efs-utils` 도구 세트의 일부입니다.

`amazon-efs-utils` 도구는 Amazon Linux와 Amazon Linux 2 Amazon Machine Image(AMI)에 설치할 수 있습니다. `amazon-efs-utils`에 대한 자세한 정보는 Amazon EFS 클라이언트 수동 설치 섹션을 참조하세요. Red Hat Enterprise Linux(RHEL)와 같은 다른 Linux 배포판을 사용하는 경우, `amazon-efs-utils`를 수동으로 빌드하고 설치하세요. 자세한 내용은 다른 Linux 배포판에 Amazon EFS 클라이언트 설치 단원을 참조하십시오.

사전 조건

이 절차를 성공적으로 구현하려면 다음 요구 사항을 충족해야 합니다.

- 자동으로 다시 탑재하려는 EFS 파일 시스템을 이미 생성했습니다. 자세한 내용은 콘솔을 사용하여 빠른 생성 단원을 참조하십시오.
- EFS 파일 시스템을 자동으로 다시 탑재하도록 구성하려는 EC2 Linux 인스턴스를 이미 생성했습니다.
- EFS 탑재 도우미는 EC2 Linux 인스턴스에 설치됩니다. 자세한 내용은 Amazon EFS 클라이언트 수동 설치 단원을 참조하십시오.

/etc/fstab 파일 업데이트

인스턴스가 다시 시작될 때 인스턴스가 EFS 탑재 도우미를 사용하여 EFS 파일 시스템을 자동으로 다시 탑재하도록 EC2 Linux 인스턴스에서 /etc/fstab을 수동으로 업데이트하려면 다음 단계를 수행합니다.

EC2 인스턴스의 /etc/fstab 파일 업데이트

1. EC2 인스턴스에 연결합니다. 자세한 내용은 Amazon EC2 사용 설명서의 EC2 인스턴스에 연결을 참조하세요.

2. 편집기에서 /etc/fstab 파일을 엽니다.
3. IAM 권한 부여 또는 EFS 액세스 포인트를 사용하여 자동으로 탑재하려면:

   - 인스턴스 프로파일이 있는 EC2 인스턴스에 IAM 권한을 부여하여 자동으로 탑재하려면 /etc/fstab 파일에 다음 줄을 추가합니다.

     ```
     file-system-id:/ efs-mount-point efs _netdev,noresvport,tls,iam 0 0
     ```

   - 보안 인증 파일을 사용하여 Linux 인스턴스에 IAM 권한을 부여하여 자동으로 탑재하려면 /etc/fstab 파일에 다음 줄을 추가합니다.

     ```
     file-system-id:/ efs-mount-point efs
      _netdev,noresvport,tls,iam,awsprofile=namedprofile 0 0
     ```

   - EFS 액세스 포인트를 사용하여 파일 시스템을 자동으로 탑재하려면 /etc/fstab 파일에 다음 줄을 추가합니다.

     ```
     file-system-id:/ efs-mount-point efs _netdev,noresvport,tls,accesspoint=access-
     point-id 0 0
     ```

   > **Warning**
   >
   > 파일 시스템을 자동으로 마운트하는 경우 네트워크 파일 시스템 식별에 사용하는 _netdev 옵션을 사용합니다. _netdev이 빠진 경우 EC2 인스턴스가 응답을 중지합니다. 컴퓨팅 인스턴스가 네트워킹을 시작한 후 네트워크 파일 시스템의 초기화를 완료해야 하기 때문입니다. 자세한 내용은 자동 탑재 실패 및 인스턴스 무응답 단원을 참조하십시오.

   자세한 내용은 IAM 권한 부여를 통한 탑재 및 EFS 액세스 포인트를 사용한 탑재 섹션을 참조하세요.
4. 파일에 대한 변경 사항을 저장합니다.

> **Note**
>
> 경우에 따라 탑재된 EC2 파일 시스템의 상태에 관계없이 EC2 인스턴스를 시작해야 할 수 있습니다. 이 경우 /etc/fstab 파일의 파일 시스템 항목에 nofail 옵션을 추가하세요.

/etc/fstab 파일에 추가한 코드 줄은 다음 작업을 수행합니다.

필드	설명
file-system-id :/	EFS 파일 시스템의 ID. 콘솔에서 또는 CLI 또는 AWS SDK에서 프로그래밍 방식으로이 ID를 가져올 수 있습니다.
efs-mount-point	EC2 인스턴스의 EFS 파일 시스템 탑재 지점
efs	파일 시스템의 유형 탑재 도우미를 사용하는 경우 이 유형은 항상 efs입니다.
mount options	파일 시스템의 탑재 옵션 다음 옵션의 쉼표로 구분된 목록입니다. • _netdev – 운영 체제에 파일 시스템을 네트워크 액세스를 요구하는 장치에 위치시키라고 명령하는 옵션입니다. 클라이언트에서 네트워크가 활성화되기 전에 인스턴스가 파일 시스템을 마운트하는 것을 방지하는 옵션입니다. • noresvport – 네트워크 연결이 다시 설정될 때 NFS 클라이언트가 새로운 TCP 소스 포트를 사용하도록 지시합니다. 이렇게 하면 네트워크 복구 이벤트 후에도 EFS 파일 시스템을 중단 없이 사용할 수 있습니다. • tls – 전송 중인 데이터의 암호화를 활성화합니다. • iam – 인스턴스 프로파일이 있는 EC2 인스턴스에 IAM 권한을 부여하여 탑재하려면 이 옵션을 사용합니다. iam 탑재 옵션을 사용하려면 tls 옵션도 사용해야 합니다. 자세한 내용은 IAM을 사용하여 파일 시스템에 대한 액세스 제어 단원을 참조하십시오. • awsprofile= *namedprofile* – 보안 인증 파일을 사용하여 Linux 인스턴스에 IAM 권한을 부여하여 탑재하려면 이 옵션을 iam 및 tls 옵션과 함께 사용합니다. EFS 액세스 포인트에 대한 자세한 내용은 IAM을 사용하여 파일 시스템에 대한 액세스 제어 섹션을 참조하세요. • accesspoint= *access-point-id* – EFS 액세스 포인트를 사용하여 탑재하려면 이 옵션을 tls 옵션과 함께 사용합니다. EFS 액세스 포인트에 대한 자세한 내용은 액세스 포인트 작업 섹션을 참조하세요.

필드	설명
0	0이 아닌 값은 파일 시스템을 dump로 백업해야 함을 나타냅니다. EFS의 경우 이 값은 0이어야 합니다.
0	부팅 시 fsck가 파일 시스템을 검사하는 순서입니다. EFS 파일 시스템의 경우 이 값을 0으로 하여 시작 시 fsck가 실행되지 않도록 해야 합니다.

NFS를 사용하여 EC2 Linux 또는 Mac 인스턴스에서 자동 탑재 활성화

EC2 Linux 및 Mac 인스턴스의 경우, EFS 탑재 도우미 없이 NFS를 사용하여 Amazon EC2 /etc/fstab 파일을 업데이트합니다.

EC2 인스턴스의 **/etc/fstab** 파일을 업데이트하려면

1. EC2 인스턴스에 연결합니다. 자세한 내용은 Amazon EC2 사용 설명서의 EC2 인스턴스에 연결을 참조하세요.
2. 편집기에서 /etc/fstab 파일을 엽니다.
3. EFS 탑재 도우미 대신 NFS를 사용하여 파일 시스템을 자동으로 탑재하려면 /etc/fstab 파일에 다음 줄을 추가합니다.

   - *file_system_id*를 탑재하려는 파일 시스템의 ID로 바꾸세요.
   - *aws-region*을와 같이 파일 시스템이 AWS 리전 있는 로 바꿉니다us-east-1.
   - *mount_point#* 파일 시스템의 탑재 지점으로 바꾸세요.

```
file_system_id.efs.aws-region.amazonaws.com:/ mount_point nfs4
 nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport,_netdev
 0 0
```

/etc/fstab 파일에 추가한 코드 줄은 다음 작업을 수행합니다.

필드	설명
file-system-id :/	EFS 파일 시스템의 ID. 콘솔에서 또는 CLI 또는 AWS SDK에서 프로그래밍 방식으로이 ID를 가져올 수 있습니다.
efs-mount-point	EC2 인스턴스의 EFS 파일 시스템 탑재 지점
nfs4	파일 시스템 유형을 지정합니다.
mount options	파일 시스템에 대한 탑재 옵션을 쉼표로 구분한 목록입니다. • nfsvers=4.1 – NFS v4.1을 사용하여 지정합니다. • rsize=1048576 – 성능 향상을 위해 EFS 파일 시스템의 파일에서 데이터를 읽을 때 NFS 클라이언트가 각 네트워크 READ 요청에 대해 수신할 수 있는 최대 데이터 바이트 수를 설정합니다. 1048576은 가능한 최대 크기입니다. • wsize=1048576 – 성능 향상을 위해 EFS 파일 시스템의 파일에서 데이터를 쓸 때 NFS 클라이언트가 각 네트워크 WRITE 요청에 대해 송신할 수 있는 최대 데이터 바이트 수를 설정합니다. 1048576은 가능한 최대 크기입니다. • hard – NFS 요청 시간이 초과된 후에는 NFS 클라이언트의 복구 동작을 설정하여 서버가 응답할 때까지 NFS 요청을 무기한 재시도합니다. 데이터 무결성을 위하여 하드 탑재 옵션(hard)을 사용하는 것이 좋습니다. 그러나 soft 탑재를 사용하는 경우 timeo 파라미터를 150데시초(15초) 이상으로 설정해야 합니다. 이렇게 하면 소프트 탑재에 고유한 데이터 손상 위험을 최소화하는 데 도움이 됩니다. • timeo=600 – NFS 클라이언트가 600데시초(60초) 후에 재시도하기 전에 NFS 클라이언트가 응답을 기다리는 데 사용하는 시간 초과 값을 설정합니다. 시간 제한 파라미터(timeo)를 변경해야 하는 경우 15초에 해당하는 150 이상의 값을 사용하는 것이 좋습니다. 이렇게 하면 원하지 않는 성능 저하를 방지할 수 있습니다. • retrans=2 – NFS 클라이언트가 추가 복구 작업을 시도하기 전에 요청을 재시도하는 횟수를 2로 설정합니다. • noresvport – 네트워크 연결이 다시 설정될 때 NFS 클라이언트가 새로운 TCP 소스 포트를 사용하도록 지시합니다. 이렇게 하면 네트

필드	설명
	워크 복구 이벤트 후에도 EFS 파일 시스템을 중단 없이 사용할 수 있습니다. • _netdev – 네트워크가 활성화될 때까지 클라이언트가 EFS 파일 시스템을 탑재하는 것을 방지합니다.
0	dump 값을 지정합니다. 0은 dump 유틸리티에 파일 시스템을 백업하지 않도록 지시합니다.
0	시작 시 fsck 유틸리티를 실행하지 않도록 지시합니다.

파일 시스템 탑재 해제

파일 시스템을 삭제하기 전에 파일 시스템이 연결된 모든 Amazon EC2 인스턴스에서 파일 시스템을 탑재 해제하는 것이 좋습니다. 인스턴스 자체에서 umount 명령을 실행하여 Amazon EC2 인스턴스에서 파일 시스템의 탑재를 해제할 수 있습니다. AWS CLI, AWS Management Console또는 AWS SDKs를 통해 EFS 파일 시스템의 탑재를 해제할 수 없습니다. Linux를 실행하는 EC2 인스턴스에 연결된 EFS 파일 시스템을 탑재 해제하려면 다음과 같이 umount 명령을 사용하십시오.

```
umount /mnt/efs
```

다른 umount 옵션은 지정하지 않는 것이 좋습니다. 기본값과 다른 그 밖의 umount 옵션은 설정하지 않는 것이 좋습니다.

df 명령을 실행하여 EFS 파일 시스템이 탑재 해제되었는지 확인할 수 있습니다. 이 명령은 Linux 기반 Amazon EC2 인스턴스에 현재 탑재된 파일 시스템에 대한 디스크 사용량 통계를 표시합니다. 탑재 해제하려는 EFS 파일 시스템이 df 명령 출력에 나열되어 있지 않으면 그 파일 시스템은 탑재 해제된 것입니다.

Example– EFS 파일 시스템의 탑재 상태 식별 및 탑재 해제

```
$ df -T
Filesystem Type 1K-blocks Used Available Use% Mounted on
/dev/sda1 ext4 8123812 1138920 6884644 15% /
availability-zone.file-system-id.efs.aws-region.amazonaws.com :/ nfs4 9007199254740992
 0 9007199254740992 0% /mnt/efs
```

```
$ umount /mnt/efs
```

```
$ df -T
```

```
Filesystem Type 1K-blocks Used Available Use% Mounted on
/dev/sda1 ext4 8123812 1138920 6884644 15% /
```

자습서:를 사용하여 EFS 파일 시스템을 생성하고 EC2 인스턴스에 탑재 AWS CLI

암호화된 EFS 파일 시스템을 생성하고 VPC의 EC2 인스턴스에 탑재한 다음 AWS CLI를 사용하여 설정을 테스트합니다.

> **Note**
>
> 시작하기 자습서에서는 콘솔을 사용하여 Amazon EC2 및 EFS 리소스를 생성합니다. 이 자습서에서는 AWS CLI 를 사용하여 동일한 작업을 수행합니다. 주로 Amazon EFS API에 익숙해집니다.

이 자습서에서는 계정에 다음 AWS 리소스를 생성합니다.

- Amazon EC2 리소스:
  - 보안 그룹 2개(EC2 인스턴스 및 EFS 파일 시스템용)

    적절한 인바운드/아웃바운드 액세스를 승인하는 규칙을 이러한 보안 그룹에 추가합니다. 이렇게 하면 표준 NFSv4.1 TCP 포트를 사용하여 탑재 대상을 통해 EC2 인스턴스를 파일 시스템에 연결할 수 있습니다.
  - VPC의 EC2 인스턴스
- Amazon EFS 리소스:
  - 파일 시스템.
  - 파일 시스템의 탑재 대상입니다.

    EC2 인스턴스에 파일 시스템을 탑재하려면 VPC에서 탑재 대상을 만들어야 합니다. VPC의 각 가용 영역에 탑재 대상을 하나씩 만들 수 있습니다. 자세한 내용은 Amazon EFS 작동 방식 단원을 참조하십시오.

그런 다음 EC2 인스턴스에서 파일 시스템을 테스트합니다. 자습서의 마지막에 나오는 정리 단계에서는 이러한 리소스를 제거하는 방법을 설명합니다.

이 자습서에서는 이러한 모든 리소스를 미국 서부(오리건) 리전(`us-west-2`)에 생성합니다. 어떤 방법을 사용 AWS 리전 하든 일관되게 사용해야 합니다. VPC, EC2 리소스, EFS 리소스 등 모든 리소스가 동일한 AWS 리전에 있어야 합니다.

주제

- 사전 조건
- 설정 AWS CLI
- 1단계: EC2 리소스 생성
- 2단계: EFS 리소스 생성
- 3단계: EC2 인스턴스에 파일 시스템 탑재 및 테스트
- 4단계: 정리

사전 조건

- 의 루트 자격 증명을 사용하여 콘솔에 로그인하고 시작하기 연습을 시도 AWS 계정 할 수 있습니다. 그러나 AWS Identity and Access Management (IAM)에서는의 루트 자격 증명을 사용하지 않을 것을 권장합니다 AWS 계정. 대신 계정에서 관리자 사용자를 만들어 해당 보안 인증을 사용하여 계정에서 리소스를 관리합니다. 자세한 내용은 AWS IAM Identity Center 사용 설명서의 IAM Identity Center 사용자에 대한 AWS 계정 액세스 할당을 참조하세요.
- 자신의 계정에서 만든 기본 VPC 또는 사용자 지정 VPC를 사용할 수 있습니다. 이 연습에서는 기본 VPC 구성을 사용합니다. 단, 사용자 지정 VPC를 사용하는 경우 다음을 확인하세요.
  - DNS 호스트 이름이 사용됩니다. 자세한 내용을 알아보려면 Amazon VPC 사용 설명서의 VPC의 DNS 속성을 참조하세요.
  - 인터넷 게이트웨이가 VPC에 연결되어 있습니다. 자세한 내용은 Amazon VPC 사용 설명서의 인터넷 게이트웨이를 사용한 인터넷 연결을 참조하세요.
  - VPC 서브넷이 VPC 서브넷에서 시작된 인스턴스의 퍼블릭 IP 주소를 요청하도록 구성되어 있습니다. 자세한 내용은 Amazon VPC 사용 설명서의 VPC 및 서브넷의 IP 주소 지정을 참조하세요.
  - VPC 라우팅 테이블에 모든 인터넷 바인딩된 트래픽을 인터넷 게이트웨이로 보내는 규칙이 포함되어 있습니다.
- 를 설정하고 관리자 프로필을 AWS CLI 추가해야 합니다.

설정 AWS CLI

다음 지침에 따라 AWS CLI 및 사용자 프로필을 설정합니다.

를 설정하려면 AWS CLI

1. AWS CLI를 다운로드하고 구성합니다. 지침은 AWS Command Line Interface 사용 설명서의 AWS CLI시작하기를 참조하세요.
2. 프로필을 설정합니다.

   사용자 자격 증명을 AWS CLI `config` 파일에 저장합니다. 이 자습서의 예제 CLI 명령은 adminuser 프로필을 지정합니다. `config` 파일에 adminuser 프로필을 생성합니다. 또 다음과 같이 `config` 파일에 관리자 사용자 프로필을 기본값으로 설정할 수도 있습니다.

   ```
   [profile adminuser]
   aws_access_key_id = admin user access key ID
   aws_secret_access_key = admin user secret access key
   region = us-west-2

   [default]
   aws_access_key_id = admin user access key ID
   aws_secret_access_key = admin user secret access key
   region = us-west-2
   ```

   앞의 프로필은 기본값도 설정합니다 AWS 리전. CLI 명령에서 리전을 지정하지 않으면 us-west-2 리전으로 간주됩니다.
3. 명령 프롬프트에 다음 명령을 입력하여 설정을 확인합니다. 이들 명령은 명시적으로 보안 인증을 제공하지 않으므로 기본 프로필의 자격 증명이 사용됩니다.

   - help 명령 사용해 보기

     또 `--profile` 파라미터를 사용하여 명시적으로 사용자 프로필을 지정할 수 있습니다.

     ```
     aws help
     ```

     ```
     aws help \
     --profile adminuser
     ```

1단계: EC2 리소스 생성

이 단계에서는 다음 작업을 수행합니다.

- 보안 그룹 2개를 만듭니다.
- 추가 액세스를 승인하는 규칙을 보안 그룹에 추가합니다.
- EC2 인스턴스를 시작합니다. 다음 단계에서는 이 인스턴스에 EFS 파일 시스템을 생성하여 탑재합니다.

1.1단계: 보안 그룹 2개 만들기

이 단원에서는 EC2 인스턴스 및 EFS 탑재 대상의 VPC에 보안 그룹을 생성합니다. 이 자습서의 후반부에서는 이러한 보안 그룹을 EC2 인스턴스 및 EFS 탑재 대상에 할당합니다. 보안 그룹에 대한 자세한 내용은 Linux 인스턴스에 대한 Amazon EC2 보안 그룹을 참조하세요.

보안 그룹 생성

1. `create-security-group` CLI 명령을 사용해 두 개의 보안 그룹을 생성합니다.

   a. EC2 인스턴스에 대한 보안 그룹(`efs-walkthrough1-ec2-sg`)을 생성하고 VPC ID를 제공합니다.

   ```
   $ aws ec2 create-security-group \
   --region us-west-2 \
   --group-name efs-walkthrough1-ec2-sg \
   --description "Amazon EFS walkthrough 1, SG for EC2 instance" \
   --vpc-id vpc-id-in-us-west-2 \
   --profile adminuser
   ```

   보안 그룹 ID를 기록합니다. 다음은 응답의 예입니다.

   ```
   {
       "GroupId": "sg-aexample"
   }
   ```

   다음 명령을 사용하여 VPC ID를 찾을 수 있습니다.

   ```
   $ aws  ec2 describe-vpcs
   ```

b. EFS 탑재 대상에 대한 보안 그룹(efs-walkthrough1-mt-sg)을 생성합니다. VPC ID를 제공해야 합니다.

```
$ aws ec2 create-security-group \
--region us-west-2 \
--group-name efs-walkthrough1-mt-sg \
--description "Amazon EFS walkthrough 1, SG for mount target" \
--vpc-id vpc-id-in-us-west-2 \
--profile adminuser
```

보안 그룹 ID를 기록합니다. 다음은 응답의 예입니다.

```
{
    "GroupId": "sg-aexample"
}
```

2. 보안 그룹을 확인합니다.

```
aws ec2 describe-security-groups \
--group-ids list of security group IDs separated by space \
--profile adminuser \
--region us-west-2
```

두 보안 그룹에는 모든 트래픽이 나가도록 허용하는 아웃바운드 규칙 하나를 갖고 있습니다.

다음 섹션에서 다음 사항을 활성화하는 추가 액세스 권한을 승인합니다.

- EC2 인스턴스에 연결되도록 활성화합니다.
- EC2 인스턴스와 EFS 탑재 대상 간 트래픽을 활성화합니다(이 자습서의 후반부에서 이 보안 그룹을 연결함).

1.2단계: 인바운드/아웃바운드 액세스를 승인하는 규칙을 보안 그룹에 추가

이 단계에서는 인바운드/아웃바운드 액세스를 승인하는 규칙을 보안 그룹에 추가합니다.

규칙 추가

1. 모든 호스트에서 SSH를 사용해 EC2 인스턴스에 연결할 수 있도록 EC2 인스턴스(efs-walkthrough1-ec2-sg)의 보안 그룹에 대해 수신 Secure Shell(SSH) 연결을 승인합니다.

```
$ aws ec2 authorize-security-group-ingress \
--group-id id of the security group created for EC2 instance \
--protocol tcp \
--port 22 \
--cidr 0.0.0.0/0 \
--profile adminuser \
--region us-west-2
```

보안 그룹에 추가한 인바운드 및 아웃바운드 규칙이 있는지 확인합니다.

```
aws ec2 describe-security-groups \
--region us-west-2 \
--profile adminuser \
--group-id security-group-id
```

2. EFS 탑재 대상(efs-walkthrough1-mt-sg)의 보안 그룹에 대해 인바운드 액세스 권한을 승인합니다.

   명령 프롬프트에서 관리자 프로파일을 사용하여 다음 AWS CLI authorize-security-group-ingress 명령을 실행하여 인바운드 규칙을 추가합니다.

```
$ aws ec2 authorize-security-group-ingress \
--group-id ID of the security group created for Amazon EFS mount target \
--protocol tcp \
--port 2049 \
--source-group ID of the security group created for EC2 instance \
--profile adminuser \
--region us-west-2
```

3. 이제 두 보안 그룹이 인바운드 액세스를 승인하는지 확인합니다.

```
aws ec2 describe-security-groups \
--group-names efs-walkthrough1-ec2-sg   efs-walkthrough1-mt-sg \
--profile adminuser \
--region us-west-2
```

1.3단계: EC2 인스턴스 시작

이 단계에서는 EC2 인스턴스를 시작합니다.

EC2 인스턴스 시작

1. EC2 인스턴스 시작 시 제공해야 하는 다음 정보를 수집합니다.

   - 키 페어 이름. 키 페어 생성 지침은 Amazon EC2 사용 설명서의 Amazon EC2 인스턴스에 대한 키 페어 생성을 참조하세요.
   - 시작하려는 Amazon Machine Image(AMI)의 ID입니다.

     EC2 인스턴스를 시작하는 데 사용하는 AWS CLI 명령에는 파라미터로 배포하려는 Amazon Machine Image(AMI)의 ID가 필요합니다. 이 연습에서는 Amazon Linux HVM AMI를 사용합니다.

     > **Note**
     >
     > 대부분의 범용 Linux 기반 AMI를 사용할 수 있습니다. 다른 Linux AMI를 사용하는 경우, 배포의 패키지 관리자를 사용하여 인스턴스에 NFS 클라이언트를 설치해야 합니다. 또한 필요할 때 소프트웨어 패키지를 추가해야 할 수도 있습니다.

     Amazon Linux HVM AMI의 경우 Amazon Linux AMI에서 최신 ID를 찾을 수 있습니다. 다음과 같이 Amazon Linux AMI ID 테이블에서 ID 값을 선택합니다.

     - US West Oregon(미국 서부 오레곤) 리전을 선택합니다. 이 연습은 미국 서부(오레곤) 리전(us-west-2)에 모든 리소스를 생성했다고 가정합니다.
     - EBS 지원 HVM 64비트 유형을 선택합니다(CLI 명령에서 인스턴스 스토어를 지원하지 않는 `t2.micro` 인스턴스 유형을 지정하기 때문).

   - EC2 인스턴스에 생성한 보안 그룹 ID.
   - AWS 리전. 이 연습에서는 us-west-2 리전을 사용합니다.
   - 인스턴스를 시작하려는 VPC 서브넷 ID. `describe-subnets` 명령을 사용해 서브넷 목록을 가져올 수 있습니다.

     ```
     $ aws ec2 describe-subnets \
     --region us-west-2 \
     --filters "Name=vpc-id,Values=vpc-id" \
     --profile adminuser
     ```

     서브넷 ID를 선택한 후 `describe-subnets` 결과의 다음 값을 기록합니다.

- 서브넷 ID – 탑재 대상 생성 때 이 값이 필요합니다. 이 연습에서는 EC2 인스턴스를 시작한 동일한 서브넷에 탑재 대상을 생성합니다.
- 서브넷 가용 영역 – 이 값은 EC2 인스턴스에 파일 시스템을 탑재할 때 사용하는 탑재 대상의 DNS 이름을 구성하기 위해 필요합니다.

2. 다음 AWS CLI `run-instances` 명령을 실행하여 EC2 인스턴스를 시작합니다.

```
$ aws ec2 run-instances \
--image-id AMI ID \
--count 1 \
--instance-type t2.micro \
--associate-public-ip-address \
--key-name key-pair-name \
--security-group-ids ID of the security group created for EC2 instance \
--subnet-id VPC subnet ID \
--region us-west-2 \
--profile adminuser
```

3. `run-instances` 명령이 반환한 인스턴스 ID를 기록합니다.
4. 생성한 EC2 인스턴스에 EC2 인스턴스 연결과 파일 시스템 탑재에 사용할 퍼블릭 DNS 이름이 있어야 합니다. 퍼블릭 DNS 이름은 다음 형식을 갖고 있습니다.

```
ec2-xx-xx-xx-xxx.compute-1.amazonaws.com
```

다음 CLI 명령을 실행하고 퍼블릭 DNS 이름을 기록합니다.

```
aws ec2 describe-instances \
--instance-ids EC2 instance ID \
--region us-west-2 \
--profile adminuser
```

퍼블릭 DNS 이름을 찾지 못하는 경우 EC2 인스턴스를 시작한 VPC 구성을 확인합니다. 자세한 내용은 사전 조건 단원을 참조하십시오.

5. (선택 사항)생성한 EC2 인스턴스에 이름을 할당합니다. 이렇게 하려면 키 이름과 인스턴스에 할당하려는 이름으로 설정된 값으로 태그를 추가합니다. 다음 AWS CLI `create-tags` 명령을 실행하여이 작업을 수행합니다.

```
$  aws ec2 create-tags \
--resources  EC2-instance-ID \
```

```
--tags Key=Name,Value=Provide-instance-name  \
--region us-west-2 \
--profile adminuser
```

2단계: EFS 리소스 생성

이 단계에서는 다음 작업을 수행합니다.

- 암호화된 EFS 파일 시스템 생성
- 수명 주기 관리를 활성화
- EFS 인스턴스를 시작한 가용 영역에 탑재 대상 생성

2.1단계: EFS 파일 시스템 생성

이 단계에서는 EFS 파일 시스템을 생성합니다. `FileSystemId`를 적어 두고 다음 단계에서 파일 시스템의 탑재 대상을 생성할 때 사용합니다.

파일 시스템을 만들려면

- 파일 시스템을 생성하고, 필요한 경우 `Name` 태그를 추가합니다.

  a. 명령 프롬프트에서 다음 AWS CLI `create-file-system` 명령을 실행합니다.

  ```
  $  aws efs create-file-system \
  --encrypted \
  --creation-token FileSystemForWalkthrough1 \
  --tags Key=Name,Value=SomeExampleNameValue \
  --region us-west-2 \
  --profile adminuser
  ```

  응답은 다음과 같습니다.

  ```
  {
      "OwnerId": "111122223333",
      "CreationToken": "FileSystemForWalkthrough1",
      "FileSystemId": "fs-c657c8bf",
      "CreationTime": 1548950706.0,
      "LifeCycleState": "creating",
      "NumberOfMountTargets": 0,
  ```

```
    "SizeInBytes": {
        "Value": 0,
        "ValueInIA": 0,
        "ValueInStandard": 0
    },
    "PerformanceMode": "generalPurpose",
    "Encrypted": true,
    "KmsKeyId": "arn:aws:kms:us-west-2:111122223333:a5c11222-7a99-43c8-9dcc-
abcdef123456",
    "ThroughputMode": "bursting",
    "Tags": [
        {
            "Key": "Name",
            "Value": "SomeExampleNameValue"
        }
    ]
}
```

b. `FileSystemId` 값을 기록해 둡니다. 이 값은 2.3 단계: 탑재 대상 만들기 단원에서 이 파일 시스템에 대한 탑재 대상을 생성할 때 필요합니다.

2.2단계: 수명 주기 관리 활성화

이 단계에서는 EFS Infrequent Access(IA) 스토리지 클래스를 사용하기 위해 파일 시스템에서 수명 주기 관리를 활성화합니다. 자세한 내용은 스토리지 수명 주기 관리 및 EFS 스토리지 클래스 섹션을 참조하세요.

수명 주기 관리 활성화

- 명령 프롬프트에서 다음 AWS CLI `put-lifecycle-configuration` 명령을 실행합니다.

```
$  aws efs put-lifecycle-configuration \
--file-system-id fs-c657c8bf \
--lifecycle-policies TransitionToIA=AFTER_30_DAYS \
--region us-west-2 \
--profile adminuser
```

응답은 다음과 같습니다.

```
{
```

```
  "LifecyclePolicies": [
    {
        "TransitionToIA": "AFTER_30_DAYS"
    }
  ]
}
```

2.3 단계: 탑재 대상 만들기

이 단계에서는 EC2 인스턴스에서 시작한 가용 영역에서 파일 시스템의 탑재 대상을 만듭니다.

1. 다음 정보가 있는지 확인합니다.

   - 탑재 대상을 생성하려는 파일 시스템의 ID(예: `fs-example`).
   - 1단계: EC2 리소스 생성단계에서 EC2 인스턴스를 시작한 VPC 서브넷 ID.

     이 자습서에서는 EC2 인스턴스를 시작한 동일한 서브넷에 탑재 대상을 생성하기 때문에 서브넷 ID(예: `subnet-example`)가 필요합니다.
   - 이전 단계에서 탑재 대상에 대해 생성한 보안 그룹의 ID.
2. 명령 프롬프트에서 다음 AWS CLI `create-mount-target` 명령을 실행합니다.

```
$ aws efs create-mount-target \
--file-system-id file-system-id \
--subnet-id  subnet-id \
--security-group ID-of-the security-group-created-for-mount-target \
--region us-west-2 \
--profile adminuser
```

응답은 다음과 같습니다.

```
{
    "MountTargetId": "fsmt-example",
    "NetworkInterfaceId": "eni-example",
    "FileSystemId": "fs-example",
    "PerformanceMode" : "generalPurpose",
    "LifeCycleState": "available",
    "SubnetId": "fs-subnet-example",
    "OwnerId": "account-id",
    "IpAddress": "xxx.xx.xx.xxx"
```

```
}
```

3. 또 describe-mount-targets 명령을 사용해 파일 시스템에 생성한 탑재 대상에 대한 설명을 가져올 수 있습니다.

```
$ aws efs describe-mount-targets \
--file-system-id file-system-id \
--region us-west-2 \
--profile adminuser
```

3단계: EC2 인스턴스에 파일 시스템 탑재 및 테스트

이 단계에서는 다음 작업을 수행합니다.

- 필수 정보 수집
- EC2 인스턴스에 NFS 클라이언트를 설치합니다.
- EC2 인스턴스에 파일 시스템을 탑재하고 테스트합니다.

주제

- 3.1단계: 정보 수집
- 3.2단계: EC2 인스턴스에 NFS 클라이언트 설치
- 3.3단계: EC2 인스턴스에 파일 시스템 탑재 및 테스트

3.1단계: 정보 수집

이 단원의 단계를 수행하면서 다음 정보가 있는지 확인합니다.

- 다음 형식으로 된 EC2 인스턴스의 퍼블릭 DNS 이름:

```
ec2-xx-xxx-xxx-xx.aws-region.compute.amazonaws.com
```

- 파일 시스템의 DNS 이름. 다음 일반 형식을 사용하여 DNS 이름을 구성할 수 있습니다.

```
file-system-id.efs.aws-region.amazonaws.com
```

탑재 대상을 사용하여 파일 시스템을 탑재한 EC2 인스턴스에서는 탑재 대상의 IP 주소에 대해 파일 시스템의 DNS 이름을 확인합니다.

> **Note**
>
> Amazon EFS에서는 EC2 인스턴스에 퍼블릭 IP 주소 또는 퍼블릭 DNS 이름이 없어도 됩니다. 앞에 나열된 사항은 이 연습 예제에서 SSH를 사용하여 VPC 외부에서 인스턴스에 연결할 수 있도록 하기 위한 요구 사항입니다.

3.2단계: EC2 인스턴스에 NFS 클라이언트 설치

Windows에서 또는 Linux, macOS X나 기타 Unix 변형을 실행하는 컴퓨터에서 EC2 인스턴스에 연결할 수 있습니다.

NFS 클라이언트 설치

1. EC2 인스턴스에 연결합니다. 자세한 내용은 Amazon EC2 사용 설명서의 EC2 인스턴스에 연결을 참조하세요.
2. SSH 세션을 사용해 EC2 인스턴스에서 다음 명령을 실행합니다.

   a. (선택 사항)업데이트를 가져오고 재부팅합니다.

   ```
   $  sudo yum -y update
   $  sudo reboot
   ```

   재부팅 후 EC2 인스턴스에 다시 연결합니다.

   b. NFS 클라이언트를 설치합니다.

   ```
   $ sudo yum -y install nfs-utils
   ```

   > **Note**
   >
   > EC2 인스턴스를 시작할 때 Amazon Linux AMI 2016.03.0 Amazon Linux AMI를 선택하면 기본적으로 `nfs-utils`가 AMI에 이미 포함되어 있으므로 설치할 필요가 없습니다.

3.3단계: EC2 인스턴스에 파일 시스템 탑재 및 테스트

이제 EC2 인스턴스에 파일 시스템을 탑재합니다.

1. 디렉터리("efs-mount-point")를 만듭니다.

```
$ mkdir ~/efs-mount-point
```

2. EFS 파일 시스템을 탑재합니다.

```
$ sudo mount -t nfs -o
 nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport mount-
target-DNS:/   ~/efs-mount-point
```

EC2 인스턴스는 IP 주소에 대해 탑재 대상의 DNS 이름을 확인할 수 있습니다. 직접 탑재 대상의 IP 주소를 지정할 수도 있습니다.

```
$ sudo mount -t nfs -o
 nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport mount-
target-ip:/  ~/efs-mount-point
```

3. EC2 인스턴스에 EFS 파일 시스템을 탑재했으므로 이제 파일을 생성할 수 있습니다.

   a. 디렉터리를 변경합니다.

   ```
   $ cd ~/efs-mount-point
   ```

   b. 디렉터리 내용을 나열합니다.

   ```
   $ ls -al
   ```

   비어 있어야 합니다.

   ```
   drwxr-xr-x 2 root     root     4096 Dec 29 22:33 .
   drwx------ 4 ec2-user ec2-user 4096 Dec 29 22:54 ..
   ```

   c. 생성 시 파일 시스템의 루트 디렉터리는 루트 사용자가 소유하며 루트 사용자의 쓰기가 가능합니다. 따라서 파일을 추가할 수 있도록 권한을 변경해야 합니다.

   ```
   $ sudo chmod go+rw .
   ```

   `ls -al` 명령을 사용하면 권한이 변경된 것을 확인할 수 있습니다.

```
drwxrwxrwx 2 root     root     4096 Dec 29 22:33 .
drwx------ 4 ec2-user ec2-user 4096 Dec 29 22:54 ..
```

d. 텍스트 파일을 생성합니다.

```
$ touch test-file.txt
```

e. 디렉터리 내용을 나열합니다.

```
$ ls -l
```

이제 VPC의 EC2 인스턴스에 EFS 파일 시스템을 성공적으로 생성하고 탑재했습니다.

탑재한 파일 시스템은 재부팅하면 탑재가 해제됩니다. 디렉터리를 자동으로 다시 탑재하려면 `fstab` 파일을 사용합니다. Auto Scaling 그룹을 사용하여 EC2 인스턴스를 시작하는 경우 시작 구성에서 스크립트를 설정할 수도 있습니다.

4단계: 정리

만든 리소스가 더 이상 필요 없는 경우 제거해야 합니다. CLI를 사용하여 제거할 수 있습니다.

- EC2 리소스(EC2 인스턴스 및 두 보안 그룹)를 제거합니다. 탑재 대상을 삭제하면 Amazon EFS는 네트워크 인터페이스를 삭제합니다.
- EFS 리소스(파일 시스템, 탑재 대상)를 제거합니다.

이 연습에서 생성된 AWS 리소스를 삭제하려면

1. 이 자습서를 위해 생성한 EC2 인스턴스를 종료합니다.

   ```
   $ aws ec2 terminate-instances \
   --instance-ids instance-id \
   --profile adminuser
   ```

   콘솔을 사용해 EC2 리소스 또한 삭제할 수 있습니다. 지침은 인스턴스 종료 섹션을 참조하세요.
2. 탑재 대상을 삭제합니다.

   파일 시스템을 삭제하기 전에 파일 시스템에 생성한 탑재 대상을 삭제해야 합니다. `describe-mount-targets` CLI 명령을 사용해 탑재 대상 목록을 가져올 수 있습니다.

```
$  aws efs describe-mount-targets \
--file-system-id file-system-ID \
--profile adminuser \
--region aws-region
```

그런 다음 `delete-mount-target` CLI 명령을 사용해 탑재 대상을 삭제합니다.

```
$ aws efs delete-mount-target \
--mount-target-id ID-of-mount-target-to-delete \
--profile adminuser \
--region aws-region
```

3. (선택 사항)생성한 두 개 보안 그룹을 삭제합니다. 보안 그룹 생성은 무료입니다.

   먼저 탑재 대상의 보안 그룹을 생성한 후 EC2 인스턴스의 보안 그룹을 삭제해야 합니다. 탑재 대상의 보안 그룹에 EC2 보안 그룹을 참조하는 규칙이 있기 때문입니다. 따라서 EC2 인스턴스의 보안 그룹을 먼저 삭제할 수 없습니다.

   자세한 내용은 Amazon EC2 사용 설명서의 보안 그룹 삭제를 참조하세요.

4. `delete-file-system` CLI 명령을 사용해 파일 시스템을 삭제합니다. `describe-file-systems` CLI 명령을 사용해 파일 시스템 목록을 가져올 수 있습니다. 이 응답에서 파일 시스템 ID를 얻을 수 있습니다.

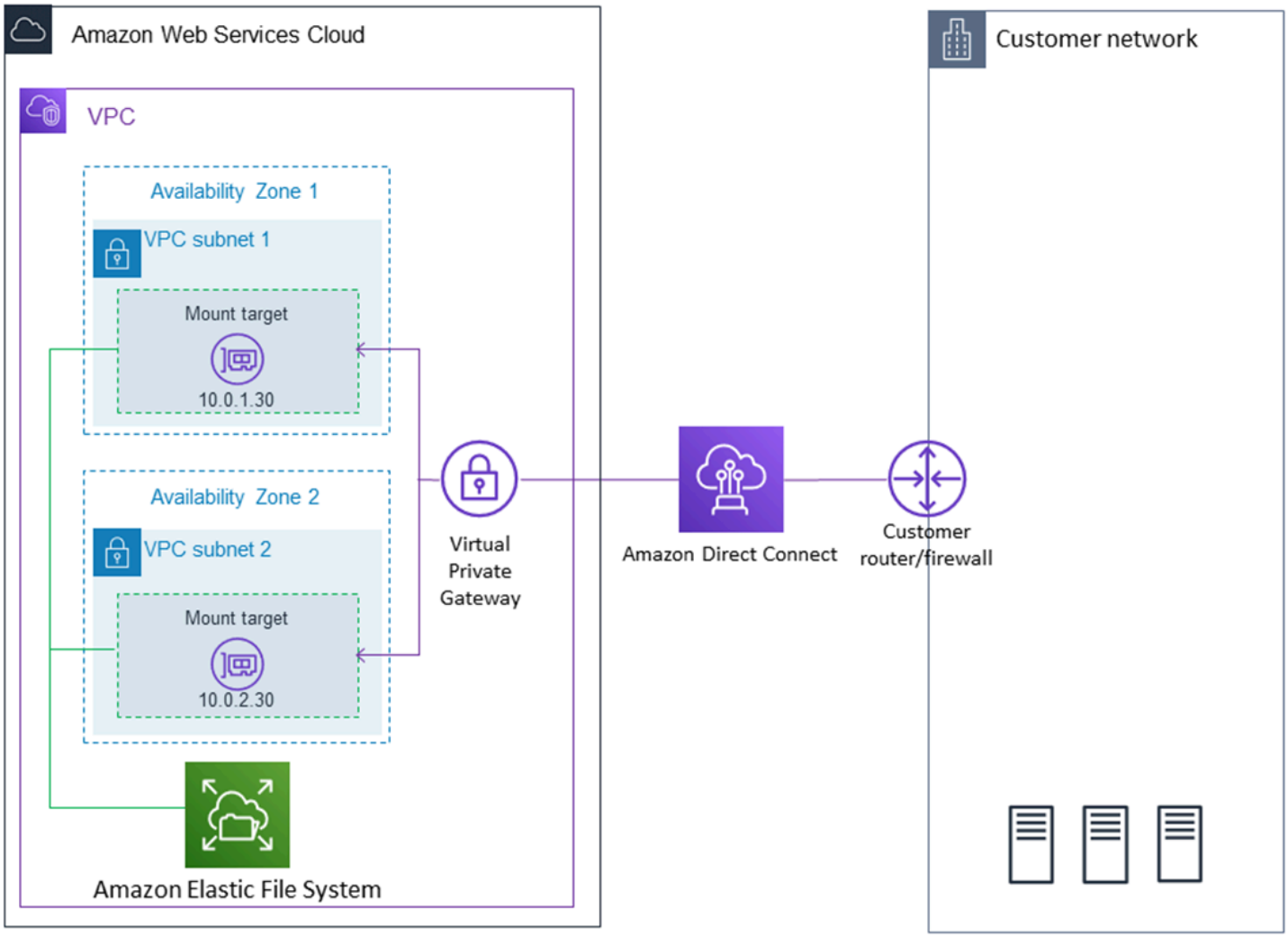
   ```
   aws efs describe-file-systems \
   --profile adminuser \
   --region aws-region
   ```

   파일 시스템 ID를 제공해 파일 시스템을 삭제합니다.

   ```
   $ aws efs delete-file-system \
   --file-system-id ID-of-file-system-to-delete \
   --region aws-region \
   --profile adminuser
   ```

자습서: 온프레미스 Linux 클라이언트로 탑재

Direct Connect 또는 VPN을 사용하여 Amazon VPC에 연결할 때 온프레미스 데이터 센터 서버에 EFS 파일 시스템을 탑재할 수 있습니다. 다음 그림은 온프레미스에서 EFS 파일 시스템을 탑재하는 데 AWS 서비스 필요한의 개략적인 개략도를 보여줍니다.



> **Note**
>
> Microsoft Windows 기반 클라이언트에서 Amazon EFS를 사용하는 것은 지원되지 않습니다.

주제

- 사전 조건
- 1단계: EFS 리소스 생성

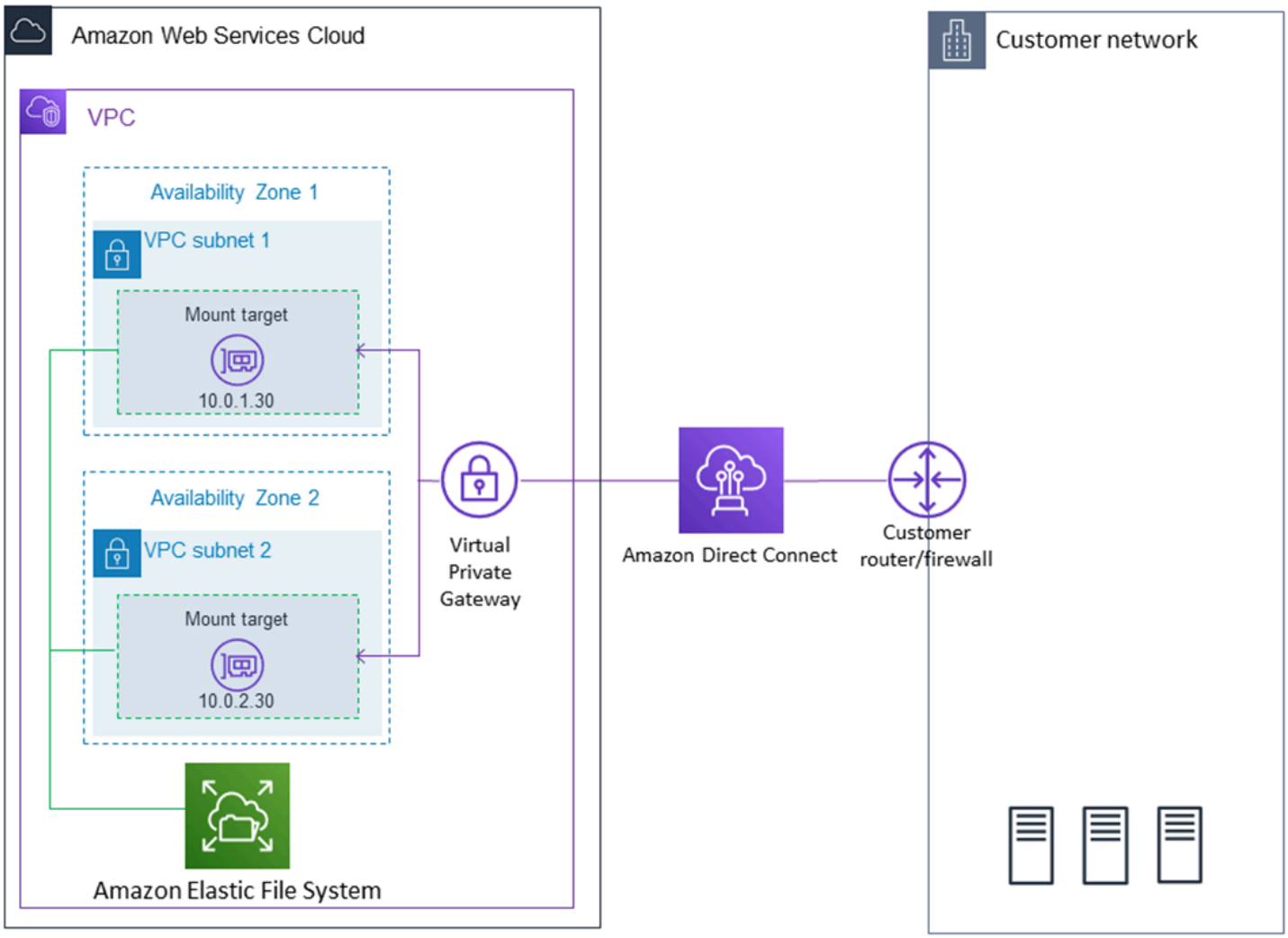
- 2단계: NFS 클라이언트 설치
- 3단계: 온프레미스 클라이언트에 EFS 파일 시스템 탑재
- 4단계: 리소스 정리 및 AWS 계정 보호
- 선택 사항: 전송 중 데이터 암호화

사전 조건

Direct Connect 또는 VPN 연결이 이미 있는지 확인합니다. Direct Connect에 대한 자세한 내용은 Direct Connect 사용 설명서를 참조하세요. VPN 연결 설정에 대한 자세한 내용은 Amazon VPC 사용 설명서의 AWS Virtual Private Network를 사용하여 VPC를 원격 네트워크에 연결을 참조하세요.

Direct Connect 또는 VPN 연결이 완료되면 Amazon VPC에서 EFS 파일 시스템과 탑재 대상을 생성합니다. 이후 amazon-efs utils 도구를 다운로드 및 설치합니다. 그런 다음 온프레미스 클라이언트에서 파일 시스템을 테스트합니다. 이 연습 마지막의 정리 단계에서는 이러한 리소스를 제거하는 방법을 설명합니다.

이 연습에서는 미국 서부(오레곤) 리전(`us-west-2`)에 이러한 리소스를 모두 만듭니다. 어떤 방법을 사용 AWS 리전 하든 일관되게 사용해야 합니다. 다음 다이어그램과 AWS 리전같이 VPC, 탑재 대상, EFS 파일 시스템 등 모든 리소스가 동일해야 합니다.



> **Note**
>
> 경우에 따라 로컬 애플리케이션이 EFS 파일 시스템을 사용할 수 있는지 알아야 할 수도 있습니다. 이러한 경우 첫 번째 탑재 지점을 일시적으로 사용할 수 없게 되더라도 애플리케이션이 다른 탑재 지점 IP 주소를 가리킬 수 있어야 합니다. 이 시나리오에서는 가용성을 높이기 위해 두 개의 온프레미스 클라이언트를 서로 다른 가용 영역을 통해 파일 시스템에 연결하는 것이 좋습니다.

의 루트 자격 증명을 사용하여 콘솔에 로그인하고이 연습을 시도 AWS 계정 할 수 있습니다. 그러나 AWS Identity and Access Management (IAM) 모범 사례에서는의 루트 자격 증명을 사용하지 않는 것이 좋습니다 AWS 계정. 대신 계정에서 관리자 사용자를 만들어 해당 보안 인증을 사용하여 계정에서 리소스를 관리합니다. 자세한 내용은 AWS IAM Identity Center 사용 설명서의 AWS 계정에 대한 Single Sign-on 액세스를 참조하세요.

자신의 계정에서 만든 기본 VPC 또는 사용자 지정 VPC를 사용할 수 있습니다. 이 연습에서는 기본 VPC 구성을 사용합니다. 단, 사용자 지정 VPC를 사용하는 경우 다음을 확인하세요.

- 인터넷 게이트웨이가 VPC에 연결되어 있습니다. 자세한 내용은 Amazon VPC 사용 설명서의 인터넷 게이트웨이를 사용하여 VPC에 대한 인터넷 액세스 활성화를 참조하세요.
- VPC 라우팅 테이블에 모든 인터넷 바인딩된 트래픽을 인터넷 게이트웨이로 보내는 규칙이 포함되어 있습니다.

1단계: EFS 리소스 생성

이 단계에서는 EFS 파일 시스템 및 탑재 대상을 생성합니다.

EFS 파일 시스템을 생성하려면

1. Amazon Elastic File System 콘솔(https://console.aws.amazon.com/efs/)을 엽니다.
2. 파일 시스템 생성을 선택합니다.
3. VPC 목록에서 기본 VPC를 선택합니다.
4. 모든 가용 영역이 확인란을 선택합니다. 모두 기본 서브넷, 자동 IP 주소, 기본 보안 그룹이 선택되어 있어야 합니다. 이것이 탑재 대상입니다. 자세한 내용은 탑재 대상 생성 단원을 참조하십시오.
5. 다음 단계를 선택합니다.
6. 파일 시스템의 이름을 지정하고 범용을 기본 성능 모드로 선택한 후 다음 단계를 선택합니다.
7. 파일 시스템 생성을 선택합니다.
8. 목록에서 파일 시스템을 선택하고 보안 그룹 값을 기록해 둡니다. 다음 단계에서 이 값을 사용합니다.

방금 생성한 파일 시스템에는 탑재 대상이 있습니다. 각 탑재 대상에는 연결된 보안 그룹이 있습니다. 보안 그룹은 네트워크 트래픽을 제어하는 가상 방화벽의 역할을 수행합니다. 탑재 대상을 만들 때 보안 그룹을 제공하지 않으면 Amazon EFS에서는 VPC의 기본 보안 그룹을 탑재 대상과 연결합니다. 이전 단계를 정확히 따랐다면 탑재 대상은 기본 보안 그룹을 사용하고 있는 것입니다.

다음으로 탑재 대상의 보안 그룹에 네트워크 파일 시스템(NFS) 포트(2049)로의 인바운드 트래픽을 허용하는 규칙을 추가합니다. AWS Management Console 를 사용하여 VPC에서 탑재 대상의 보안 그룹에 규칙을 추가할 수 있습니다.

NFS 포트로의 인바운드 트래픽을 허용하려면

1. 에 로그인 AWS Management Console 하고 https://console.aws.amazon.com/ec2/ Amazon EC2 콘솔을 엽니다.
2. 네트워크 및 보안에서 보안 그룹을 선택합니다.
3. 파일 시스템과 연결된 보안 그룹을 선택합니다. 1단계: EFS 리소스 생성의 마지막 부분에서 이 값을 기록해 두었습니다.
4. 보안 그룹 목록 아래에 나타나는 탭 창에서 인바운드 탭을 선택합니다.
5. 편집을 선택합니다.
6. 규칙 추가를 선택하고 다음 유형의 규칙을 선택합니다.

   - 유형 - NFS
   - 출처 – 모든 소스

   모든 소스는 테스트용으로만 사용하는 것이 좋습니다. 온프레미스 클라이언트의 IP 주소로 사용자 지정 소스 세트를 만들거나 클라이언트 자체의 콘솔을 사용하고 내 IP를 선택할 수 있습니다.

   > **Note**
   >
   > 기본 아웃바운드 규칙이 모든 트래픽이 나가도록 허용하고 있기 때문에 아웃바운드 규칙을 추가할 필요는 없습니다. 이 기본 아웃바운드 규칙이 없는 경우, NFS 포트 2049에서 TCP 연결을 열어 탑재 대상 보안 그룹을 대상으로 식별하는 아웃바운드 규칙을 추가해야 합니다.

2단계: NFS 클라이언트 설치

이 단계에서는 NFS 클라이언트를 설치합니다.

온프레미스 서버에 NFS 클라이언트를 설치하려면

> **Note**
>
> 전송 중에 데이터를 암호화해야 하는 경우 NFS 클라이언트 대신 Amazon EFS 탑재 도우미 `amazon-efs-utils`를 사용하세요. amazon-efs-utils 설치에 대한 자세한 내용은 선택 사항: 전송 중 데이터 암호화 섹션을 참조하세요.

1. 온프레미스 클라이언트용 터미널에 액세스하세요.
2. NFS를 설치합니다.

   Red Hat Linux를 사용하는 경우 다음 명령으로 NFS를 설치합니다.

   ```
   $ sudo yum -y install nfs-utils
   ```

   Ubuntu를 사용하는 경우 다음 명령으로 NFS를 설치합니다.

   ```
   $ sudo apt-get -y install nfs-common
   ```

3단계: 온프레미스 클라이언트에 EFS 파일 시스템 탑재

탑재 디렉터리를 생성하려면

1. 다음 명령으로 마운트 지점에 대한 디렉터리를 만듭니다.

   Example

   ```
   mkdir ~/efs
   ```

2. 가용 영역에서 탑재 대상의 선호하는 IP 주소를 선택합니다. 온프레미스 Linux 클라이언트에서 지연 시간을 측정할 수 있습니다. 그러려면 여러 가용 영역에 있는 EC2 인스턴스의 IP 주소를 기준으로 ping 같은 터미널 기반 도구를 사용하여 지연 시간이 가장 낮은 인스턴스를 찾으세요.

- 탑재 명령을 실행하여 탑재 대상의 IP 주소를 사용하여 파일 시스템을 탑재합니다.

  ```
  $ sudo mount -t nfs -o
   nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport mount-target-IP:/   ~/efs
  ```

EFS 파일 시스템을 탑재했으므로 다음 절차에 따라 테스트할 수 있습니다.

EFS 파일 시스템 연결을 테스트하려면

1. 만든 디렉터리를 다음 명령을 사용하여 새 디렉터리로 변경합니다.

```
$ cd ~/efs
```

2. 하위 디렉터리를 만들고 해당 하위 디렉터리의 소유권을 EC2 인스턴스 사용자로 변경합니다. 그런 다음 아래 명령을 사용하여 새 디렉터리로 이동합니다.

```
$ sudo mkdir getting-started
$ sudo chown ec2-user getting-started
$ cd getting-started
```

3. 다음 명령을 사용하여 텍스트 파일을 만듭니다.

```
$ touch test-file.txt
```

4. 다음 명령을 사용하여 디렉터리 콘텐츠를 나열합니다.

```
$ ls -al
```

따라서 다음 파일이 생성됩니다.

```
-rw-rw-r-- 1 username username 0 Nov 15 15:32 test-file.txt
```

> **Warning**
>
> 파일 시스템을 자동으로 마운트하는 경우 네트워크 파일 시스템 식별에 사용하는 _netdev 옵션을 사용합니다. _netdev이 빠진 경우 EC2 인스턴스가 응답을 중지합니다. 컴퓨팅 인스턴스가 네트워킹을 시작한 후 네트워크 파일 시스템의 초기화를 완료해야 하기 때문입니다. 자세한 내용은 자동 탑재 실패 및 인스턴스 무응답 단원을 참조하십시오.

4단계: 리소스 정리 및 AWS 계정 보호

연습을 마친 뒤에, 또는 연습을 건너뛰고 다음 단계에 따라 리소스를 정리해 AWS 계정을 보호해야 합니다.

리소스를 정리하고를 보호하려면 AWS 계정

1. 다음 명령을 사용하여 EFS 파일 시스템 탑재를 해제합니다.

```
$ sudo umount ~/efs
```

2. Amazon EFS 콘솔에서 Amazon EKS 콘솔을 엽니다.
3. 파일 시스템 목록에서 삭제하려는 EFS 파일 시스템을 선택합니다.
4. 작업에서 파일 시스템 삭제를 선택합니다.
5. 영구적으로 파일 시스템 삭제 대화 상자에서 삭제하려는 EFS 파일 시스템에 대한 파일 시스템 ID를 입력한 다음 파일 시스템 삭제를 선택합니다.
6. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
7. 탐색 창에서 보안 그룹을 선택합니다.
8. 이 연습에서 규칙을 추가했던 보안 그룹의 이름을 선택합니다.

> **Warning**
>
> VPC에 대한 기본 보안 그룹은 삭제할 수 없습니다.

9. 작업에서 인바운드 규칙을 선택합니다.
10. 추가한 인바운드 규칙 마지막 부분의 X를 선택한 다음 저장을 선택합니다.

선택 사항: 전송 중 데이터 암호화

전송 중 데이터를 암호화하려면 NFS 클라이언트 대신 Amazon EFS 탑재 도우미인 amazon-efs-utils를 사용하세요.

amazon-efs-utils 패키지는 오픈 소스 Amazon EFS 도구 모음입니다. amazon-efs-utils 컬렉션에는 Amazon EFS에서 전송 중 데이터를 쉽게 암호화할 수 있는 도구와 탑재 도우미가 함께 제공됩니다. 이 패키지에 대한 자세한 내용은 Amazon EFS 클라이언트 수동 설치 섹션을 참조하세요. 이 패키지는 GitHub에서 무료로 다운로드할 수 있으며, 패키지 저장소를 복제하여 다운로드할 수 있습니다.

GitHub에서 amazon-efs-utils 복제

1. 온프레미스 클라이언트용 터미널에 액세스하세요.
2. 터미널에서 GitHub로부터 선택한 디렉터리로 다음 명령을 사용해 amazon-efs-utils 도구를 복제합니다.

```
git clone https://github.com/aws/efs-utils
```

이제 패키지가 준비되었으니 설치할 수 있습니다. 이 설치는 온프레미스 클라이언트의 Linux 배포판에 따라 다르게 처리됩니다. 다음 배포판이 지원됩니다.

- Amazon Linux 2
- Amazon Linux
- Red Hat Enterprise Linux(및 CentOS와 같은 파생 버전) 버전 8 이상
- Ubuntu 16.04 LTS 이상

amazon-efs-utils를 RPM 패키지로 빌드 및 설치

1. 클라이언트 터미널을 열어 GitHub에서 복제한 amazon-efs-utils 패키지가 위치한 디렉터리를 찾습니다.
2. 다음 명령을 사용하여 패키지를 빌드합니다.

```
make rpm
```

> **Note**
>
> 빌드하지 않은 경우 다음 명령을 사용해 rpm-builder 패키지를 설치합니다.
>
> ```
> sudo yum -y install rpm-build
> ```

3. 다음 명령을 사용하여 패키지를 설치합니다.

```
sudo yum -y install build/amazon-efs-utils*rpm
```

amazon-efs-utils를 DEB 패키지로 빌드 및 설치

1. 클라이언트 터미널을 열어 GitHub에서 복제한 amazon-efs-utils 패키지가 위치한 디렉터리를 찾습니다.
2. 다음 명령을 사용하여 패키지를 빌드합니다.

```
./build-deb.sh
```

3. 다음 명령을 사용하여 패키지를 설치합니다.

```
sudo apt-get install build/amazon-efs-utils*deb
```

패키지를 설치한 후 Direct Connect 또는 VPN과 AWS 리전 함께에서 사용할 수 있도록 amazon-efs-utils 구성합니다.

에서 사용할 amazon-efs-utils를 구성하려면 AWS 리전

1. Secure Shell(SSH)를 통해 EC2 인스턴스의 터미널에 액세스하고, 적절한 사용자 이름으로 로그인합니다. 자세한 내용은 Amazon EC2 사용 설명서의 EC2 인스턴스에 연결을 참조하세요.
2. 선택한 텍스트 편집기를 사용해 `/etc/amazon/efs/efs-utils.conf` 파일을 엽니다.
3. "`dns_name_format = {fs_id}.efs.{region}.amazonaws.com`" 라인을 찾습니다.
4. *`{region}`*을 해당 AWS 리전의 ID로 변경하세요(예: `us-west-2`).

온프레미스 클라이언트에 EFS 파일 시스템을 탑재하려면 먼저 온프레미스 Linux 클라이언트에서 터미널을 여세요. 시스템을 탑재하려면 파일 시스템 ID, 탑재 대상 중 하나의 탑재 대상 IP 주소 및 파일 시스템의 AWS 리전이 필요합니다. 파일 시스템에 탑재 대상을 여러 개 만든 경우 이 중 하나를 선택할 수 있습니다.

해당 정보가 있으면 다음 세 단계로 파일 시스템을 탑재할 수 있습니다.

탑재 디렉터리를 생성하려면

1. 다음 명령으로 마운트 지점에 대한 디렉터리를 만듭니다.

   Example

   ```
   mkdir ~/efs
   ```

2. 가용 영역에서 탑재 대상의 선호하는 IP 주소를 선택합니다. 온프레미스 Linux 클라이언트에서 지연 시간을 측정할 수 있습니다. 그러려면 여러 가용 영역에 있는 EC2 인스턴스의 IP 주소를 기준으로 `ping` 같은 터미널 기반 도구를 사용하여 지연 시간이 가장 낮은 인스턴스를 찾으세요.

`/etc/hosts`를 업데이트하려면

- 파일 시스템 ID 및 탑재 대상 IP 주소를 포함하는 로컬 `/etc/hosts` 파일에 다음 형식으로 항목을 추가합니다.

```
mount-target-IP-Address file-system-ID.efs.region.amazonaws.com
```

Example

```
192.0.2.0 fs-12345678.efs.us-west-2.amazonaws.com
```

탑재 디렉터리를 생성하려면

1. 다음 명령으로 마운트 지점에 대한 디렉터리를 만듭니다.

   Example

   ```
   mkdir ~/efs
   ```

2. 그런 다음 탑재 명령을 실행하여 파일 시스템을 탑재합니다.

   Example

   ```
   sudo mount -t efs fs-12345678 ~/efs
   ```

   전송 중 데이터의 암호화를 사용하려는 경우 탑재 명령은 다음과 같습니다.

   Example

   ```
   sudo mount -t efs -o tls fs-12345678 ~/efs
   ```

자습서: 다른 VPC에서 파일 시스템 탑재하기

이 자습서에서는 다른 가상 프라이빗 클라우드(VPC)에 있는 EFS 파일 시스템을 탑재하도록 EC2 인스턴스를 설정합니다. 이 작업은 EFS 탑재 도우미를 사용하여 수행할 수 있습니다. 탑재 도우미는 `amazon-efs-utils` 도구 세트의 일부입니다. `amazon-efs-utils`에 대한 자세한 정보는 Amazon EFS 클라이언트 수동 설치 섹션을 참조하세요.

클라이언트의 VPC와 EFS 파일 시스템의 VPC는 VPC 피어링 연결 또는 VPC 전송 게이트웨이를 사용하여 연결되어야 합니다. VPC 피어링 연결 또는 전송 게이트웨이를 사용하여 VPC를 연결하면 VPC가

다른 계정에 속해 있더라도 하나의 VPC에 있는 EC2 인스턴스가 다른 VPC의 EFS 파일 시스템에 액세스할 수 있습니다.

> **Note**
>
> Microsoft Windows 기반 클라이언트에서 Amazon EFS를 사용하는 것은 지원되지 않습니다.

주제

- 사전 조건
- 1단계: EFS 탑재 대상의 가용 영역 ID 확인
- 2단계: 탑재 대상 IP 주소 확인
- 3단계: 탑재 대상에 대한 호스트 항목 추가
- 4단계: EFS 탑재 도우미를 사용하여 파일 시스템 탑재
- 5단계: 리소스 정리 및 보호 AWS 계정

사전 조건

이 자습서를 완료하려면 다음과 같은 사전 조건이 필요합니다.

- `amazon-efs-utils` 도구 세트는 이 절차를 사용하기 전에 EC2 인스턴스에 설치됩니다. `amazon-efs-utils`를 설치하는 방법에 대한 지침은 Amazon EFS 클라이언트 수동 설치 단원을 참조하세요.
- 다음 중 하나입니다.
  - EFS 파일 시스템이 있는 VPC와 EC2 인스턴스가 있는 VPC 간의 VPC 피어링 연결. VPC 피어링 연결은 두 VPC 간의 네트워킹 연결입니다. 이러한 유형의 연결을 사용하면 프라이빗 Internet Protocol version 4(IPv4) 또는 Internet Protocol version 6(IPv6) 주소를 사용하여 이들 간의 트래픽을 라우팅할 수 있습니다. VPC 피어링을 사용하여 동일하거나 AWS 리전 그 사이에 VPCs를 연결할 수 있습니다 AWS 리전. 자세한 정보는 Amazon VPC 피어링 가이드의 Amazon VPC 피어링 연결 생성 및 수락을 참조하세요.
  - EFS 파일 시스템이 있는 VPC와 EC2 인스턴스가 있는 VPC를 연결하는 전송 게이트웨이. 전송 게이트웨이는 VPC와 온프레미스 네트워크를 상호 연결하는 데 사용할 수 있는 네트워크 전송 허브입니다. 자세한 내용은 Amazon VPC Transit Gateways 설명서의 Amazon VPC Transit Gateways를 사용하여 시작하기를 참조하세요.

1단계: EFS 탑재 대상의 가용 영역 ID 확인

파일 시스템의 고가용성을 보장하려면 항상 NFS 클라이언트와 동일한 가용 영역에 있는 EC2 탑재 대상 IP 주소를 사용하는 것이 좋습니다. 다른 계정에 있는 EFS 파일 시스템을 탑재하는 경우, NFS 클라이언트와 EFS 탑재 대상이 동일한 가용 영역 ID에 있는지 확인하세요. 가용 영역 이름은 계정마다 다를 수 있으므로 이 요구 사항이 적용됩니다.

EC2 인스턴스의 가용 영역 ID를 확인하려면

1. EC2 인스턴스에 연결합니다. 자세한 내용은 Amazon EC2 사용 설명서의 EC2 인스턴스에 연결을 참조하세요.
2. 다음과 같이 `describe-availability-zones` CLI 명령을 사용하여 EC2 인스턴스가 있는 가용 영역 ID를 확인합니다.

```
[ec2-user@ip-10.0.0.1] $ aws ec2 describe-availability-zones --zone-name
{
    "AvailabilityZones": [
        {
            "State": "available",
            "ZoneName": "us-east-2b",
            "Messages": [],
            "ZoneId": "use2-az2",
            "RegionName": "us-east-2"
        }
    ]
}
```

가용 영역 ID는 `ZoneId` 속성인 `use2-az2`에 반환됩니다.

2단계: 탑재 대상 IP 주소 확인

이제 EC2 인스턴스의 가용 영역 ID를 알았으므로 동일한 가용 영역 ID에 있는 탑재 대상의 IP 주소를 검색할 수 있습니다.

동일한 가용 영역 ID에서 탑재 대상 IP 주소 확인

- 다음과 같이 `describe-mount-targets` CLI 명령을 사용하여 `use2-az2` AZ ID에서 파일 시스템의 탑재 대상 IP 주소를 검색합니다.

```
$ aws efs describe-mount-targets --file-system-id file_system_id
```

```
{
    "MountTargets": [
        {
            "OwnerId": "111122223333",
            "MountTargetId": "fsmt-11223344",
  =====>     "AvailabilityZoneId": "use2-az2",
            "NetworkInterfaceId": "eni-048c09a306023eeec",
            "AvailabilityZoneName": "us-east-2b",
            "FileSystemId": "fs-01234567",
            "LifeCycleState": "available",
            "SubnetId": "subnet-06eb0da37ee82a64f",
            "OwnerId": "958322738406",
  =====>     "IpAddress": "10.0.2.153"
        },
...
        {
            "OwnerId": "111122223333",
            "MountTargetId": "fsmt-667788aa",
            "AvailabilityZoneId": "use2-az3",
            "NetworkInterfaceId": "eni-0edb579d21ed39261",
            "AvailabilityZoneName": "us-east-2c",
            "FileSystemId": "fs-01234567",
            "LifeCycleState": "available",
            "SubnetId": "subnet-0ee85556822c441af",
            "OwnerId": "958322738406",
            "IpAddress": "10.0.3.107"
        }
    ]
}
```

use2-az2 가용 영역 ID에 있는 탑재 대상의 IP 주소는 10.0.2.153입니다.

3단계: 탑재 대상에 대한 호스트 항목 추가

이제 탑재 대상 IP 주소를 EFS 파일 시스템의 호스트 이름에 매핑하는 EC2 인스턴스의 /etc/hosts 파일에 입력할 수 있습니다.

탑재 대상에 대한 호스트 항목 추가

1. EC2 인스턴스의 /etc/hosts 파일에 탑재 대상 IP 주소에 대한 줄을 추가합니다. 사용되는 항목 형식은 m*ount-target-IP-Address file-system-ID*.efs.*region*.amazonaws.com입니다. 다음 명령을 사용하여 파일에 줄을 추가합니다.

```
echo "10.0.2.153 fs-01234567.efs.us-east-2.amazonaws.com" | sudo tee -a /etc/hosts
```

2. EC2 인스턴스 및 탑재 대상의 VPC 보안 그룹에 필요에 따라 EFS 파일 시스템에 대한 액세스를 허용하는 규칙이 있는지 확인합니다. 자세한 내용은 VPC 보안 그룹 사용 단원을 참조하십시오.

4단계: EFS 탑재 도우미를 사용하여 파일 시스템 탑재

EFS 파일 시스템을 탑재하려면 먼저 EC2 인스턴스에 탑재 디렉터리를 생성합니다. 그런 다음 EFS 탑재 도우미를 사용하여 AWS Identity and Access Management (IAM) 권한 부여 또는 EFS 액세스 포인트로 파일 시스템을 탑재할 수 있습니다. 자세한 내용은 IAM을 사용하여 파일 시스템에 대한 액세스 제어 및 액세스 포인트 작업 섹션을 참조하세요.

탑재 디렉터리를 생성하려면

- 다음 명령을 사용하여 파일 시스템을 탑재할 디렉터리를 생성합니다.

```
$ sudo mkdir /mnt/efs/
```

IAM 권한 부여를 사용하여 파일 시스템 탑재

- 다음 명령을 사용하여 IAM 권한 부여를 통해 파일 시스템을 탑재합니다.

```
$ sudo mount -t efs -o tls,iam file-system-id /mnt/efs/
```

EFS 액세스 포인트를 사용하여 파일 시스템 탑재

- EFS 액세스 포인트를 사용하여 파일 시스템을 탑재하려면 다음 명령을 사용합니다.

```
$ sudo mount -t efs -o tls,accesspoint=access-point-id file-system-id /mnt/efs/
```

EFS 파일 시스템을 탑재했으므로 다음 절차에 따라 테스트할 수 있습니다.

EFS 파일 시스템 연결을 테스트하려면

1. 만든 디렉터리를 다음 명령을 사용하여 새 디렉터리로 변경합니다.

```
$ cd ~/mnt/efs
```

2. 하위 디렉터리를 만들고 해당 하위 디렉터리의 소유권을 EC2 인스턴스 사용자로 변경합니다. 그런 다음 아래 명령을 사용하여 새 디렉터리로 이동합니다.

```
$ sudo mkdir getting-started
$ sudo chown ec2-user getting-started
$ cd getting-started
```

3. 다음 명령을 사용하여 텍스트 파일을 만듭니다.

```
$ touch test-file.txt
```

4. 다음 명령을 사용하여 디렉터리 콘텐츠를 나열합니다.

```
$ ls -al
```

따라서 다음 파일이 생성됩니다.

```
-rw-rw-r-- 1 username username 0 Nov 15 15:32 test-file.txt
```

/etc/fstab 파일에 항목을 추가하는 방법으로 자동으로 파일 시스템을 탑재할 수도 있습니다. 자세한 내용은 기존 EC2 Linux 인스턴스에서 자동 탑재 활성화 단원을 참조하십시오.

> **Warning**
>
> 파일 시스템을 자동으로 마운트하는 경우 네트워크 파일 시스템 식별에 사용하는 _netdev 옵션을 사용합니다. _netdev이 빠진 경우 EC2 인스턴스가 응답을 중지합니다. 컴퓨팅 인스턴스가 네트워킹을 시작한 후 네트워크 파일 시스템의 초기화를 완료해야 하기 때문입니다. 자세한 내용은 자동 탑재 실패 및 인스턴스 무응답 단원을 참조하십시오.

5단계: 리소스 정리 및 보호 AWS 계정

이 자습서를 완료한 후에는 다음 단계를 수행하여 리소스를 정리하고 AWS 계정을 보호합니다.

리소스를 정리하고를 보호하려면 AWS 계정

1. 다음 명령을 사용하여 EFS 파일 시스템 탑재를 해제합니다.

```
$ sudo umount ~/efs
```

2. Amazon Elastic File System 콘솔(https://console.aws.amazon.com/efs/)을 엽니다.
3. 파일 시스템 목록에서 삭제하려는 EFS 파일 시스템을 선택합니다.
4. 작업에서 파일 시스템 삭제를 선택합니다.
5. 영구적으로 파일 시스템 삭제 대화 상자에서 삭제하려는 EFS 파일 시스템에 대한 파일 시스템 ID를 입력한 다음 파일 시스템 삭제를 선택합니다.
6. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
7. 탐색 창에서 보안 그룹을 선택합니다.
8. 이 자습서에 사용할 규칙을 추가한 보안 그룹의 이름을 선택합니다.

> **Warning**
>
> VPC에 대한 기본 보안 그룹은 삭제할 수 없습니다.

9. 작업에서 인바운드 규칙을 선택합니다.
10. 추가한 인바운드 규칙 마지막 부분의 X를 선택한 다음 저장을 선택합니다.

탑재 문제 해결

아래에서 EFS 파일 시스템 탑재 문제 해결에 대한 정보를 확인할 수 있습니다.

Windows 인스턴스에서 파일 시스템 탑재에 실패

Microsoft Windows의 Amazon EC2 인스턴스에서 파일 시스템 탑재에 실패.

취할 조치

Windows EC2 인스턴스에 Amazon EFS를 사용하지 않습니다. 지원하지 않기 때문입니다.

서버에 의한 액세스 거부

파일 시스템 탑재가 실패하고 다음 메시지가 표시됩니다.

```
/efs mount.nfs4: access denied by server while mounting 127.0.0.1:/
```

NFS 클라이언트에 파일 시스템을 탑재할 권한이 없는 경우 이 문제가 발생할 수 있습니다.

취할 조치

IAM을 사용하여 파일 시스템을 탑재하려는 경우 탑재 명령에서 `-o iam` 또는 -o tls옵션을 사용해야 합니다. 그러면 보안 인증을 EFS 탑재 대상에 전달하도록 EFS 탑재 도우미에 지시합니다. 여전히 액세스 권한이 없는 경우 파일 시스템 정책과 자격 증명 정책을 확인하여 연결에 적용되는 DENY 절이 없는지, 연결에 적용되는 ALLOW 절이 하나 이상 있는지 확인합니다. 자세한 내용은 IAM을 사용하여 파일 시스템에 대한 액세스 제어 및 파일 시스템 정책 생성 섹션을 참조하세요.

자동 탑재 실패 및 인스턴스 무응답

`_netdev` 옵션을 선언하지 않은 상태에서 파일 시스템을 인스턴스에 자동 탑재하는 경우에 발생할 수 있는 문제입니다. `_netdev`이 빠진 경우 EC2 인스턴스가 응답을 중지합니다. 컴퓨팅 인스턴스가 네트워킹을 시작한 후 네트워크 파일 시스템의 초기화를 완료해야 하기 때문입니다.

취할 조치

이 문제가 발생하면 AWS Support에 문의하십시오.

/etc/fstab에 여러 Amazon EFS 파일 시스템을 탑재하는 데 실패

`/etc/fstab`에 2개 이상의 Amazon EFS 항목이 있는 systemd init system을 사용하는 인스턴스의 경우, 항목 일부 또는 전체가 탑재되지 않는 경우가 있을 수 있습니다. 이 경우, `dmesg` 출력에 다음과 유사하게 한 줄 이상의 줄이 표시됩니다.

```
NFS: nfs4_discover_server_trunking unhandled error -512. Exiting with error EIO
```

취할 조치

이 경우 `/etc/systemd/system/mount-nfs-sequentially.service`에 새 systemd service 파일을 생성하는 것이 좋습니다. 파일에 포함할 코드는 파일 시스템을 수동으로 탑재하는지 아니면 Amazon EFS 탑재 도우미를 사용하는지에 따라 달라집니다.

- 파일 시스템을 수동으로 탑재하는 경우 `ExecStart` 명령이 네트워크 파일 시스템(NFS4)을 가리켜야 합니다. 파일에 다음 코드를 붙여넣습니다.

  ```
  [Unit]
  ```

```
Description=Workaround for mounting NFS file systems sequentially at boot time
After=remote-fs.target

[Service]
Type=oneshot
ExecStart=/bin/mount -avt nfs4
RemainAfterExit=yes

[Install]
WantedBy=multi-user.target
```

- Amazon EFS 탑재 도우미를 사용하는 경우 전송 계층 보안(TLS)을 사용하려면 `ExecStart` 명령이 NFS4 대신 EFS를 가리켜야 합니다. 파일에 다음 코드를 붙여넣습니다.

```
[Unit]
Description=Workaround for mounting NFS file systems sequentially at boot time
After=remote-fs.target

[Service]
Type=oneshot
ExecStart=/bin/mount -avt efs
RemainAfterExit=yes

[Install]
WantedBy=multi-user.target
```

파일 생성 후 다음 두 명령을 실행합니다.

1. `sudo systemctl daemon-reload`
2. `sudo systemctl enable mount-nfs-sequentially.service`

그리고 Amazon EC2 인스턴스를 다시 시작합니다. 통상 1초 이내에 요청대로 파일 시스템이 탑재됩니다.

탑재 명령에 실패하고 "잘못된 fs 유형" 오류 메시지가 표시됨

탑재 명령에 실패하고 다음 오류 메시지가 표시됩니다.

```
mount: wrong fs type, bad option, bad superblock on 10.1.25.30:/,
missing codepage or helper program, or other error (for several filesystems
```

```
(e.g. nfs, cifs) you might need a /sbin/mount.<type> helper program)
In some cases useful info is found in syslog - try dmesg | tail or so.
```

취할 조치

이 메시지를 수신한 경우 `nfs-utils`(또는 Ubuntu의 경우 `nfs-common`) 패키지를 설치합니다. 자세한 내용은 NFS 클라이언트 설치 단원을 참조하십시오.

탑재 명령에 실패하고 "잘못된 탑재 옵션" 오류 메시지가 표시됨

탑재 명령에 실패하고 다음 오류 메시지가 표시됩니다.

```
mount.nfs: an incorrect mount option was specified
```

취할 조치

이 오류 메시지는 주로 사용 중인 Linux 배포판이 Network File System 버전 4.0 및 4.1(NFSv4)을 지원하지 않음을 의미합니다. 이러한 경우에 해당하는지 확인하려면 다음 명령을 실행합니다.

```
$ grep CONFIG_NFS_V4_1 /boot/config*
```

앞선 명령에서 `# CONFIG_NFS_V4_1 is not set`이 반환되면 사용 중인 Linux 배포판에서 NFSv4.1이 지원되지 않는 것입니다. NFSv4.1을 지원하는 Amazon Elastic Compute Cloud(Amazon EC2)용 Amazon Machine Image(AMI) 목록은 NFS 지원 섹션을 참조하세요.

액세스 포인트를 사용한 탑재 실패

액세스 포인트로 탑재할 때 탑재 명령이 실패하고 다음 오류 메시지가 표시됩니다.

```
mount.nfs4: mounting access_point failed, reason given by server: No such file or
 directory
```

취할 조치

이 오류 메시지는 지정된 EFS 경로가 존재하지 않는 것을 나타냅니다. 액세스 포인트 루트 디렉터리에 대한 소유권과 권한을 제공해야 합니다. EFS는 이 정보를 사용하여 디렉터리를 생성합니다. 자세한 내용은 액세스 포인트 작업 단원을 참조하십시오.

루트 디렉터리 소유권 및 권한을 지정하지 않고 루트 디렉터리가 아직 없는 경우 EFS는 루트 디렉터리를 생성하지 않습니다. 이 경우 액세스 포인트를 사용하여 파일 시스템을 탑재하려는 시도가 실패합니다.

파일 시스템 생성 직후 파일 시스템 탑재에 실패함

탑재 대상 생성 후 DNS(Domain Name Service) 레코드가 AWS 리전내에 완전히 전파되는 데 최대 90초까지 걸릴 수 있습니다.

취할 조치

예를 들어 CloudFormation 템플릿을 사용하여 프로그래밍 방식으로 파일 시스템을 생성하고 탑재하는 경우 대기 조건을 구현하는 것이 좋습니다.

파일 시스템 탑재가 중단된 후 실패하고 제한 시간 초과 오류가 표시됨

파일 시스템 탑재 명령이 1~2분 동안 중단된 후 실패하고 제한 시간 초과 오류가 표시됩니다. 다음 코드에 예가 나와 있습니다.

```
$ sudo mount -t nfs -o
 nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport mount-target-ip:/ mnt

[2+ minute wait here]
mount.nfs: Connection timed out
$Â
```

취할 조치

이 오류는 Amazon EC2 인스턴스 또는 탑재 대상 보안 그룹이 제대로 구성되지 않았기 때문에 발생할 수 있습니다. 탑재 대상 보안 그룹에 EC2 보안 그룹에서 포트 2049를 통한 NFS 액세스를 허용하는 인바운드 규칙이 있는지 확인합니다. 자세한 내용은 VPC 보안 그룹 사용 단원을 참조하십시오.

지정한 탑재 대상 IP 주소가 올바른지 확인합니다. 잘못된 IP 주소를 지정했고 해당 IP 주소에는 탑재를 거부할 다른 이유가 없는 경우 이 문제가 발생할 수 있습니다.

DNS 이름을 사용한 파일 시스템 탑재에 실패

`amazon-efs-utils` 클라이언트를 사용하지 않고 NFS 클라이언트를 사용하여 파일 시스템의 DNS 이름을 사용하여 파일 시스템을 탑재하려고 하면 다음 예와 같이 실패합니다.

```
$ sudo mount -t nfs -o
 nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport file-system-id.efs.aws-region.amazonaws.com:/ mnt
```

```
mount.nfs: Failed to resolve server file-system-id.efs.aws-region.amazonaws.com:
  Name or service not known.

$
```

취할 조치

VPC 구성을 확인합니다. 사용자 지정 VPC를 사용하는 경우, DNS 설정이 활성화되어 있는지 확인합니다. 자세한 내용을 알아보려면 Amazon VPC 사용 설명서의 VPC에 대한 DNS 속성을 참조하세요. 또한 파일 시스템 및 탑재 대상 DNS 이름은 존재하는 VPC 외부에서 확인할 수 없습니다.

`mount` 명령에서 DNS 이름을 사용하여 파일 시스템을 탑재하려면 먼저 다음과 같이 해야 합니다.

- Amazon EFS 탑재 대상이 Amazon EC2 인스턴스와 동일한 가용 영역에 있는지 확인합니다.
- Amazon EC2 인스턴스와 동일한 VPC에 탑재 대상이 있는지 확인합니다. 그렇지 않으면 다른 VPC에 있는 EFS 탑재 대상에 대해 DNS 이름 확인을 사용할 수 없습니다. 자세한 내용은 다른 AWS 계정 또는 VPC에서 EFS 파일 시스템 탑재 단원을 참조하십시오.
- Amazon에서 제공하는 DNS 서버를 사용하도록 구성된 Amazon VPC 내에서 Amazon EC2 인스턴스를 연결합니다. 자세한 정보는 Amazon VPC 사용 설명서의 Amazon VPC 내의 DHCP 옵션 세트를 참조하세요.
- 연결 중인 Amazon EC2 인스턴스의 Amazon VPC에 DNS 호스트 이름이 활성화되어 있는지 확인합니다. 자세한 내용을 알아보려면 Amazon VPC 사용 설명서의 VPC의 DNS 속성을 참조하세요.

파일 시스템 탑재에 실패하고 "nfs 응답 없음" 오류가 표시됨

`"nfs: server_name still not responding"`와 TCP(Transmission Control Protocol) 다시 연결 이벤트 시 메시지와 함께 Amazon EFS 파일 시스템 탑재에 실패한 경우.

취할 조치

네트워크 연결이 다시 구현될 때 NFS 클라이언트가 새로운 TCP 소스 포트를 사용하도록 `noresvport` 탑재 옵션을 사용하세요. 이렇게 하면 네트워크 복구 이벤트 후 네트워크를 중단 없이 사용할 수 있습니다.

탑재 대상 수명 주기가 특정 상태에서 멈춤

탑재 대상 수명 주기 상태가 creating 또는 deleting 상태에서 멈췄습니다.

취할 조치

`CreateMountTarget` 또는 `DeleteMountTarget` 호출을 다시 시도해 보세요.

탑재 대상 수명 주기 상태에 오류가 표시됨

탑재 대상 수명 주기 상태가 오류로 표시됩니다.

취할 조치

Virtual Private Cloud(VPC)의 호스팅 영역이 충돌하는 경우 Amazon EFS는 새 파일 시스템 탑재 대상에 필요한 도메인 이름 시스템(DNS) 레코드를 생성할 수 없습니다. Amazon EFS는 고객 소유 호스팅 영역 내에 새 레코드를 생성할 수 없습니다. `efs.<region>.amazonaws.com` DNS 범위가 충돌하는 호스팅 영역을 유지 관리해야 하는 경우 별도의 VPC에 호스팅 영역을 생성하세요. VPC의 DNS 고려 사항에 대한 자세한 내용은 VPC의 DNS 속성을 참조하세요.

이 문제를 해결하려면 VPC에서 충돌하는 `efs.<region>.amazonaws.com` 호스트를 삭제하고 탑재 대상을 다시 생성하세요. 탑재 대상 삭제에 대한 자세한 내용은 탑재 대상 생성 섹션을 참조하세요.

탑재가 응답하지 않음

Amazon EFS 탑재가 응답하지 않는 것으로 나타납니다. 예를 들어 `ls`와 같은 명령이 중단됩니다.

취할 조치

이 오류는 다른 애플리케이션이 대량의 데이터를 파일 시스템에 쓰는 경우 발생할 수 있습니다. 기록 중인 파일에 대한 액세스를 작업이 완료될 때까지 차단할 수 있습니다. 일반적으로, 기록 중인 파일에 액세스하려고 하는 모든 명령 또는 애플리케이션은 중단 상태로 보일 수 있습니다. 예를 들어, `ls` 명령이 기록 중인 파일에 접근하려고 하면 이 명령이 중단될 수 있습니다. 일부 Linux 배포판에서 `ls` 명령에 별칭을 지정해서 이 명령이 디렉터리 내용을 나열하는 것 외에 파일 속성까지 검색하도록 했기 때문입니다.

이 문제를 해결하려면 다음 예에서처럼 다른 애플리케이션이 Amazon EFS 탑재에 파일을 쓰고 있으며 `Uninterruptible sleep`(D) 상태인지 확인합니다.

```
$ ps aux | grep large_io.py
root 33253 0.5 0.0 126652 5020 pts/3 D+ 18:22 0:00 python large_io.py /efs/large_file
```

여기에 해당하는 사례임이 확인되면 다른 쓰기 작업이 완료될 때까지 기다리거나 차선책을 실행해 문제를 해결할 수 있습니다. `ls` 예제에서는 별칭 대신 `/bin/ls` 명령을 직접 사용할 수 있습니다. 이렇게 하면 파일 쓰기를 중단하지 않고 명령을 처리할 수 있습니다. 일반적으로, 데이터를 쓰는 애플리케이션이 `fsync(2)` 등을 사용하여 정기적으로 데이터 플러시를 강행하면 다른 응용 프로그램에 대한 파일

시스템의 응답성이 개선될 수 있습니다. 그러나 이렇게 하면 해당 애플리케이션이 데이터를 쓸 때는 성능이 저하될 수 있습니다.

탑재된 클라이언트의 연결이 끊깁니다.

Amazon EFS 파일 시스템에 탑재된 클라이언트는 여러 가지 원인으로 인해 연결이 끊길 수 있습니다. NFS 클라이언트는 중단 시 자동으로 다시 연결되도록 설계되어 일상적인 연결 끊김이 애플리케이션 성능 및 가용성에 미치는 영향을 최소화합니다. 대부분의 경우 클라이언트는 몇 초 내에 투명하게 다시 연결됩니다.

그러나 이전 버전의 Linux 커널(버전 v5.4 이하)에 포함된 NFS 클라이언트 소프트웨어에는 연결이 끊긴 경우 NFS 클라이언트가 동일한 TCP 소스 포트에서 재연결을 시도하도록 하는 동작이 포함되어 있습니다. 이 동작은 TCP RFC를 준수하지 않으므로 이러한 클라이언트가 NFS 서버(이 경우 EFS 파일 시스템)에 대한 연결을 빠르게 다시 설정하지 못할 수 있습니다.

이 문제를 해결하려면 Amazon EFS 탑재 도우미를 사용하여 EFS 파일 시스템을 탑재하는 것이 좋습니다. EFS 탑재 도우미는 Amazon EFS 파일 시스템에 최적화된 탑재 설정을 사용합니다. EFS 클라이언트 및 탑재 도우미에 대한 자세한 내용은 Amazon EFS 클라이언트 수동 설치 섹션을 참조하세요.

EFS 탑재 도우미를 사용할 수 없는 경우 이 문제를 방지하려면 NFS 클라이언트가 새 TCP 소스 포트를 사용하여 연결을 다시 설정하도록 지시하는 `noresvport` NFS 탑재 옵션을 사용하는 것이 좋습니다. 자세한 내용은 권장 NFS 탑재 설정 단원을 참조하십시오.

새로 탑재한 파일 시스템에 대한 작업이 "잘못된 파일 핸들" 오류를 반환함

새로 탑재한 파일 시스템에 대해 수행한 작업이 `bad file handle` 오류를 반환합니다.

Amazon EC2 인스턴스가 지정된 IP 주소를 사용하여 파일 시스템 하나와 탑재 대상 하나에 연결되어 있는데 파일 시스템 및 탑재 대상이 삭제된 경우 이 오류가 발생할 수 있습니다. 새 파일 시스템 및 탑재 대상을 만들어 동일한 탑재 대상 IP 주소를 사용하는 Amazon EC2 인스턴스에 연결하면 이 문제가 발생할 수 있습니다.

취할 조치

이 문제를 파일 시스템을 탑재 해제한 다음 Amazon EC2 인스턴스에 파일 시스템을 다시 탑재하여 해결할 수 있습니다. Amazon EFS 파일 시스템 탑재 해제에 대한 자세한 내용은 파일 시스템 탑재 해제 섹션을 참조하세요.

파일 시스템 탑재 해제 실패

파일 시스템이 사용 중인 경우 탑재를 해제할 수 없습니다.

취할 조치

다음 방법을 통해 이 문제를 해결할 수 있습니다.

- 지연 탑재 해제를 사용하면 실행 시 파일 시스템 계층 구조에서 파일 시스템을 분리한 umount -l가 더 이상 사용량이 많지 않은 파일 시스템에 대한 모든 참조를 즉시 정리합니다.
- 읽기 및 쓰기 작업이 완료되기 기다린 후 다시 umount 명령을 실행합니다.
- umount -f 명령을 사용하여 강제로 탑재 해제합니다.

> **Warning**
>
> 탑재 해제를 강제 적용하면 현재 파일 시스템에서 수행되고 있는 데이터 읽기 및 쓰기 작업이 중단됩니다. 이 옵션을 사용하는 방법에 대한 자세한 내용 및 지침은 umount 매뉴얼 페이지를 참조하세요.

Amazon EFS로 또는 Amazon EFS에서 데이터 전송

AWS DataSync 및 AWS Transfer Family 를 사용하여 Amazon EFS 파일 시스템 안팎으로 데이터를 전송할 수 있습니다. AWS DataSync 는 NFS(Network File System), SMB(Server Message Block) 파일 서버, 자체 관리형 객체 스토리지 및 서비스 간에 AWS 데이터를 복사할 수 있는 온라인 데이터 전송 서비스입니다. Amazon EFS에서 DataSync를 사용하는 방법에 대한 자세한 내용은 AWS DataSync 를 사용하여 데이터 전송 섹션을 참조하세요.

AWS Transfer Family 는 Secure File Transfer Protocol(SFTP), File Transfer Protocol(FTP) 및 FTP over Secure Sockets Layer(FTPS) 프로토콜을 통해 Amazon EFS 파일 시스템으로 또는 Amazon EFS 파일 시스템에서 파일을 전송하는 데 사용할 수 있는 완전 관리형 AWS 서비스입니다. Transfer Family 를 사용하면 비즈니스 파트너에게 데이터 배포, 공급망, 콘텐츠 관리 및 웹 서비스 애플리케이션과 같은 사용 사례에 대해 Amazon EFS 파일 시스템에 저장된 파일에 대한 액세스 권한을 제공할 수 있습니다. Transfer Family를 Amazon EFS에서 사용하는 방법에 대한 자세한 내용은 AWS Transfer Family 를 사용하여 데이터 전송 섹션을 참조하세요.

주제

- AWS DataSync 를 사용하여 데이터 전송
- AWS Transfer Family 를 사용하여 데이터 전송

AWS DataSync 를 사용하여 데이터 전송

AWS DataSync 는 온프레미스 스토리지 시스템 간 및 스토리지 서비스 간 AWS 데이터 이동 및 복제를 간소화, 자동화 및 가속화하는 온라인 데이터 전송 서비스입니다. DataSync는 NFS(Network File System), SMB(Server Message Block) 파일 서버, 자체 관리형 객체 스토리지 AWS Snowball Edge, Amazon S3 버킷, EFS 파일 시스템 및 FSx for Windows File Server 파일 시스템 간에 데이터를 복사할 수 있습니다.

또한 DataSync를 사용하여 서로 다른의 파일 시스템과 서로 다른 AWS 리전이 소유한 파일 시스템을 포함하여 두 EFS 파일 시스템 간에 파일을 전송할 수 AWS 계정있습니다. DataSync를 사용하여 EFS 파일 시스템 간에 데이터를 복사하면 일회성 데이터 마이그레이션을 수행하고, 분산 워크로드를 위해 주기적으로 데이터 모으기를 수행하며, 복제를 자동화하여 데이터를 보호 및 복구할 수 있습니다.

자세한 내용은 Amazon EFS 시작하기 및 AWS DataSync 사용 설명서를 참조하세요.

AWS Transfer Family 를 사용하여 데이터 전송

AWS Transfer Family 는 다음 프로토콜을 통해 Amazon EFS 파일 시스템 안팎으로 파일을 전송하는 데 사용할 수 있는 완전 관리형 AWS 서비스입니다.

- Secure Shell(SSH) File Transfer 프로토콜(SFTP)(AWS Transfer for SFTP)
- File Transfer 프로토콜 보안(FTPS)(AWS Transfer for FTPS)
- File Transfer 프로토콜(FTP)(AWS Transfer for FTP)

Transfer Family를 사용하면 공급업체, 파트너 또는 고객과 같은 제3자가 인프라를 관리할 필요 없이 전 세계 규모로 지원되는 프로토콜을 통해 파일에 안전하게 액세스할 수 있습니다. 또한 이제 SFTP, FTPS 및 FTP 클라이언트를 사용하여 Windows, macOS 및 Linux 환경에서 EFS 파일 시스템에 쉽게 액세스할 수 있습니다. 이를 통해 NFS 클라이언트 및 액세스 포인트를 넘어 여러 환경의 사용자까지 데이터 접근성을 확장할 수 있습니다.

Transfer Family를 사용하여 Amazon EFS 파일 시스템에서 데이터를 전송하는 것은 다른 클라이언트 사용과 동일한 방식으로 처리됩니다. 자세한 내용은 처리량 모드 및 Amazon EFS 할당량 섹션을 참조하세요.

에 대해 자세히 알아보려면 AWS Transfer Family 사용 설명서를 AWS Transfer Family참조하세요.

> **Note**
>
> Amazon EFS에서 Transfer Family를 사용하는 것은 2021년 AWS 계정 1월 6일 이전에 생성된 퍼블릭 액세스를 허용하는 정책이 있는 Amazon EFS 파일 시스템이 있는에 대해 기본적으로 비활성화됩니다. Transfer Family를 사용하여 파일 시스템에 액세스할 수 있도록 하려면에 문의하세요 지원.

주제

- Amazon EFS AWS Transfer Family 에서를 사용하기 위한 사전 조건
- 에 대한 EFS 파일 시스템 구성 AWS Transfer Family

Amazon EFS AWS Transfer Family 에서를 사용하기 위한 사전 조건

Transfer Family를 사용하여 Amazon EFS 파일 시스템의 파일에 액세스하려면 구성이 다음 조건을 충족해야 합니다.

- Transfer Family 서버와 Amazon EFS 파일 시스템은 같은 AWS 리전에 있습니다.
- Transfer Family에서 사용하는 IAM 역할에는 Amazon EFS 권한(예: `elasticfilesystem:ClientMount` 및 `elasticfilesystem:ClientWrite`)이 연결되어 있어야 합니다. 권한 및 정책에 대한 자세한 내용은 AWS Transfer Family 사용 설명서의 IAM 역할 및 정책 생성을 참조하세요.
- (선택 사항)Transfer Family 서버를 다른 계정에서 소유한 경우 크로스 계정 액세스를 활성화하세요.
  - 파일 시스템 정책이 공개 액세스를 허용하지 않는지 확인하세요. 자세한 내용은 EFS 파일 시스템에 대한 퍼블릭 액세스 차단 단원을 참조하십시오.
  - 크로스 계정 액세스를 활성화하도록 파일 시스템 정책을 수정합니다. 자세한 내용은 Transfer Family에 대한 크로스 계정 액세스 구성 단원을 참조하십시오.

에 대한 EFS 파일 시스템 구성 AWS Transfer Family

Transfer Family와 함께 작동하도록 Amazon EFS 파일 시스템을 구성하려면 다음 단계가 필요합니다.

- 단계 1. Transfer Family 사용자에게 할당된 POSIX ID 목록을 가져옵니다.
- 단계 2. Transfer Family 사용자에게 할당된 POSIX ID를 사용하여 Transfer Family 사용자가 파일 시스템의 디렉터리에 액세스할 수 있는지 확인합니다.
- 단계 3. Transfer Family에서 사용하는 IAM 역할에 액세스할 수 있도록 IAM을 구성합니다.

Transfer Family 사용자의 파일 및 디렉터리 권한 설정

Transfer Family 사용자가 EFS 파일 시스템의 필요한 파일 및 디렉터리에 액세스할 수 있는지 확인하세요. Transfer Family 사용자에게 할당된 POSIX ID 목록을 사용하여 디렉터리에 액세스 권한을 할당합니다. 이 예제에서 사용자는 Amazon EFS 탑재 지점 아래에 `transferFam`이라는 이름이 지정된 디렉터리를 생성합니다. 사용 사례에 따라 디렉터리 생성은 선택 사항입니다. 필요한 경우 EFS 파일 시스템에서 이름과 위치를 선택할 수 있습니다.

Transfer Family의 POSIX 사용자에게 파일 및 디렉터리 권한을 할당하려면

1. Amazon EC2 인스턴스에 연결합니다. Amazon EFS는 Linux 기반 EC2 인스턴스에 의한 탑재만 지원합니다.
2. EFS 파일 시스템이 EC2 인스턴스에 아직 탑재되어 있지 않은 경우 이를 탑재합니다. 자세한 내용은 EFS 파일 시스템 탑재 단원을 참조하십시오.
3. 다음 예에서는 EFS 파일 시스템에 디렉터리를 생성하고 해당 그룹을 Transfer Family 사용자의 POSIX 그룹 ID(이 예에서는 1101)로 변경합니다.

a. 다음 명령을 실행해 `efs/transferFam` 디렉터리를 생성합니다. 실제로는 선택한 파일 시스템의 이름과 위치를 사용할 수 있습니다.

```
[ec2-user@ip-192-0-2-0 ~]$ ls
efs  efs-mount-point  efs-mount-point2
[ec2-user@ip-192-0-2-0 ~]$ ls efs
[ec2-user@ip-192-0-2-0 ~]$ sudo mkdir efs/transferFam
[ec2-user@ip-192-0-2-0 ~]$ ls -l efs
total 0
drwxr-xr-x 2 root root 6 Jan  6 15:58 transferFam
```

b. 다음 명령을 사용하여 Transfer Family 사용자에게 할당된 POSIX GID로 `efs/transferFam` 그룹을 변경합니다.

```
[ec2-user@ip-192-0-2-0 ~]$ sudo chown :1101 efs/transferFam/
```

c. 변경을 확인합니다.

```
[ec2-user@ip-192-0-2-0 ~]$ ls -l efs
total 0
drwxr-xr-x 2 root 1101 6 Jan  6 15:58 transferFam
```

Transfer Family에서 사용하는 IAM 역할에 대한 액세스를 활성화합니다.

Transfer Family에서는 EFS 파일 시스템에 대한 사용자 액세스를 정의하는 리소스 기반 IAM 정책과 IAM 역할을 생성합니다. 자세한 내용은AWS Transfer Family 사용 설명서의 IAM 역할 및 정책 생성을 참조하세요. IAM 자격 증명 정책 또는 파일 시스템 정책을 사용하여 EFS 파일 시스템에 대한 Transfer Family IAM 역할 액세스 권한을 부여해야 합니다.

다음은 IAM 역할 `EFS-role-for-transfer`에 `ClientMount`(읽기) 및 `ClientWrite` 액세스 권한을 부여하는 파일 시스템 정책의 예시입니다.

JSON

```
{
    "Version":"2012-10-17",
    "Id": "efs-policy-wizard-8698b356-4212-4d30-901e-ad2030b57762",
    "Statement": [
        {
```

```
            "Sid": "Grant-transfer-role-access",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:role/EFS-role-for-transfer"
            },
            "Action": [
                "elasticfilesystem:ClientWrite",
                "elasticfilesystem:ClientMount"
            ],
            "Resource": "arn:aws:elasticfilesystem:us-east-1:111122223333:file-
system/*"
        }
    ]
}
```

파일 시스템 정책 생성에 대한 자세한 내용은 파일 시스템 정책 생성 섹션을 참조하세요. 자격 증명 기반 IAM 정책을 사용하여 EFS 리소스에 대한 액세스를 제한하는 방법에 대한 자세한 내용은 Amazon EFS의 자격 증명 기반 정책 섹션을 참조하세요.

Transfer Family에 대한 크로스 계정 액세스 구성

파일 시스템에 액세스하는 데 사용되는 Transfer Family 서버가 다른에 속하는 경우 해당 계정에 파일 시스템에 대한 액세스 권한을 부여 AWS 계정해야 합니다. 또한 파일 시스템 정책은 비공개여야 합니다. 파일 시스템에 대한 퍼블릭 액세스 차단에 대한 자세한 내용은 EFS 파일 시스템에 대한 퍼블릭 액세스 차단 섹션을 참조하세요.

파일 시스템 정책에서 파일 시스템에 다른 AWS 계정 액세스 권한을 부여할 수 있습니다. Amazon EFS 콘솔에서 파일 시스템 정책 편집기의 추가 권한 부여 섹션을 사용하여 부여하려는 파일 시스템 액세스 수준 AWS 계정 및를 지정합니다. 파일 시스템 정책 생성 또는 편집에 대한 자세한 내용은 파일 시스템 정책 생성 섹션을 참조하세요.

계정 ID 또는 계정 Amazon 리소스 이름(ARN)을 사용하여 계정을 지정할 수 있습니다. ARN에 대한 자세한 내용은 IAM 사용 설명서의 IAM ARN 단원을 참조하세요.

다음 예는 파일 시스템에 대한 크로스 계정 액세스 권한을 부여하는 비공개 파일 시스템 정책입니다. 여기에는 다음과 같은 두 개의 명령문이 있습니다.

1. 첫 번째 명령문인 `NFS-client-read-write-via-fsmt`는 파일 시스템 탑재 대상을 사용하여 파일 시스템에 액세스하는 NFS 클라이언트에 읽기, 쓰기 및 루트 권한을 부여합니다.

2. 두 번째 문인는 계정의이 EFS 파일 시스템에 액세스해야 하는 Transfer Family 서버를 소유한 계정인 AWS 계정 111122223333에 읽기 및 쓰기 권한만 `Grant-cross-account-access`부여합니다.

```
{
    "Statement": [
        {
            "Sid": "NFS-client-read-write-via-fsmt",
            "Effect": "Allow",
            "Action": [
                "elasticfilesystem:ClientRootAccess",
                "elasticfilesystem:ClientWrite",
                "elasticfilesystem:ClientMount"
            ],
            "Condition": {
                "Bool": {
                    "elasticfilesystem:AccessedViaMountTarget": "true"
                }
            }
        },
        {
            "Sid": "Grant-cross-account-access",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:root"
            },
            "Action": [
                "elasticfilesystem:ClientWrite",
                "elasticfilesystem:ClientMount"
            ]
        }
    ]
}
```

다음 파일 시스템 정책은 Transfer Family에서 사용하는 IAM 역할에 대한 액세스 권한을 부여하는 설명문을 추가합니다.

```
{
    "Statement": [
        {
            "Sid": "NFS-client-read-write-via-fsmt",
```

```
            "Effect": "Allow",
            "Action": [
                "elasticfilesystem:ClientRootAccess",
                "elasticfilesystem:ClientWrite",
                "elasticfilesystem:ClientMount"
            ],
            "Condition": {
                "Bool": {
                    "elasticfilesystem:AccessedViaMountTarget": "true"
                }
            }
        },
        {
            "Sid": "Grant-cross-account-access",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:root"
            },
            "Action": [
                "elasticfilesystem:ClientWrite",
                "elasticfilesystem:ClientMount"
            ]
        },
        {
            "Sid": "Grant-transfer-role-access",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:role/EFS-role-for-transfer"
            },
            "Action": [
                "elasticfilesystem:ClientWrite",
                "elasticfilesystem:ClientMount"
            ]
        }
    ]
}
```

EFS 파일 시스템 관리

파일 시스템 관리 태스크에는 탑재 대상에 대한 파일 시스템의 네트워크 접근성 관리, 처리량 모드 변경, 수명 주기 정책 업데이트, 암호화 관리가 포함됩니다.

다음 단원에서 설명한 대로 AWS Management Console를 사용하거나 (AWS CLI) 또는 API를 AWS Command Line Interface 사용하여 프로그래밍 방식으로 이러한 파일 시스템 관리 작업을 수행할 수 있습니다.

주제

- 파일 시스템 상태에 대한 이해
- 탑재 대상 생성
- 파일 시스템 처리량 관리
- 스토리지 수명 주기 관리

파일 시스템 상태에 대한 이해

Amazon EFS 콘솔 또는 AWS CLI를 사용하여 Amazon EFS 파일 시스템의 상태를 볼 수 있습니다. Amazon EFS 파일 시스템은 다음 표에 설명된 상태 값 중 하나를 가질 수 있습니다.

파일 시스템 상태	설명
사용 가능	파일 시스템이 정상 상태이며 접속하여 사용할 수 있습니다.
CREATING	Amazon EFS는 새 파일 시스템을 생성하는 중입니다.
DELETING	Amazon EFS는 사용자가 시작한 삭제 요청에 대한 응답으로 파일 시스템을 삭제합니다. 자세한 내용은 EFS 파일 시스템 삭제 단원을 참조하십시오.
DELETED	Amazon EFS는 사용자가 시작한 삭제 요청에 대한 응답으로 파일 시스템을 삭제했습니다. 자세한 내용은 EFS 파일 시스템 삭제 단원을 참조하십시오.
업데이트 중	사용자가 시작한 업데이트 요청에 대한 응답으로 파일 시스템이 업데이트 중입니다.
오류	복제 구성의 파일 시스템을 포함하여 One Zone 파일 시스템에 적용됩니다.

파일 시스템 상태	설명
	파일 시스템이 장애 상태이며 복구할 수 없습니다. 파일 시스템 데이터에 액세스하려면 이 파일 시스템의 백업을 새 파일 시스템에 복원하세요. 자세한 내용은 Amazon EFS에서 데이터 백업 및 복제 섹션을 참조하세요.

탑재 대상 생성

파일 시스템에 대해 생성한 하나 이상의 탑재 대상을 사용하여 Virtual Private Cloud(VPC)의 Amazon EC2 또는 기타 AWS 컴퓨팅 인스턴스에 파일 시스템을 탑재합니다. 탑재 대상은 파일 시스템 생성 시 또는 생성 후에 생성할 수 있습니다.

파일 시스템에 대한 탑재 대상이 생성된 후 추가 탑재 대상을 생성하고, 탑재 대상을 삭제하고, 탑재 대상의 보안 그룹을 수정할 수 있습니다. 탑재 대상의 VPC를 수정하려면 먼저 기존 탑재 대상을 삭제해야 합니다.

> **Note**
>
> 기존 탑재 대상의 IP 주소는 변경할 수 없습니다. IP 주소를 변경하려면 탑재 대상을 삭제한 다음 새 주소로 탑재 대상을 새로 생성해야 합니다.

주제

- 탑재 대상 및 가용 영역
- 탑재 대상 생성
- 탑재 대상 삭제
- 탑재 대상 VPC 변경
- 탑재 대상 보안 그룹 변경

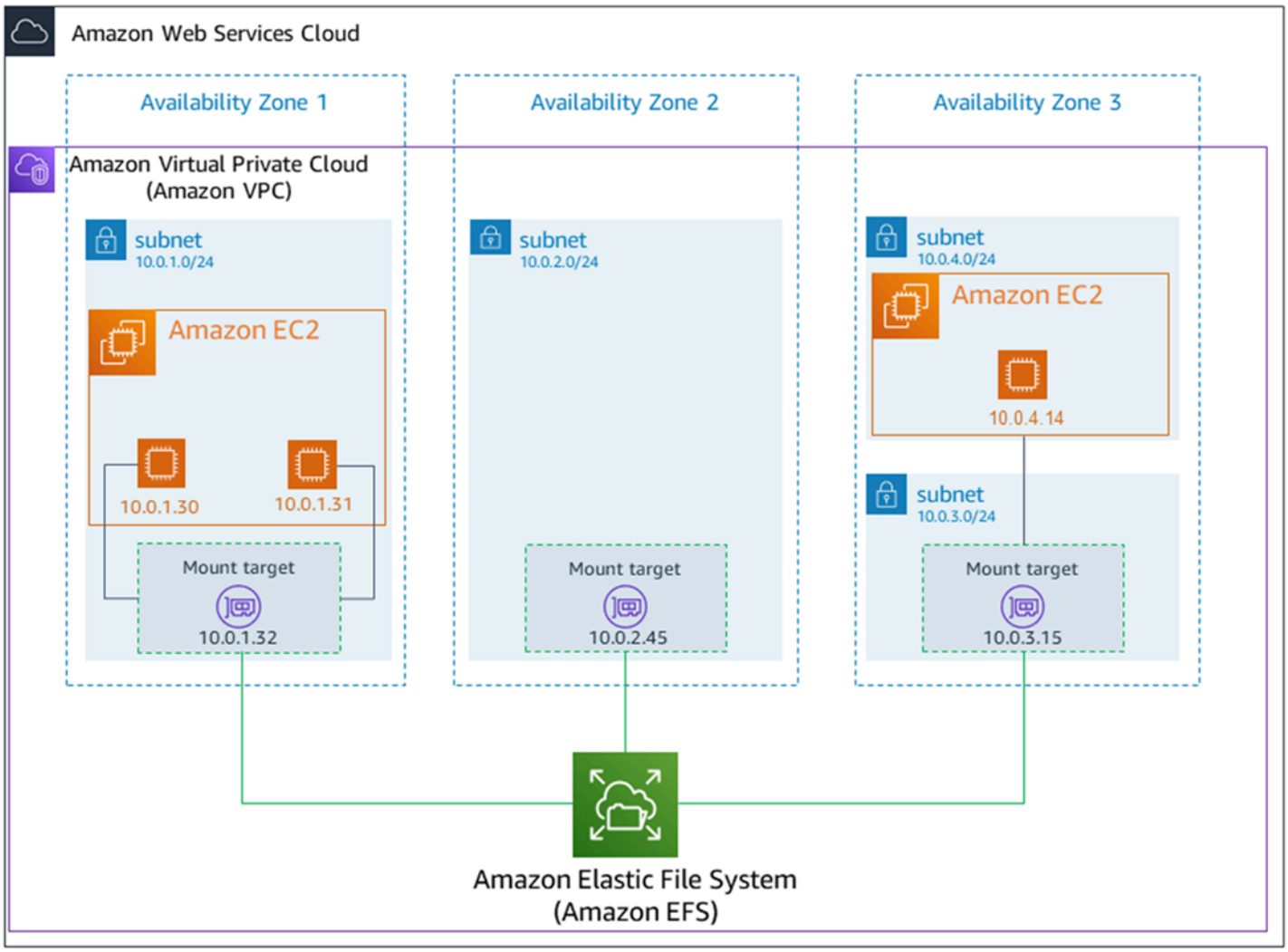
탑재 대상 및 가용 영역

리전별 스토리지 클래스를 사용하는 EFS 파일 시스템의 경우, AWS 리전의 각 가용 영역에 탑재 대상을 생성할 수 있습니다.

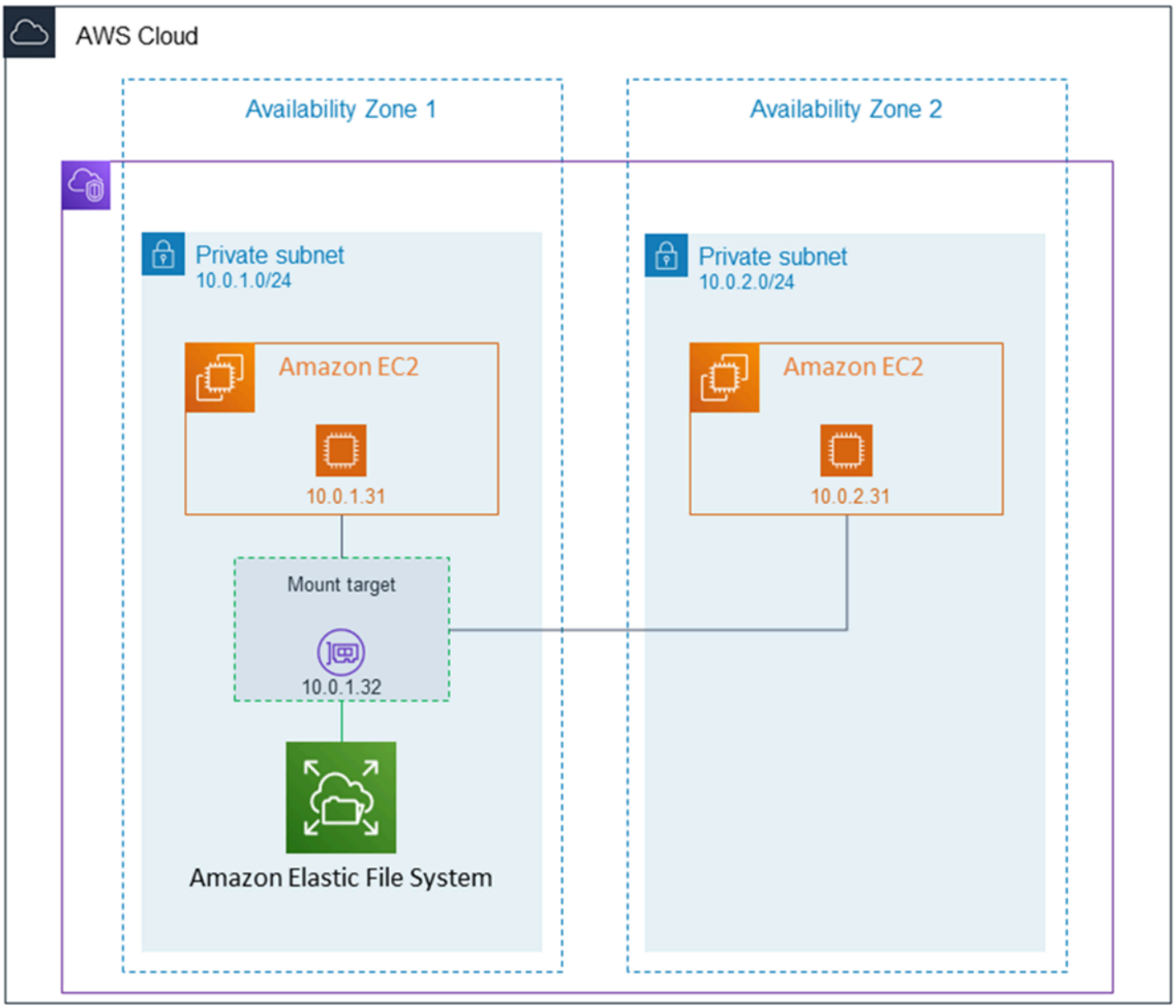
One Zone 파일 시스템의 경우 파일 시스템과 동일한 가용 영역에 단일 탑재 대상만 생성할 수 있습니다. 그런 다음 Amazon EC2, Amazon ECS 및 Virtual Private Cloud(VPC)를 포함한 컴퓨팅 인스턴스 AWS Lambda 에 파일 시스템을 탑재할 수 있습니다.

다음 다이어그램은 VPC의 모든 가용 영역에 탑재 대상이 생성된 리전 파일 시스템을 보여줍니다. 이 그림에서는 다른 VPC 서브넷에서 시작된 EC2 인스턴스 3개가 EFS 파일 시스템에 액세스합니다. 또한 각 가용 영역의 서브넷 수와 관계없이 가용 영역마다 탑재 대상이 하나씩 있는 것을 알 수 있습니다.

가용 영역당 탑재 대상은 한 개만 만들 수 있습니다. 그림의 한 가용 영역에서 보듯이, 가용 영역에 서브넷이 여러 개 있는 경우에는 여러 서브넷 중 하나에서만 탑재 대상을 만듭니다. 가용 영역에 탑재 대상이 하나만 있는 한 해당 가용 영역의 서브넷 중 하나에서 시작된 EC2 인스턴스는 동일한 탑재 대상을 공유할 수 있습니다.



다음 다이어그램은 파일 시스템과 동일한 가용 영역에 단일 탑재 대상을 생성한 One Zone 파일 시스템을 보여줍니다. us-west2c 가용 영역에서 EC2 인스턴스를 사용하여 파일 시스템에 액세스하면 탑재 대상과 다른 가용 영역에 위치하기 때문에 데이터 액세스 요금이 발생합니다.



탑재 대상 보안 그룹은 트래픽을 제어하는 가상 방화벽의 역할을 수행합니다. 예를 들어, 보안 그룹은 파일 시스템에 액세스할 수 있는 클라이언트를 결정합니다. 이 섹션에서는 다음에 대해 설명합니다.

- 탑재 대상 보안 그룹 및 트래픽 활성화 관리.
- 클라이언트에 파일 시스템 탑재.
- NFS 수준 권한 고려 사항

처음에는 Amazon EC2 인스턴스의 루트 사용자만 파일 시스템에 대한 읽기-쓰기-실행 권한을 가지고 있습니다. 이 주제에서는 NFS 수준 권한에 대해 살펴보고 일반적인 시나리오에서 권한을 부여하는 방법을 보여주는 예를 제공합니다. 자세한 내용은 NFS(Network File System) 수준 사용자, 그룹 및 권한 사용 단원을 참조하십시오.

탑재 대상 생성

VPC에서 EFS 파일 시스템에 액세스하려면 해당 파일 시스템에 대한 탑재 대상을 생성해야 합니다.

EFS 파일 시스템에는 다음이 적용됩니다.

- 탑재 대상은 한 번에 VPC 하나의 파일 시스템에 대해 여러 개를 만들 수 있습니다. 다른 VPC에서 파일 시스템에 액세스하려면 현재 VPC에서 탑재 대상을 삭제한 후 다른 VPC에 새 탑재 대상을 생성해야 합니다. 자세한 내용은 탑재 대상 VPC 변경 단원을 참조하십시오.
- VPC의 가용 영역에 서브넷이 여러 개 있는 경우 여러 서브넷 중 하나에만 탑재 대상을 만들 수 있습니다. 가용 영역의 모든 EC2 인스턴스는 탑재 대상 하나를 공유할 수 있습니다.
- 파일 시스템에 액세스하려는 각 가용 영역에 탑재 대상을 적어도 하나씩 생성해야 합니다.

> **Note**
>
> 다른 가용 영역에서 만든 탑재 대상을 통해 가용 영역의 EC2 인스턴스에 파일 시스템을 탑재하기 위해서는 비용과 관련된 고려 사항이 있습니다. 자세한 내용은 Amazon EFS 요금을 참조하세요. 또한 항상 인스턴스의 가용 영역에 로컬로 배치된 탑재 대상을 사용하여 부분적인 장애 가능성을 없앱니다. 탑재 대상의 영역이 다운되면 해당 탑재 대상을 통해 파일 시스템에 액세스할 수 없습니다.

를 사용하거나 SDK를 사용하여 AWS Management Console AWS CLI프로그래밍 방식으로 파일 시스템의 탑재 대상을 생성할 수 있습니다. AWS SDKs 콘솔에서는 파일 시스템을 생성할 때 또는 생성 후 탑재 대상을 생성할 수 있습니다. 파일 시스템 생성 시 탑재 대상 생성 방법에 대한 지침은 콘솔을 사용하여 사용자 지정 생성 섹션을 참조하세요.

콘솔 사용

기존 EFS 파일 시스템에 탑재 대상을 추가하려면 다음 절차를 따르세요.

EFS 파일 시스템에 탑재 대상을 생성하는 방법

1. Amazon Elastic File System 콘솔(https://console.aws.amazon.com/efs/)을 엽니다.
2. 왼쪽 탐색 창에서 파일 시스템을 선택한 다음 VPC를 변경하려는 파일 시스템을 선택합니다.
3. 네트워크를 선택한 다음 관리를 선택하여 파일 시스템의 탑재 대상을 표시합니다.
4. 이름 또는 파일 시스템 ID를 선택하여 탑재 대상을 추가할 파일 시스템을 선택합니다.

   > **Note**
   >
   > One Zone 파일 시스템의 경우 파일 시스템과 동일한 가용 영역에 있는 단일 탑재 대상만 생성할 수 있습니다.

5. EFS 리전별 스토리지 클래스를 사용하는 파일 시스템의 경우, 해당 파일 시스템에 대해 생성하려는 각 탑재 대상에 대해 탑재 대상 추가를 선택합니다.
6. 탑재 대상 설정을 정의합니다.

   a. 탑재 대상의 가용 영역 및 서브넷 ID를 선택합니다.

   b. IP 주소 유형에서 IPv4 전용을 선택하면 IPv4 주소만 지원하고, IPv6 전용을 선택하면 IPv6 주소만 지원하며, 듀얼 스택을 선택하면 IPv4 및 IPv6 주소를 모두 지원합니다.

      > **Note**
      >
      > IP 주소 유형은 서브넷의 IP 유형과 일치해야 합니다. 또한 IP 주소 유형은 서브넷의 IP 주소 지정 속성을 재정의합니다. 예를 들어, IP 주소 유형이 IPv4 전용이고 서브넷에 대해 IPv6 주소 지정 속성이 활성화된 경우, 해당 서브넷에서 생성된 네트워크 인터페이스는 서브넷의 주소 범위에서 IPv4 주소를 할당받습니다. 자세한 내용은 서브넷의 IP 주소 지정 속성 수정을 참조하세요.

   c. 탑재 대상을 배치할 IP 주소를 알고 있는 경우 IP 주소 유형과 일치하는 IP 주소 상자에 해당 주소를 입력합니다. 값을 지정하지 않으면 Amazon EFS가 지정된 서브넷에서 사용되지 않은 IP 주소를 선택합니다.

      > **Note**
      >
      > 탑재 대상 생성 후 IP 주소를 변경할 수 없습니다. IP 주소를 변경하려면 탑재 대상을 삭제한 다음 새 주소로 탑재 대상을 새로 생성해야 합니다.

7. 탑재 대상과 연결할 보안 그룹을 하나 이상 선택합니다. 나중에 보안 그룹을 수정할 수 있습니다.

8. 저장을 선택합니다.

사용 AWS CLI

이 섹션에서는 AWS CLI 에서 `create-mount-target` 명령을 사용하여 탑재 대상 생성 예제를 제공합니다. 동등한 API 명령은 CreateMountTarget입니다.

- 탑재 대상에 IP 주소 유형을 지정하지 않으면 IPv4 전용이 사용됩니다.
- 탑재 대상의 IP 주소를 지정하지 않으면 Amazon EFS가 지정된 서브넷에 사용 가능한 주소를 할당합니다.
- IP 주소 유형은 서브넷의 IP 주소 지정 속성을 재정의합니다. 예를 들어, IP 주소 유형이 IPv4 전용이고 서브넷에 대해 IPv6 주소 지정 속성이 활성화된 경우, 해당 서브넷에서 생성된 네트워크 인터페이스는 서브넷의 주소 범위에서 IPv4 주소를 할당받습니다. 자세한 내용은 서브넷의 IP 주소 지정 속성 수정을 참조하세요.

> **Note**
>
> One Zone 파일 시스템의 경우 파일 시스템과 동일한 가용 영역에 있는 단일 탑재 대상만 생성할 수 있습니다.

예: 서브넷에서 사용 가능한 IPv4 주소에 탑재 대상 생성

다음 명령은 탑재 대상의 파일 시스템, 서브넷 및 보안 그룹을 지정합니다. 대상은 지정된 서브넷의 사용 가능한 IPv4 주소에서 생성됩니다.

```
$ aws efs create-mount-target \
--file-system-id file-system-id \
--subnet-id  subnet-id \
--security-group ID-of-the-security-group-created-for-mount-target \
--region aws-region \
```

다음 예제는 샘플 데이터가 있는 명령을 나타낸 것입니다.

```
$ aws efs create-mount-target \
--file-system-id fs-0123456789abcdef1 \
--subnet-id  subnet-b3983dc4 \
--security-group sg-01234567 \
```

```
--region us-east-2 \
```

탑재 대상을 만든 후 Amazon EFS에서는 다음 예에서처럼 탑재 대상 설명을 JSON으로 반환합니다.

```
{
    "OwnerID": "111122223333"
    "MountTargetId": "fsmt-f9a14450",
    "FileSystemId": "fs-0123456789abcdef1",
    "SubnetId": "subnet-b3983dc4",
    "LifeCycleState": "available",
    "IpAddress": "10.0.1.24",
    "NetworkInterfaceId": "eni-3851ec4e",
    "AvailabilityZoneId": "use2-az1",
    "AvailabilityZoneName": "us-east-2a",
    "VpcId": "vpc-3c39ef57"
}
```

예: 특정 IPv4 주소에서 탑재 대상 생성

다음 명령은 탑재 대상에 사용할 파일 시스템, 서브넷, 보안 그룹 및 IPv4 주소를 지정합니다. 대상은 지정된 서브넷의 지정된 IPv4 주소에 생성됩니다.

```
$ aws efs create-mount-target \
--file-system-id file-system-id \
--subnet-id  subnet-id \
--security-group ID-of-the-security-group-created-for-mount-target \
--ip-address IPv4-address
--region aws-region \
```

다음 예제는 샘플 데이터가 있는 명령을 나타낸 것입니다.

```
$ aws efs create-mount-target \
--file-system-id fs-0123456789abcdef1 \
--subnet-id  subnet-b3983dc4 \
--security-group sg-01234567 \
--ip-address 10.0.1.24 \
--region us-east-2 \
```

탑재 대상을 만든 후 Amazon EFS에서는 다음 예에서처럼 탑재 대상 설명을 JSON으로 반환합니다.

```
{
```

```
    "OwnerID": "111122223333"
    "MountTargetId": "fsmt-f9a14450",
    "FileSystemId": "fs-0123456789abcdef1",
    "SubnetId": "subnet-b3983dc4",
    "LifeCycleState": "available",
    "IpAddress": "10.0.1.24",
    "NetworkInterfaceId": "eni-3851ec4e",
    "AvailabilityZoneId": "use2-az1",
    "AvailabilityZoneName": "us-east-2a",
    "VpcId": "vpc-3c39ef57"
}
```

예: 특정 IPv6 주소에서 탑재 대상 생성

다음 명령은 탑재 대상에 사용할 파일 시스템, 서브넷, 보안 그룹 및 IPv6 주소를 지정합니다. 대상은 지정된 서브넷의 지정된 IPv6 주소에 생성됩니다.

```
$ aws efs create-mount-target \
--file-system-id file-system-id \
--subnet-id  subnet-id \
--security-group ID-of-the-security-group-created-for-mount-target \
--ip-address-type IP-address-type \
--ipv6-address IPv6-address \
--region aws-region \
```

다음 예제는 샘플 데이터가 있는 명령을 나타낸 것입니다.

```
$ aws efs create-mount-target \
--file-system-id fs-0123456789abcdef1 \
--subnet-id  subnet-b3983dc4 \
--security-group sg-01234567 \
--ip-address-type IPV6_ONLY \
--ipv6-address 2001:0db8:85a3:0000:0000:8a2e:0370:7334 \
--region us-east-2 \
```

탑재 대상을 만든 후 Amazon EFS에서는 다음 예에서처럼 탑재 대상 설명을 JSON으로 반환합니다.

```
{
    "OwnerID": "111122223333"
    "MountTargetId": "fsmt-f9a14450",
    "FileSystemId": "fs-0123456789abcdef1",
```

```
    "SubnetId": "subnet-b3983dc4",
    "LifeCycleState": "available",
    "Ipv6Address": "2001:0db8:85a3:0000:0000:8a2e:0370:7334",
    "NetworkInterfaceId": "eni-3851ec4e",
    "AvailabilityZoneId": "use2-az1",
    "AvailabilityZoneName": "us-east-2a",
    "VpcId": "vpc-3c39ef57"
}
```

예: 듀얼 스택 서브넷에서 사용 가능한 IPv4 주소에 탑재 대상 생성

명령은 탑재 대상의 파일 시스템, 서브넷, 보안 그룹, 듀얼 스택 IP 주소 유형 및 IPv6 주소를 지정합니다. 대상은 사용 가능한 IPv4 주소와 듀얼 스택 서브넷의 지정된 IPv6 주소에서 생성됩니다.

```
$ aws efs create-mount-target \
--file-system-id file-system-id \
--subnet-id  subnet-id \
--security-group ID-of-the-security-group-created-for-mount-target \
--ip-address-type IP-address-type
--ipv6-address IPv6-address \
--region aws-region \
```

다음 예제는 샘플 데이터가 있는 명령을 나타낸 것입니다.

```
$ aws efs create-mount-target \
--file-system-id fs-0123456789abcdef1 \
--subnet-id  subnet-b3983dc4 \
--security-group sg-01234567 \
--ip-address-type DUAL_STACK \
--ipv6-address 2001:0db8:85a3:0000:0000:8a2e:0370:7334 \
--region us-east-2 \
```

탑재 대상을 만든 후 Amazon EFS에서는 다음 예에서처럼 탑재 대상 설명을 JSON으로 반환합니다.

```
{
    "OwnerID": "111122223333"
    "MountTargetId": "fsmt-f9a14450",
    "FileSystemId": "fs-0123456789abcdef1",
    "SubnetId": "subnet-b3983dc4",
    "LifeCycleState": "available",
    "IpAddress": "10.0.1.24",
    "Ipv6Address": "2001:0db8:85a3:0000:0000:8a2e:0370:7334",
```

```
    "NetworkInterfaceId": "eni-3851ec4e",
    "AvailabilityZoneId": "use2-az1",
    "AvailabilityZoneName": "us-east-2a",
    "VpcId": "vpc-3c39ef57"
}
```

탑재 대상 삭제

탑재 대상을 삭제하면 파일 시스템의 모든 탑재가 강제로 해제되어 이러한 탑재를 사용하는 인스턴스 또는 애플리케이션이 중단될 수 있습니다. 애플리케이션 중단을 피하기 위해서는 탑재 대상을 삭제하기 전에 애플리케이션을 중지하고 파일 시스템의 탑재를 해제합니다. 자세한 내용은 파일 시스템 탑재 해제 단원을 참조하십시오.

를 사용하거나 SDK를 사용하여 AWS Management Console AWS CLI프로그래밍 방식으로 파일 시스템의 탑재 대상을 삭제할 수 있습니다. AWS SDKs

콘솔 사용

기존 EFS 파일 시스템에 탑재 대상을 삭제하려면 다음 절차를 따르세요.

EFS 파일 시스템에서 탑재 대상을 삭제하는 방법

1. 파일 시스템을 탑재 해제합니다. 지침은 파일 시스템 탑재 해제 섹션을 참조하세요.
2. Amazon Elastic File System 콘솔(https://console.aws.amazon.com/efs/)을 엽니다.
3. 왼쪽 탐색 창에서 파일 시스템을 선택한 다음 탑재 대상을 삭제하려는 파일 시스템을 선택합니다.
4. 네트워크를 선택한 다음 관리를 선택하여 파일 시스템의 탑재 대상을 표시합니다.
5. 삭제하려는 각 탑재 대상에 대해 제거를 선택합니다.
6. 저장을 선택합니다.

사용 AWS CLI

기존 탑재 대상을 삭제하려면 다음과 같이 `delete-mount-target` AWS CLI 명령(해당 작업: DeleteMountTarget)을 사용합니다.

> **Note**
>
> 탑재 대상을 삭제하기 전에 파일 시스템의 탑재를 해제합니다.

```
$ aws efs delete-mount-target \
--mount-target-id mount-target-ID-to-delete \
--region aws-region-where-mount-target-exists
```

다음은 샘플 데이터가 있는 예제입니다.

```
$ aws efs delete-mount-target \
--mount-target-id fsmt-5751852e \
--region us-east-2 \
```

탑재 대상 VPC 변경

EFS 파일 시스템은 한 번에 Amazon VPC 서비스를 기반으로 하는 VPC 하나에서 한 개만 사용할 수 있습니다. 즉, 파일 시스템의 VPC에서 탑재 대상을 만들고 이러한 탑재 대상을 사용하여 파일 시스템에 액세스할 수 있습니다.

이러한 대상에서 EFS 파일 시스템을 탑재할 수 있습니다.

- 동일한 VPC에 있는 EC2 인스턴스
- VPC 피어링으로 연결된 VPC의 EC2 인스턴스
- 를 사용한 온프레미스 서버 AWS Direct Connect
- Amazon VPC를 사용하여 AWS 가상 프라이빗 네트워크(VPN)를 통한 온프레미스 서버

VPC 피어링 연결은 두 VPC 간에 트래픽을 라우팅할 수 있도록 하기 위한 두 VPC 사이의 네트워킹 연결입니다. 연결은 프라이빗 인터넷 프로토콜 버전 4(IPv4) 또는 버전 6(IPv6) 주소를 사용할 수 있습니다. Amazon EFS에서 VPC 피어링을 사용하는 방법에 대한 자세한 내용은 다른 AWS 계정 또는 VPC에서 EFS 파일 시스템 탑재 섹션을 참조하세요.

파일 시스템의 VPC를 변경하는 방법

다음은 EFS 파일 시스템의 네트워크 구성에 대한 VPC를 변경하기 위해 수행할 단계입니다.

1. 파일 시스템에 할당된 각 탑재 대상을 삭제합니다. 지침은 탑재 대상 삭제 섹션을 참조하세요.
2. 각 탑재 대상의 탑재 대상 상태가 삭제됨이면 새 VPC를 할당하고 파일 시스템에 대한 새 탑재 대상을 생성합니다. 지침은 탑재 대상 생성 섹션을 참조하세요.

탑재 대상 보안 그룹 변경

보안 그룹은 인바운드 및 아웃바운드 액세스를 정의합니다. 탑재 대상과 연결된 보안 그룹을 변경하는 경우 필수 인바운드 및 아웃바운드 액세스를 승인해야 합니다. 그렇게 하면 EC2 인스턴스가 파일 시스템과 통신할 수 있습니다. 보안 그룹에 대한 자세한 내용은 VPC 보안 그룹 사용를(을) 참조하세요.

AWS Management Console, 또는 SDK를 사용하여 프로그래밍 방식으로 파일 시스템의 탑재 대상에 대한 보안 그룹을 추가 AWS CLI하거나 제거할 수 AWS SDKs.

콘솔 사용

탑재 대상의 보안 그룹을 수정하는 방법

기존 EFS 파일 시스템에 탑재 대상 보안 그룹을 추가하거나 제거하려면 다음 절차를 사용합니다.

1. Amazon Elastic File System 콘솔(https://console.aws.amazon.com/efs/)을 엽니다.
2. 왼쪽 탐색 창에서 파일 시스템을 선택한 다음 탑재 대상을 관리할 파일 시스템을 선택합니다.
3. 네트워크를 선택한 다음 관리를 선택하여 파일 시스템의 탑재 대상을 표시합니다.
4. 탑재 대상에서 보안 그룹을 제거하려면 보안 그룹 ID 옆의 X를 선택합니다.
5. 탑재 대상에 보안 그룹을 추가하려면 보안 그룹 목록에서 보안을 선택합니다.
6. 저장을 선택합니다.

사용 AWS CLI

탑재 대상에 적용되는 보안 그룹을 수정하려면 다음과 같이 `modify-mount-target-security-group` AWS CLI 명령(해당 작업은)ModifyMountTargetSecurityGroups을 사용하여 기존 보안 그룹을 대체합니다.

```
$ aws efs modify-mount-target-security-groups \
--mount-target-id mount-target-ID-whose-configuration-to-update \
--security-groups  security-group-ids-separated-by-space \
--region aws-region-where-mount-target-exists \
--profile adminuser
```

다음은 샘플 데이터가 있는 예제입니다.

```
$ aws efs modify-mount-target-security-groups \
--mount-target-id fsmt-5751852e \
--security-groups  sg-1004395a sg-1114433a \
```

```
--region us-east-2
```

파일 시스템 처리량 관리

탄력적 처리량은 Amazon EFS 콘솔의 기본 처리량 모드이며 대부분의 사용 사례에 권장됩니다. 탄력적 처리량을 사용하면 워크로드 활동의 요구 사항에 맞게 성능이 자동으로 확장 또는 축소됩니다. 하지만 워크로드의 특정 액세스 패턴(처리량, 지연 시간, 스토리지 요구 사항 포함)을 알고 있다면 처리량 모드를 변경할 수 있습니다.

> **Note**
>
> 탄력적 처리량은 처리량에 따라 탄력적으로 확장되도록 설계되었지만 운영 모범 사례의 일환으로 CloudWatch(MeteredIOBytes) 및 사용량 알림을 사용하여 지표를 모니터링하여 적절한 거버넌스를 구현하는 것이 좋습니다. 이를 통해 최적의 리소스 사용률을 유지하고 계획된 운영 파라미터를 유지할 수 있습니다. 자세한 내용은 Amazon CloudWatch를 사용한 지표 모니터링 단원을 참조하십시오.

선택할 수 있는 다른 처리량 모드는 다음과 같습니다.

- 프로비저닝 처리량 - 파일 시스템의 크기나 버스트 크레딧 잔고에 관계없이 파일 시스템이 제공할 수 있는 처리량 수준을 지정합니다.
- 버스팅 처리량 - 처리량은 파일 시스템의 스토리지 용량에 따라 조정되며 하루 최대 12시간 동안 더 높은 수준으로 버스팅을 지원합니다.

Amazon EFS 처리량 모드에 대한 자세한 내용은 처리량 모드 섹션을 참조하세요.

> **Note**
>
> 파일 시스템을 사용할 수 있게 된 후에 처리량 모드와 프로비저닝된 처리량을 변경할 수 있습니다. 처리량 모드를 변경해도 애플리케이션 가동 중지는 발생하지 않습니다. 하지만 파일 시스템을 프로비저닝 처리량으로 변경하거나 프로비저닝 처리량을 늘릴 때마다 최소 24시간을 기다려야 처리량을 다시 변경하거나 프로비저닝된 양을 줄일 수 있습니다.

Amazon EFS 콘솔, AWS Command Line Interface (AWS CLI) 및 Amazon EFS API를 사용하여 파일 시스템 처리량 모드를 관리할 수 있습니다.

콘솔 사용

파일 시스템 처리량 관리 방법

1. Amazon Elastic File System 콘솔(https://console.aws.amazon.com/efs/)을 엽니다.
2. 왼쪽 탐색 창에서 파일 시스템을 선택하여 계정의 EFS 파일 시스템 목록을 표시합니다.
3. 처리량 모드를 변경하려는 파일 시스템을 선택합니다.
4. 파일 시스템 세부 정보 페이지의 일반 섹션에서 편집을 선택합니다. 편집 페이지가 표시됩니다.
5. 처리량 모드 설정을 수정합니다.

   - 탄력적 처리량 또는 프로비저닝 처리량을 사용하려면 향상됨을 선택한 다음 탄력적 또는 프로비저닝을 선택합니다.

     프로비저닝을 선택한 경우 프로비저닝 처리량(MiB/s)에 파일 시스템 요청에 따라 프로비저닝할 처리량을 입력합니다. 최대 읽기 처리량은 입력한 처리량의 3배로 표시됩니다. EFS 파일 시스템은 다른 요청의 1/3 속도로 읽기 요청을 측정합니다. 처리량을 입력하면 파일 시스템의 월별 예상 비용이 표시됩니다.

     > **Note**
     >
     > 파일 시스템을 사용할 수 있게 된 후에 처리량 모드와 프로비저닝된 처리량을 변경할 수 있습니다. 그러나 파일 시스템 처리량을 프로비저닝된 처리량으로 변경하거나 프로비저닝된 처리량을 늘릴 때마다 최소 24시간을 기다려야 처리량 모드를 다시 변경하거나 프로비저닝된 처리량을 줄일 수 있습니다.

   - 버스팅 처리량을 사용하려면 버스팅을 선택합니다.

   성능 요구 사항에 맞는 올바른 처리량 모드를 선택하는 방법에 대한 자세한 내용은 처리량 모드을 참조하세요.
6. 변경 사항을 적용하려면 변경 사항 저장을 선택합니다.

사용 AWS CLI

update-file-system CLI 명령 또는 UpdateFileSystem API 작업을 사용하여 파일 시스템의 처리량 모드를 변경합니다.

스토리지 수명 주기 관리

파일 시스템이 수명 주기 내내 비용 효율적인 스토리지를 확보하도록 관리할 수 있습니다. 수명 주기 관리를 사용하면 파일 시스템의 수명 주기 구성에 따라 스토리지 클래스 간에 데이터를 자동으로 전환할 수 있습니다. 수명 주기 구성은 파일 시스템에 대해 설정한 세 가지 수명 주기 정책으로 구성됩니다.

수명 주기 정책은 파일을 EFS Infrequent Access(IA) 및 EFS Archive 스토리지 클래스로 전환하거나 해당 클래스에서 다른 클래스로 전환하는 시점을 수명 주기 관리에 지시합니다. 전환 시간은 Standard 스토리지 클래스에서 파일에 마지막으로 액세스한 시간을 기준으로 합니다. Standard 스토리지 클래스에서 마지막으로 액세스한 시간을 확인하기 위해 내부 타이머는 파일이 마지막으로 액세스된 시간을 추적합니다(공개적으로 볼 수 있는 POSIX 파일 시스템 속성은 제외). Standard의 파일에 액세스할 때마다 수명 주기 관리 타이머가 재설정됩니다.

수명 주기 정책은 EFS 파일 시스템 전체에 적용됩니다.

EFS 수명 주기 정책은 다음과 같습니다.

- IA로 전환 - 분기별로 몇 번만 액세스하는 데이터에 대해 비용을 최적화하는 Infrequent Access(IA) 스토리지 클래스로 파일을 이동할 시기를 수명 주기 관리에 지시합니다. 기본적으로 Standard 스토리지에서 30일 동안 액세스하지 않은 파일은 IA로 전환됩니다.
- Archive로 전환 - 매년 몇 번 이하로만 액세스되는 데이터에 대해 비용 최적화된 Archive 스토리지 클래스로 파일을 이동할 시점을 수명 주기 관리에 지시합니다. 기본적으로 Standard 스토리지에서 90일 동안 액세스되지 않은 파일은 Archive 스토리지 클래스로 전환됩니다.
- Standard로 전환 - 파일의 수명 주기 관리에 대해, 해당 파일이 IA 또는 Archive 스토리지 클래스에서 액세스될 때 해당 파일을 IA 또는 Archive 스토리지 클래스에서 Standard 스토리지 클래스로 전환하거나 다시 Standard 스토리지 클래스로 되돌릴지 여부를 지시합니다. 기본적으로 파일은 Standard 스토리지 클래스로 다시 이동되지 않고 액세스 시 IA 또는 Archive 스토리지 클래스에 남아 있습니다.

  가장 빠른 지연 시간을 요구하는, 성능에 민감한 사용 사례(예: 대용량의 작은 파일을 처리하는 애플리케이션)의 경우 최초 액세스 시 파일을 Standard 스토리지로 전환하도록 선택하십시오.

파일 시스템의 수명 주기 정책 구성에 대한 자세한 내용은 수명 주기 정책 구성 섹션을 참조하십시오.

수명 주기 관리를 위한 파일 시스템 작업

수명 주기 관리에 대한 파일 시스템 작업은 EFS 파일 시스템 워크로드에 대한 작업보다 우선 순위가 낮습니다. 파일을 IA 및 Archive 스토리지로 전환하거나 탈퇴하는 데 필요한 시간은 파일 크기와 파일

시스템 워크로드에 따라 달라집니다. 예를 들어, 수백만 개의 작은 파일을 전환하면 총 스토리지 크기가 동일한 더 큰 파일을 더 적게 전환하는 것보다 더 오래 걸릴 수 있습니다.

파일 이름, 소유권 정보 및 파일 시스템 디렉터리 구조가 포함된 파일 메타데이터는 일관된 메타데이터 성능 보장을 위해 항상 Standard 스토리지에 저장됩니다.

IA 또는 Archive 스토리지에 있는 파일 시스템에 대한 메타데이터 작업(예: 디렉터리 콘텐츠 나열)은 파일 액세스로 간주되지 않습니다. 파일의 내용을 IA 또는 Archive 스토리지 클래스로 전환하는 동안 파일이 Standard 스토리지 클래스에 저장되고 해당 스토리지 요금이 청구됩니다.

파일 시스템의 IA 또는 Archive 스토리지 클래스에 있는 파일에 대한 모든 쓰기 작업은 먼저 Standard 스토리지 클래스에 기록된 후 24시간 후에 해당 스토리지 클래스로 전환될 수 있습니다.

수명 주기 정책 구성

를 사용하여 권장 설정이 있는 EFS 파일 시스템을 생성하면 AWS Management Console파일 시스템이 다음과 같은 기본 수명 주기 구성으로 자동으로 구성됩니다.

- IA로의 전환은 마지막 액세스 이후 30일로 설정됩니다.
- Archive로 전환은 마지막 액세스 이후 90일로 설정되어 있습니다.
- Standard로 전환은 없음으로 설정되어 있습니다.

를 사용하여 사용자 지정 설정으로 파일 시스템을 생성할 때 AWS Management Console 또는를 사용하여 파일 시스템을 생성할 때 기본 수명 주기 정책을 변경할 수 있습니다 AWS CLI. 또는 다음 절차에 설명된 대로 파일 시스템을 생성한 후 정책을 변경할 수 있습니다.

콘솔 사용

AWS Management Console 를 사용하여 기존 파일 시스템에 대한 수명 주기 정책을 설정할 수 있습니다.

1. 에 로그인 AWS Management Console 하고 https://console.aws.amazon.com/efs/ Amazon EFS 콘솔을 엽니다.
2. 파일 시스템을 선택하면 계정의 파일 시스템 목록이 표시됩니다.
3. 수명 주기 정책을 수정하려는 파일 시스템을 선택합니다.
4. 파일 시스템 세부 정보 페이지의 일반 섹션에서 편집을 선택합니다. 편집 페이지가 표시됩니다.
5. 수명 주기 관리에서 수명 주기 정책을 구성합니다.

- IA로 전환을 사용 가능한 옵션 중 하나로 설정합니다. IA 스토리지로의 파일 이동을 중지하려면 없음을 선택합니다.
- Archive로 전환을 사용 가능한 옵션 중 하나로 설정합니다. Archive 스토리지로의 파일 이동을 중지하려면 없음을 선택합니다.
- 메타데이터가 아닌 작업을 위해 액세스하는 경우 IA 스토리지에 있는 파일을 Standard 스토리지로 옮기려면 Standard로 전환에서 첫 액세스 시로 설정하십시오.

  처음 액세스할 때 IA 또는 Archive에서 Standard 스토리지로 파일 이동을 중지하려면 없음을 선택합니다.

6. 변경 사항을 저장하려면 변경 사항 저장을 선택합니다.

사용 AWS CLI

AWS CLI 를 사용하여 파일 시스템의 수명 주기 정책을 설정하거나 수정할 수 있습니다.

- `put-lifecycle-configuration` AWS CLI 수명 주기 관리를 관리하는 파일 시스템의 파일 시스템 ID를 지정하여 명령 또는 PutLifecycleConfiguration API 명령을 실행합니다.

```
$  aws efs put-lifecycle-configuration \
--file-system-id File-System-ID \
--lifecycle-policies "[{\"TransitionToIA\":\"AFTER_60_DAYS\"},
{\"TransitionToPrimaryStorageClass\":\"AFTER_1_ACCESS\"},{\"TransitionToArchive\":
\"AFTER_90_DAYS\"}]" \
--region us-west-2 \
--profile adminuser
```

응답은 다음과 같습니다.

```
{
    "LifecyclePolicies": [
        {
            "TransitionToIA": "AFTER_60_DAYS"
        },
        {
            "TransitionToPrimaryStorageClass": "AFTER_1_ACCESS"
        },
        {
            "TransitionToArchive": "AFTER_90_DAYS"
```

```
        }
    ]
}
```

기존 파일 시스템에서 수명 주기 관리 중지(CLI)

- `put-lifecycle-configuration` 명령을 실행하여 수명 주기 관리를 중지하는 파일 시스템의 파일 시스템 ID를 지정합니다. `--lifecycle-policies` 속성을 비워 두세요.

```
$  aws efs put-lifecycle-configuration \
--file-system-id File-System-ID \
--lifecycle-policies \
--region us-west-2 \
--profile adminuser
```

응답은 다음과 같습니다.

```
{
    "LifecyclePolicies": []
}
```

Amazon EFS 모니터링

모니터링은 Amazon EFS 및 AWS 솔루션의 안정성, 가용성 및 성능을 유지하는 데 중요한 부분입니다. 다중 지점 장애가 발생할 경우 보다 쉽게 디버깅할 수 있도록 AWS 솔루션의 모든 부분으로부터 모니터링 데이터를 수집하는 것이 좋습니다. 하지만 Amazon EFS 모니터링을 시작하기 전에 다음 질문에 대한 답변을 포함하는 모니터링 계획을 작성하세요.

- 모니터링의 목표
- 모니터링할 리소스
- 이러한 리소스를 모니터링하는 빈도
- 사용할 모니터링 도구
- 모니터링 작업을 수행할 사람
- 문제 발생 시 알려야 할 대상

다음 단계에서는 다양한 시간과 다양한 부하 조건에서 성능을 측정하여 환경에서 일반 Amazon EFS 성능의 기준선을 설정합니다. Amazon EFS를 모니터링할 때 과거 모니터링 데이터를 저장하는 것이 좋습니다. 이 저장된 데이터는 현재 성능 데이터와 비교하고, 일반 성능 패턴과 성능 이상을 식별하고, 문제 해결 방법을 제안하는 기준이 됩니다.

예를 들어 Amazon EFS에서 네트워크 처리량, 읽기, 쓰기 및 메타데이터 작업에 대한 I/O, 클라이언트 연결, 파일 시스템에 대한 버스트 크레딧 밸런스 등을 모니터링할 수 있습니다. 성능이 정해진 기준에 못 미치는 경우, 워크로드에 맞게 파일 시스템을 최적화하기 위해 파일 시스템의 크기 또는 연결된 클라이언트 수를 변경해야 할 수 있습니다.

기준선을 설정하려면 최소한 다음 항목을 모니터링해야 합니다.

- 파일 시스템의 네트워크 처리량
- 파일 시스템에 연결된 클라이언트 수
- 각 파일 시스템 작업(데이터 읽기, 쓰기 및 메타데이터 작업 포함)에 필요한 바이트 수

주제

- 모니터링 도구
- Amazon EFS에서 파일 시스템 및 객체 크기를 보고하는 방법
- 스토리지 클래스 크기 보기
- Amazon CloudWatch를 사용한 지표 모니터링

- 를 사용하여 Amazon EFS API 호출 로깅 AWS CloudTrail

모니터링 도구

AWS 는 Amazon EFS를 모니터링하는 데 사용할 수 있는 다양한 도구를 제공합니다. 이들 도구 중에는 모니터링을 자동으로 수행하도록 구성할 수 있는 도구도 있지만, 수동 작업이 필요한 도구도 있습니다. 모니터링 작업은 최대한 자동화하는 것이 좋습니다.

자동 모니터링 도구

다음과 같은 자동 모니터링 도구를 사용하여 Amazon EFS를 관찰하고 문제 발생 시 보고할 수 있습니다.

- Amazon CloudWatch 경보 – 지정한 기간 동안 단일 지표를 감시하고, 여러 기간에 대해 지정된 임곗값과 관련하여 지표 값을 기준으로 하나 이상의 작업을 수행합니다. 이 태스크는 Amazon Simple Notification Service(SNS) 주제 또는 Amazon EC2 Auto Scaling 정책에 전송되는 알림입니다. CloudWatch 경보는 특정 상태에 있다는 이유만으로는 작업을 간접 호출하지 않습니다. 상태가 변경되고 지정한 기간 동안 유지되어야 합니다. 자세한 내용은 Amazon CloudWatch를 사용한 지표 모니터링 단원을 참조하십시오.
- Amazon CloudWatch Logs – AWS CloudTrail 또는 기타 소스의 로그 파일을 모니터링, 저장 및 액세스합니다. 자세한 내용은 Amazon CloudWatch 사용 설명서에서 Amazon CloudWatch Logs란 무엇입니까?를 참조하세요.
- Amazon CloudWatch Events – 이벤트를 일치시키고 하나 이상의 대상 함수 또는 스트림으로 라우팅하여 값을 변경하거나 상태 정보를 캡처하거나 수정 작업을 수행합니다. 자세한 내용은 Amazon CloudWatch 사용 설명서의 Amazon CloudWatch Events란 무엇인가요를 참조하세요.
- AWS CloudTrail 로그 모니터링 - 계정 간에 로그 파일을 공유하고, CloudTrail 로그 파일을 CloudWatch Logs로 전송하여 실시간으로 모니터링하고, Java로 로그 처리 애플리케이션을 작성하고, CloudTrail에서 전송한 후 로그 파일이 변경되지 않았는지 확인합니다. 자세한 내용은 AWS CloudTrail 사용 설명서의 CloudTrail 로그 파일 작업을 참조하십시오.

수동 모니터링 도구

Amazon EFS 모니터링의 또 한 가지 중요한 부분은 Amazon CloudWatch 경보에 포함되지 않는 항목을 수동으로 모니터링해야 한다는 점입니다. Amazon EFS, CloudWatch 및 기타 AWS Management Console 대시보드는 AWS 환경 상태를 at-a-glance 볼 수 있습니다. 파일 시스템의 로그 파일도 확인하는 것이 좋습니다.

- Amazon EFS 콘솔에서 파일 시스템에 대해 다음과 같은 항목을 찾을 수 있습니다.
  - 현재 측정된 크기
  - 탑재 대상 수
  - 수명 주기 상태
- CloudWatch 홈 페이지에 표시되는 항목은 다음과 같습니다.
  - 현재 경보 및 상태
  - 경보 및 리소스 그래프
  - 서비스 상태

  또한 CloudWatch를 사용하여 다음을 수행할 수 있습니다.
  - 사용자 지정 대시보드를 만들어 사용하는 서비스 모니터링
  - 지표 데이터를 그래프로 작성하여 문제를 해결하고 추세 파악
  - 모든 AWS 리소스 지표를 검색하고 검색합니다.
  - 문제에 대해 알려주는 경보 생성 및 편집

Amazon EFS에서 파일 시스템 및 객체 크기를 보고하는 방법

다음 단원에서는 Amazon EFS가 파일 시스템 크기, 파일 시스템 내 객체 크기 및 파일 시스템 처리량을 보고하는 방법에 대해 설명합니다.

EFS 파일 시스템 객체 측정

EFS 파일 시스템에서 볼 수 있는 객체는 일반 파일, 디렉터리, 심볼 링크 및 특수 파일(FIFO 및 소켓)일 수 있습니다. inode에 대한 메타데이터 2KiB(키비바이트) 및 데이터 4KiB 단위로 이러한 객체를 각각 측정합니다. 다음 목록은 파일 시스템 객체의 여러 유형에 대해 측정된 데이터 크기를 설명합니다.

- 일반 파일 - 일반 파일의 측정된 데이터 크기는 다음 4KiB 증분으로 반올림한 파일의 논리적 크기입니다. 단, 스파스 파일의 경우 이보다 작을 수 있습니다.

  스파스 파일은 논리적 크기에 도달하기 전에는 파일의 어떤 위치에도 데이터를 쓰지 않는 파일입니다. 스파스 파일의 경우 사용되는 실제 스토리지가 다음 4KiB로 반올림된 논리적 크기보다 작을 수 있습니다. 이 경우 Amazon EFS에서는 사용된 실제 스토리지를 측정된 데이터 크기로 보고합니다.
- 디렉터리 - 디렉터리의 측정된 데이터 크기는 디렉터리 항목에 사용된 실제 스토리지와 해당 항목을 보관하는 데이터 구조를 다음 4KiB 단위로 반올림한 값입니다. 이 메타데이터 데이터 크기에는 파일 데이터가 사용하는 실제 스토리지가 포함되지 않습니다.

- 심볼 링크 및 특수 파일 - 이러한 객체의 측정된 데이터 크기는 항상 4KiB입니다.

Amazon EFS가 NFSv4.1 space_used 속성을 통해 객체에 사용되는 공간을 보고하는 경우 여기에는 객체의 메타데이터 크기가 아니라 객체의 현재 측정된 데이터 크기가 포함됩니다. 파일의 디스크 사용량 측정에는 du 및 stat라는 두 가지 유틸리티를 사용할 수 있습니다. 다음은 빈 파일에서 -k 옵션과 함께 du 유틸리티를 사용하여 출력을 KB 단위로 반환하는 방법을 보여주는 예시입니다.

```
$ du -k file
4       file
```

다음은 빈 파일에서 stat 유틸리티를 사용하여 파일의 디스크 사용량을 반환하는 방법을 보여주는 예시입니다.

```
$ /usr/bin/stat --format="%b*%B" file | bc
4096
```

디렉터리의 크기를 측정하려면 stat 유틸리티를 사용합니다. Blocks 값을 찾은 다음 이 값에 블록 크기를 곱합니다. 다음은 빈 디렉터리에서 stat 유틸리티를 사용하는 방법을 보여주는 예입니다.

```
$ /usr/bin/stat --format="%b*%B" . | bc
4096
```

EFS 파일 시스템의 측정된 크기

EFS 파일 시스템의 측정된 크기에는 모든 EFS 스토리지 클래스에 있는 모든 현재 오브젝트의 크기 합계가 포함됩니다. 각 객체의 크기는 측정 시간(예: 오전 8시~오전 9시) 동안 객체의 크기를 나타내는 대표 표본에서 계산합니다.

빈 파일이 파일 시스템의 측정된 크기 중 6KiB(2KiB의 메타데이터 + 4KiB의 데이터)를 차지합니다. 생성 시 파일 시스템에는 빈 루트 디렉터리가 하나뿐이므로 측정된 크기는 6KiB입니다.

특정 파일 시스템의 측정된 크기는 해당 시간 동안 해당 파일 시스템에 대해 소유자 계정에 청구되는 사용량을 정의합니다.

> **Note**
>
> 계산된 측정 크기는 해당 시간 중 특성 시점에서 파일 시스템의 일관된 스냅샷을 나타내지 않습니다. 매 시간 또는 가능한 경우 해당 시간 이전의 다양한 시점에서 파일 시스템 내에 존재하는 객체의 크기를 나타냅니다. 이러한 크기가 합산되어 해당 시간에 대해 측정된 파일 시스템

의 크기를 결정합니다. 따라서 파일 시스템의 측정된 크기는 파일 시스템에 대한 쓰기가 없는 경우 저장된 객체의 측정된 크기와 일치합니다.

EFS 파일 시스템에 대해 측정된 크기는 다음과 같은 방법으로 확인할 수 있습니다.

- describe-file-systems AWS CLI 명령과 DescribeFileSystem API 작업을 사용하여 응답에는 다음이 포함됩니다.

```
"SizeInBytes":{
            "Timestamp": 1403301078,
            "Value": 29313744866,
            "ValueInIA": 675432,
            "ValueInStandard": 29312741784
            "ValueInArchive": 327650
        }
```

  `ValueInStandard`의 측정된 크기는 버스팅 처리량을 사용하는 파일 시스템의 I/O 처리량 기준 및 버스트 속도를 결정하는 데에도 사용됩니다.
- 각 스토리지 클래스의 총 측정 데이터 크기를 표시하는 `StorageBytes` CloudWatch 지표를 확인합니다. `StorageBytes` 지표에 대한 자세한 내용은 Amazon EFS에 대한 CloudWatch 지표를 참고하세요.
- Linux의 `df` 명령은 EC2 인스턴스의 터미널 프롬프트에서 실행합니다.

  파일 시스템 루트에서 이 du 명령을 스토리지 측정 목적으로 사용하지 마십시오. 응답에는 파일 시스템 측정에 사용된 전체 집합 데이터가 반영되지 않기 때문입니다.

> **Note**
>
> `ValueInStandard`의 측정된 크기는 I/O 처리량 기준 및 버스트 속도를 확인하는 데에도 사용됩니다. 자세한 내용은 버스팅 처리량 단원을 참조하십시오.

Infrequent Access 및 Archive 스토리지 클래스 측정

EFS Infrequent Access(IA) 및 Archive 스토리지 클래스는 4KiB 단위로 측정되며 파일당 최소 청구 요금은 128KiB입니다. IA 및 Archive 파일 메타데이터(파일당 2KiB)는 항상 Standard 스토리지 클래스에 저장되고 측정됩니다. 128KiB 미만의 파일에 대한 지원은 2023년 11월 26일 오후 12시(PT) 당일 또는

그 이후에 업데이트된 수명 주기 정책에만 사용할 수 있습니다. IA 및 Archive 스토리지에 대한 데이터 액세스는 128MiB 단위로 측정됩니다.

`StorageBytes` CloudWatch 지표를 사용하여 각 스토리지 클래스에서 측정된 데이터 크기를 확인할 수 있습니다. 이 지표에는 IA 및 Archive 스토리지 클래스 내에서 작은 파일 반올림으로 소비되는 총 바이트 수도 표시됩니다. CloudWatch 지표를 확인하는 방법에 대한 자세한 내용은 Amazon EFS 용 CloudWatch 지표 액세스 단원을 참조하세요. `StorageBytes` 지표에 대한 자세한 내용은 Amazon EFS에 대한 CloudWatch 지표를 참고하세요.

처리량 측정

Amazon EFS는 다른 파일 시스템 I/O 작업의 1/3 속도로 읽기 요청 처리량을 측정합니다. 예를 들어 읽기 및 쓰기 처리량 모두에서 초당 30메비바이트(MiBps)를 구동하는 경우 읽기 부분은 유효 처리량에서 10MiBps로 계산되고 쓰기 부분은 30MiBps로 계산되며 측정된 총 처리량은 40MiBps입니다. 사용률에 맞게 조정된 이 총 처리량은 `MeteredIOBytes` CloudWatch 지표에 반영됩니다.

탄력적 처리량 측정

파일 시스템에 탄력적 처리량 모드를 사용하도록 설정하면 파일 시스템에서 읽거나 파일 시스템에 쓴 메타데이터 및 데이터 양에 대해서만 비용을 지불하면 됩니다. 탄력적 처리량 모드를 사용하는 EFS 파일 시스템은 메타데이터 읽기를 데이터 읽기 작업으로, 메타데이터 쓰기는 메타데이터 쓰기 작업으로 청구합니다. 메타데이터 작업은 처음 4KiB 이후 1KiB 단위로 측정됩니다. 데이터 작업은 처음 32KiB 이후 1KiB 단위로 측정됩니다.

> **Note**
>
> 탄력적 처리량은 처리량에 따라 탄력적으로 확장되도록 설계되었지만 운영 모범 사례의 일환으로 CloudWatch(MeteredIOBytes) 및 사용량 알림을 사용하여 지표를 모니터링하여 적절한 거버넌스를 구현하는 것이 좋습니다. 이를 통해 최적의 리소스 사용률을 유지하고 계획된 운영 파라미터를 유지할 수 있습니다. 자세한 내용은 Amazon CloudWatch를 사용한 지표 모니터링 단원을 참조하십시오.

프로비저닝 처리량 측정

프로비저닝된 처리량 모드를 사용하는 파일 시스템의 경우, 처리량이 활성화된 시간만큼만 비용을 지불합니다. Amazon EFS는 프로비저닝 처리량 모드가 활성화된 파일 시스템을 1시간에 한 번씩 측정합니다. 프로비저닝 처리량 모드가 1시간 미만으로 설정된 상황에서 측정할 경우 Amazon EFS는 밀리초 단위의 정밀도를 사용하여 시간 평균을 계산합니다.

스토리지 클래스 크기 보기

Amazon EFS 콘솔 AWS CLI, 또는 EFS API를 사용하여 파일 시스템의 각 스토리지 클래스에 저장되는 데이터의 양을 볼 수 있습니다.

콘솔 사용

파일 시스템 세부 정보 페이지의 측정된 크기 탭에는 파일 시스템의 현재 측정된 크기가 바이트의 2진수 배수(키비바이트, 메비바이트, 기가바이트, 테비바이트)로 표시됩니다. 지표는 15분마다 생성되며 시간 경과에 따른 파일 시스템의 측정된 크기를 확인할 수 있습니다. 측정된 크기에는 파일 시스템 스토리지 크기에 대한 다음 정보가 표시됩니다.

- 총 크기는 모든 스토리지 클래스를 포함하여 파일 시스템에 저장된 데이터의 크기(바이너리 바이트)입니다.
- Standard의 크기는 EFS Standard 스토리지 클래스에 저장된 데이터의 크기(바이너리 바이트)입니다.
- IA의 크기는 EFS Infrequent Access 스토리지 클래스에 저장된 데이터의 크기(바이너리 바이트)입니다. 128KiB보다 작은 파일은 128KiB로 반올림됩니다.
- Archive의 크기는 EFS Archive 스토리지 클래스에 저장된 데이터의 크기(바이너리 바이트)입니다. 128KiB보다 작은 파일은 128KiB로 반올림됩니다.

Amazon EFS 콘솔의 파일 시스템 세부 정보 페이지에 있는 모니터링 탭에서도 `Storage bytes` 지표를 볼 수 있습니다. 자세한 내용은 Amazon EFS용 CloudWatch 지표 액세스 단원을 참조하십시오.

사용 AWS CLI

AWS CLI 또는 EFS API를 사용하여 파일 시스템의 각 스토리지 클래스에 저장되는 데이터의 양을 볼 수 있습니다. `describe-file-systems`CLI 명령(해당 API 작업은 DescribeFileSystems)을 직접 호출하여 데이터 스토리지 세부 정보를 확인합니다.

```
$  aws efs describe-file-systems \
--region us-west-2 \
--profile adminuser
```

응답에서 `ValueInIA`는 파일 시스템의 Inrequent Access 스토리지 클래스에서 마지막으로 측정된 크기(바이트)를 표시합니다. `ValueInStandard`는 Standard 스토리지 클래스에서 마지막으로 측정한 크기(바이트)를 표시합니다. `ValueInArchive`는 Archive 스토리지 클래스에서 마지막으로 측정한 크기(바이트)를 표시합니다. 세 값의 합계는 `Value`에 표시된 전체 파일 시스템의 크기와 같습니다.

```
{
   "FileSystems":[
      {
         "OwnerId":"251839141158",
         "CreationToken":"MyFileSystem1",
         "FileSystemId":"fs-47a2c22e",
         "PerformanceMode" : "generalPurpose",
         "CreationTime": 1403301078,
         "LifeCycleState":"created",
         "NumberOfMountTargets":1,
         "SizeInBytes":{
            "Value": 29313746702,
            "ValueInIA": 675432,
            "ValueInStandard": 29312741784,
            "ValueInArchive":329486
        },
        "ThroughputMode": "elastic"
      }
   ]
}
```

디스크 사용량을 보기 및 측정하는 다른 방법은 EFS 파일 시스템 객체 측정 섹션을 참조하세요.

Amazon CloudWatch를 사용한 지표 모니터링

Amazon EFS에서 원시 데이터를 수집하여 읽기 가능하며 실시간에 가까운 지표로 처리하는 Amazon CloudWatch를 통해 파일 시스템을 모니터링할 수 있습니다. 이러한 통계는 15개월간 기록되므로 웹 애플리케이션이나 서비스가 어떻게 실행되고 있는지 전체적으로 더 잘 파악할 수 있습니다.

기본적으로 Amazon EFS 지표 데이터는 일부 개별 지표에 대해 별도로 명시되지 않는 한 1분 간격으로 CloudWatch로 자동 전송됩니다. Amazon EFS 콘솔에는 Amazon CloudWatch의 원시 데이터를 기초로 하는 일련의 그래프가 표시됩니다. 필요에 따라 콘솔의 그래프 대신에 Amazon CloudWatch에서 파일 시스템의 데이터를 얻는 것을 선호할 수도 있습니다.

Amazon CloudWatch에 대한 자세한 내용은 Amazon CloudWatch 사용 설명서의 Amazon CloudWatch란 무엇입니까?를 참조하세요.

아마존 EFS CloudWatch 지표는 원시 바이트로 보고됩니다. 바이트는 단위의 십진수나 이진수에 반올림되지 않습니다.

주제

- Amazon EFS에 대한 CloudWatch 지표
- Amazon EFS용 CloudWatch 지표 액세스
- Amazon EFS용 CloudWatch 지표 사용
- CloudWatch 지표와 함께 지표 수식 사용
- 탑재 시도 성공 또는 실패 모니터링
- Amazon EFS 모니터링을 위한 CloudWatch 경보 생성

Amazon EFS에 대한 CloudWatch 지표

Amazon EFS 지표에는 `EFS` 네임스페이스가 사용됩니다. `AWS/EFS` 네임스페이스에는 다음과 같은 지표가 포함됩니다. `TimeSinceLastSync`를 제외한 모든 지표는 단일 차원 `FileSystemId`에 대한 것입니다. 파일 시스템의 ID는 Amazon EFS 콘솔에서 찾아볼 수 있으며, 그 형식은 `fs-abcdef0123456789a`를 따릅니다.

`TimeSinceLastSync`

복제 구성에서 대상 파일 시스템에 마지막으로 성공적으로 동기화한 이후 경과된 시간을 표시합니다. `TimeSinceLastSync` 값이 성공적으로 복제되기 전에 발생한 소스 파일 시스템의 모든 데이터 변경은 성공적으로 복제되었습니다. `TimeSinceLastSync` 이후에 발생한 소스의 모든 변경 사항은 완전히 복제되지 않을 수 있습니다.

이 지표에는 다음 두 가지 차원이 사용됩니다.

- `FileSystemId` 차원 - 복제 구성에서 소스 파일 시스템의 ID입니다.
- `DestinationFileSystemId` 차원 - 복제 구성에서 대상 파일 시스템의 ID입니다.

단위: 초

유효한 통계: `Minimum`, `Maximum`, `Average`

`PercentIOLimit`

파일 시스템이 범용 성능 모드의 I/O 제한에 얼마나 가깝게 도달해있는지 나타냅니다.

단위: 백분율

유효한 통계: `Minimum`, `Maximum`, `Average`

BurstCreditBalance

파일 시스템의 버스트 크레딧 수 파일 시스템은 버스트 크레딧을 사용하여 일정 시간 파일 시스템의 기준 레벨을 넘는 처리량 레벨까지 버스팅할 수 있습니다.

`Minimum` 통계는 해당 기간을 통틀어서 가장 작은 버스트 크레딧 밸런스를 말합니다. `Maximum` 통계는 해당 기간을 통틀어서 가장 큰 버스트 크레딧 밸런스를 말합니다. 그리고 `Average` 통계는 해당 기간 중 평균 버스트 크레딧 밸런스를 말합니다.

단위: 바이트

유효한 통계: `Minimum`, `Maximum`, `Average`

PermittedThroughput

파일 시스템이 구동할 수 있는 최대 처리량입니다.

- 탄력적 처리량 모드를 사용하는 파일 시스템의 경우 이 값은 파일 시스템의 최대 쓰기 처리량을 반영합니다.
- 프로비저닝된 처리량 모드를 사용하는 파일 시스템의 경우 EFS Standard 스토리지 클래스에 저장된 데이터의 양으로 인해 파일 시스템이 프로비저닝된 것보다 많은 양을 처리할 수 있는 경우 이 지표는 프로비저닝된 용량 대신 더 높은 처리량을 반영합니다.
- 버스팅 처리량에서 파일 시스템의 경우, 이 값은 파일 시스템 크기 및 `BurstCreditBalance`에 따라 달라집니다.

`Minimum` 통계는 해당 기간을 통틀어서 최소 허용 처리량을 말합니다. `Maximum` 통계는 해당 기간을 통틀어서 최고 허용 처리량을 말합니다. 그리고 `Average` 통계는 해당 기간 중 평균 허용 처리량을 말합니다.

> **Note**
>
> 읽기 작업은 다른 작업의 3분의 1 비율로 측정됩니다.

단위: 바이트/초

유효한 통계: `Minimum`, `Maximum`, `Average`

MeteredIOBytes

데이터 읽기, 데이터 쓰기 및 메타데이터 작업을 포함하여 각 파일 시스템 작업에 대해 측정된 바이트 수로, 처리량 제한에 따라 읽기 작업이 할인됩니다.

MeteredIOBytes를 PermittedThroughput와 비교하는 CloudWatch 지표 수식 표현식을 생성할 수 있습니다. 이 값이 같으면 파일 시스템에 할당된 전체 처리량을 소비하게 됩니다. 이 경우 처리량을 높이기 위해 파일 시스템의 처리량 모드를 변경하는 것을 고려할 수 있습니다.

Sum 통계는 모든 파일 시스템 작업과 연결되어 측정된 총 바이트 수입니다. Minimum 통계는 해당 기간 중 가장 작은 작업의 크기입니다. Maximum 통계는 해당 기간 중 가장 큰 작업의 크기입니다. Average 통계는 해당 기간 중 작업의 평균 크기입니다. SampleCount 통계는 모든 작업의 수를 제공합니다.

단위:

- Minimum, Maximum, Average 및 Sum 통계일 때 바이트
- SampleCount 통계일 때 수

유효한 통계: Minimum, Maximum, Average, Sum, SampleCount

TotalIOBytes

읽기 할인 없이 Amazon EFS에서 처리하는 각 파일 시스템 작업의 실제 바이트 수입니다. 이 숫자에는 최소값이 포함되어 있으므로 애플리케이션에서 요청한 실제 금액과 다를 수 있습니다. 이 숫자는 PermittedThroughput에 표시된 숫자보다 높을 수도 있습니다.

데이터 작업은 32KiB에서 측정되고 다른 작업은 4KiB에서 측정됩니다. 최소값이 지나면 모든 작업이 KiB당 측정됩니다.

Sum 통계는 모든 파일 시스템 작업과 연결된 총 바이트 수입니다. Minimum 통계는 해당 기간 중 가장 작은 작업의 크기입니다. Maximum 통계는 해당 기간 중 가장 큰 작업의 크기입니다. Average 통계는 해당 기간 중 작업의 평균 크기입니다. SampleCount 통계는 모든 작업의 수를 제공합니다.

> **Note**
>
> 일정 기간 초당 평균 작업 수를 계산하려면 SampleCount 통계를 초 단위의 해당 기간으로 나누면 됩니다. 일정 기간 평균 처리량(바이트/초)을 계산하려면 Sum 통계를 초 단위의 해당 기간으로 나누면 됩니다.

단위:

- Minimum, Maximum, Average 및 Sum 통계일 때 바이트
- SampleCount 통계일 때 수

유효한 통계: Minimum, Maximum, Average, Sum, SampleCount

DataReadIOBytes

각 파일 시스템 읽기 작업의 실제 바이트 수입니다.

Sum 통계는 읽기 작업과 연결된 총 바이트 수입니다. Minimum 통계는 해당 기간 중 가장 작은 읽기 작업의 크기입니다. Maximum 통계는 해당 기간 중 가장 큰 읽기 작업의 크기입니다. Average 통계는 해당 기간 중 읽기 작업의 평균 크기입니다. 그리고 SampleCount 통계는 읽기 작업의 수를 제공합니다.

단위:

- Minimum, Maximum, Average 및 Sum 통계일 때 바이트
- SampleCount 통계일 때 수

유효한 통계: Minimum, Maximum, Average, Sum, SampleCount

DataWriteIOBytes

각 파일 시스템 쓰기 작업의 실제 바이트 수입니다.

Sum 통계는 쓰기 작업과 연결된 총 바이트 수입니다. Minimum 통계는 해당 기간 중 가장 작은 쓰기 작업의 크기입니다. Maximum 통계는 해당 기간 중 가장 큰 쓰기 작업의 크기입니다. Average 통계는 해당 기간 중 쓰기 작업의 평균 크기입니다. 그리고 SampleCount 통계는 쓰기 작업의 수를 제공합니다.

단위:

- Minimum, Maximum, Average 및 Sum 통계일 때 바이트
- SampleCount 통계일 때 수

유효한 통계: Minimum, Maximum, Average, Sum, SampleCount

MetadataIOBytes

각 메타데이터 작업의 실제 바이트 수입니다.

Sum 통계는 메타데이터 작업과 연결된 총 바이트 수입니다. Minimum 통계는 해당 기간 중 가장 작은 메타데이터 작업의 크기입니다. Maximum 통계는 해당 기간 중 가장 큰 메타데이터 작업의 크기입니다. Average 통계는 해당 기간 중 평균 메타데이터 작업의 크기입니다. 그리고 SampleCount 통계는 메타데이터 작업의 수를 제공합니다.

단위:

- Minimum, Maximum, Average 및 Sum 통계일 때 바이트
- SampleCount 통계일 때 수

유효한 통계: Minimum, Maximum, Average, Sum, SampleCount

MetadataReadIOBytes

각 메타데이터 읽기 작업의 실제 바이트 수입니다.

Sum 통계는 메타데이터 읽기 작업과 관련된 총 바이트 수입니다. Minimum 통계는 해당 기간 중 가장 작은 메타데이터 읽기 작업의 크기입니다. Maximum 통계는 해당 기간 중 가장 큰 메타데이터 읽기 작업의 크기입니다. Average 통계는 해당 기간 중 메타데이터 읽기 작업의 평균 크기입니다. SampleCount 통계는 메타데이터 읽기 작업의 수를 제공합니다.

단위:

- Minimum, Maximum, Average 및 Sum 통계일 때 바이트
- SampleCount 통계일 때 수

유효한 통계: Minimum, Maximum, Average, Sum, SampleCount

MetadataWriteIOBytes

각 메타데이터 쓰기 작업의 실제 바이트 수입니다.

Sum 통계는 메타데이터 쓰기 작업과 관련된 총 바이트 수입니다. Minimum 통계는 해당 기간 중 가장 작은 메타데이터 쓰기 작업의 크기입니다. Maximum 통계는 해당 기간 중 가장 큰 메타데이터 쓰기 작업의 크기입니다. Average 통계는 해당 기간 중 메타데이터 쓰기 작업의 평균 크기입니다. SampleCount 통계는 메타데이터 쓰기 작업의 수를 제공합니다.

단위:

- Minimum, Maximum, Average 및 Sum 통계일 때 바이트
- SampleCount 통계일 때 수

유효한 통계: Minimum, Maximum, Average, Sum, SampleCount

ClientConnections

파일 시스템에 연결된 클라이언트 수 표준 클라이언트를 사용할 경우 탑재된 Amazon EC2 인스턴스마다 하나씩 연결됩니다.

> **Note**
>
> 1분 이상의 기간에 대한 평균 `ClientConnections`를 계산하려면 `Sum` 통계를 기간(분 단위)으로 나눕니다.

단위: 클라이언트 연결 수

유효한 통계: `Sum`

`StorageBytes`

EFS 스토리지 클래스에 저장된 데이터 양을 포함한 파일 시스템의 크기(바이트)입니다. 이 지표는 15분마다 CloudWatch로 전송됩니다.

이 `StorageBytes` 지표에는 다음과 같은 차원이 있습니다.

- `Total`은 모든 스토리지 클래스에서 파일 시스템에 저장된 데이터의 측정된 크기(바이트)입니다. EFS Infrequent Access(IA) 및 EFS Archive 스토리지 클래스의 경우 128KiB보다 작은 파일은 128KiB 반올림됩니다.
- `Standard`는 EFS Standard 스토리지 클래스에 저장된 데이터의 측정된 크기(바이트)입니다.
- `IA`는 EFS Infrequent Access 스토리지 클래스에 저장된 데이터의 실제 크기(바이트)입니다.
- `IASizeOverhead`는 EFS Infrequent Access 스토리지 클래스의 실제 데이터 크기(`IA` 차원에 표시됨)와 스토리지 클래스의 측정된 크기 사이의 차이(바이트)로, 작은 파일은 128KiB로 반올림한 후 계산합니다.
- `Archive`는 EFS Archive 스토리지 클래스에 저장된 데이터의 실제 크기(바이트)입니다.
- `ArchiveSizeOverhead`는 EFS Archive 스토리지 클래스의 실제 데이터 크기(`Archive` 차원에 표시됨)와 스토리지 클래스의 측정된 크기 사이의 차이(바이트)로, 작은 파일은 128KiB로 반올림한 후 계산합니다.

단위: 바이트

유효한 통계: `Minimum`, `Maximum`, `Average`

> **Note**
>
> `StorageBytes`는 기본 1024 단위(키비바이트, 메비바이트, 기가바이트, 테비바이트)를 사용하여 Amazon EFS 콘솔 파일 시스템 지표 페이지에 표시됩니다.

Amazon EFS용 CloudWatch 지표 액세스

다음과 같은 여러 가지 방법으로 CloudWatch에 대한 Amazon EFS 지표를 볼 수 있습니다.

- Amazon EFS 콘솔에서
- CloudWatch 콘솔에서
- CloudWatch CLI 사용
- CloudWatch API 사용

CloudWatch 지표 및 경보를 보려면(Amazon EFS 콘솔)

1. 에 로그인 AWS Management Console 하고 https://console.aws.amazon.com/efs/ Amazon EFS 콘솔을 엽니다.
2. 파일 시스템을 선택합니다.
3. CloudWatch 지표를 확인하고자 하는 파일 시스템을 선택합니다.
4. 모니터링을 선택하여 파일 시스템 지표 페이지를 표시합니다.

   파일 시스템 지표 페이지에는 파일 시스템에 대한 기본 CloudWatch 지표 세트가 표시됩니다. 구성한 모든 CloudWatch 경보도 이러한 지표와 함께 표시됩니다. 최대 I/O 성능 모드를 사용하는 파일 시스템의 경우 기본 지표 세트에는 입출력 비율 제한 대신 버스트 크레딧 밸런스가 포함됩니다. 설정을 열면 액세스할 수 있는 지표 설정 대화 상자를 사용하여 기본 설정을 재정의할 수 있습니다.

   > **Note**
   >
   > 처리량 사용률(%) 지표는 CloudWatch 지표가 아니라 CloudWatch 측정치 계산을 사용하여 도출됩니다.

5. 다음과 같이 파일 시스템 지표 페이지의 컨트롤을 사용하여 지표와 경보가 표시되는 방식을 조정할 수 있습니다.

   - 디스플레이 모드를 시계열 또는 단일 값 사이에서 전환합니다.
   - 파일 시스템에 구성된 CloudWatch 경보를 표시하거나 숨깁니다.
   - CloudWatch에서 더 보기를 선택하여 CloudWatch에서 지표를 확인하세요.
   - 대시보드에 추가를 선택하여 CloudWatch 대시보드를 열고 표시된 지표를 추가합니다.
   - 표시된 지표 시간 창을 1시간에서 1주일로 조정합니다.

CloudWatch 지표 및 경보를 보려면(CloudWatch 콘솔)

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 지표를 선택합니다.
3. EFS 네임스페이스를 선택합니다.
4. (선택 사항)지표를 보려면 검색 필드에 이름을 입력합니다.
5. (선택 사항)차원을 기준으로 필터링하려면 FileSystemId를 선택합니다.

에서 지표에 액세스하려면 AWS CLI

- `list-metrics` 명령과 `--namespace "AWS/EFS"` 네임스페이스를 사용합니다. 자세한 내용은 AWS CLI 명령 참조를 참조하세요.

CloudWatch API에서 지표에 액세스하려면

- `GetMetricStatistics`를 호출합니다. 자세한 내용은 CloudWatch API 참조를 참조하세요.

Amazon EFS용 CloudWatch 지표 사용

Amazon EFS에서 보고하는 지표는 다양한 방법으로 분석이 기능한 정보를 제공합니다. 다음 목록은 몇 가지 일반적인 지표 사용 사례를 보여 줍니다. 모든 사용 사례를 망라한 것은 아니지만 시작하는 데 참고가 될 것입니다.

방법	관련 지표
처리량은 어떻게 확인할 수 있습니까?	매일 `TotalIOBytes` 지표의 `Sum` 통계를 모니터링하여 처리량을 확인할 수 있습니다.
전송된 총 데이터 양을 확인하려면 어떻게 해야 하나요?	`MeteredIOBytes` 의 일일 `Sum` 통계를 모니터링하여 전송된 총 데이터를 볼 수 있습니다.
파일 시스템에 연결된 Amazon EC2 인스턴스 개수는 어떻게 추적할 수 있습니까?	`ClientConnections` 지표의 `Sum` 통계를 모니터링할 수 있습니다. 1분 이상의 기간에 대한 평균 `ClientConnections` 를 계산하려면 합계를 기간(분 단위)으로 나눕니다.

방법	관련 지표
버스트 크레딧 잔고는 어떻게 확인할 수 있나요?	파일 시스템의 BurstCreditBalance 지표를 모니터링하여 잔고를 확인할 수 있습니다. 버스팅 및 버스트 크레딧에 대한 자세한 내용은 버스팅 처리량 단원을 참조하세요.

처리량 성능 모니터링

처리량 모니터링을 위한 CloudWatch 지표(TotalIOBytes, ReadIOBytes, WriteIOBytes, MetadataIOBytes)는 파일 시스템에서 발생하는 실제 처리량을 나타냅니다. 지표 MeteredIOBytes는 구동 중인 전체 측정된 처리량의 계산을 나타냅니다. Amazon EFS 콘솔 모니터링 섹션의 처리량 사용률(%) 그래프를 사용하여 처리량 사용률을 모니터링할 수 있습니다. 사용자 지정 Amazon CloudWatch 대시보드 또는 다른 모니터링 도구를 사용하는 경우 MeteredIOBytes를 PermittedThroughput과 비교하는 CloudWatch 지표 수식을 생성할 수 있습니다.

PermittedThroughput은 파일 시스템에 허용되는 처리량을 측정합니다. 이 값은 다음 방법 중 하나를 기반으로 합니다.

- 탄력적 처리량을 사용하는 파일 시스템의 경우 이 값은 파일 시스템의 최대 쓰기 처리량을 반영합니다.
- 프로비저닝된 처리량 모드를 사용하는 파일 시스템의 경우 EFS Standard 스토리지 클래스에 저장된 데이터의 양으로 인해 파일 시스템이 프로비저닝된 것보다 많은 양을 처리할 수 있는 경우 이 지표는 프로비저닝된 용량 대신 더 높은 처리량을 반영합니다.
- 버스팅 처리량을 사용하는 파일 시스템의 경우 이 값은 파일 시스템 크기 및 BurstCreditBalance의 함수입니다. BurstCreditBalance를 모니터링하여 파일 시스템이 기본 속도가 아닌 버스트 속도로 작동하고 있는지 확인하세요. 잔고가 지속적으로 0이거나 0에 가까운 경우 추가 처리량을 확보하려면 탄력적 처리량 또는 프로비저닝된 처리량으로 전환하는 것이 좋습니다.

MeteredIOBytes과 PermittedThroughput 값이 같으면 파일 시스템이 사용 가능한 모든 처리량을 소비하게 됩니다. 프로비저닝된 처리량을 사용하는 파일 시스템의 경우 추가 처리량을 프로비저닝할 수 있습니다.

CloudWatch 지표와 함께 지표 수식 사용

지표 수식을 사용하면 여러 Amazon CloudWatch 지표를 쿼리하고 수식을 사용하여 이러한 지표를 기반으로 새로운 시계열을 생성할 수 있습니다. CloudWatch 콘솔에서 결과 시계열을 시각화하고 대시보

드에 추가할 수 있습니다. 예를 들어, Amazon EFS 지표를 사용해 60으로 나눈 DataRead 작업의 샘플 수를 가져올 수 있습니다. 1분 동안 파일 시스템의 초당 읽기 수(평균)입니다. 지표 수학에 대한 자세한 내용은 Amazon CloudWatch 사용 설명서에서 CloudWatch 지표에 수학 표현식 사용을 참조하세요.

다음은 Amazon EFS에 유용하게 사용할 수 있는 지표 수식 표현식입니다.

주제

- 지표 수식: 처리량(MiB/초)
- 지표 수식: 처리량(%)
- 지표 수식: 허용 처리량 사용률 백분율
- 지표 수식: 처리량 IOPS
- 지표 수식: IOPS(%)
- 지표 수식: 평균 I/O 크기(KiB)
- Amazon EFS용 CloudFormation 템플릿을 통한 지표 수학 사용

지표 수식: 처리량(MiB/초)

특정 기간에 대한 평균 처리량(MiB/초)을 계산하려면, 먼저 합계 통계(DataReadIOBytes, DataWriteIOBytes, MetadataIOBytes 또는 TotalIOBytes)를 선택합니다. 그런 다음 MiB로 값을 변환한 후 기간을 초로 계산한 수로 나눕니다.

예가 되는 로직이 'TotalIOBytes 합계 ÷ 1048576(MiB로 변환) ÷ 초(기간을 초로 계산)'이라고 가정하겠습니다.

이 경우, CloudWatch 지표 정보는 다음과 같습니다.

ID	사용 가능한 지표	통계	Period
m1	• DataReadIOBytes • DataWriteIOBytes • MetadataIOBytes • TotalIOBytes	합계	1분

지표 수식 ID와 표현식은 다음과 같습니다.

ID	표현식
e1	(m1/1048576)/PERIOD(m1)

지표 수식: 처리량(%)

이 지표 수식 표현식은 다양한 I/O 유형에 사용되는 전체 처리량 중 백분율(예: 총 처리량 중 읽기 요청으로 구동되는 비율)을 계산합니다. 특정 기간의 여러 I/O 유형(DataReadIOBytes, DataWriteIOBytes 또는 MetadataIOBytes) 중 하나가 사용하는 전체 처리량(%)을 계산하려면 먼저 각 합계 통계에 100을 곱합니다. 그런 다음 같은 기간 동안 TotalIOBytes의 합계 통계로 결과를 나눕니다.

예가 되는 로직이 '(DataReadIOBytes 합계 x 100 (퍼센트로 변환)) ÷ TotalIOBytes 합계'라고 가정하겠습니다.

이 경우, CloudWatch 지표 정보는 다음과 같습니다.

ID	사용 가능한 지표	통계	Period
m1	• TotalIOBytes	합계	1분
m2	• DataReadIOBytes	합계	1분

지표 수식 ID와 표현식은 다음과 같습니다.

ID	표현식
e1	(m2*100)/m1

지표 수식: 허용 처리량 사용률 백분율

특정 기간의 허용 처리량 사용률(MeteredIOBytes)의 비율(%)을 계산하려면 먼저 처리량(MiB/초)에 100을 곱합니다. 그런 다음 같은 기간 동안 MiB로 변환한 PermittedThroughput의 평균 통계로 결과를 나눕니다.

예시가 되는 로직이 '(처리량(Mib/초)에 대한 지표 수식 표현식 x 100(퍼센트로 변환)) ÷ (PermittedThroughput의 합계 ÷ 1,048,576(바이트를 MiB로 변환))'이라고 가정하겠습니다.

이 경우, CloudWatch 지표 정보는 다음과 같습니다.

ID	사용 가능한 지표	통계	Period
m1	MeteredIOBytes	합계	1분
m2	PermittedThroughput	평균	1분

지표 수식 ID와 표현식은 다음과 같습니다.

ID	표현식
e1	(m1/1048576)/PERIOD(m1)
e2	m2/1048576
e3	((e1)*100)/(e2)

지표 수식: 처리량 IOPS

특정 기간에 대한 초당 평균 작업(IOPS)를 계산하려면 샘플 수 통계(DataReadIOBytes, DataWriteIOBytes, MetadataIOBytes, TotalIOBytes)를 해당 기간의 초로 나눕니다.

예가 되는 로직이 'DataWriteIOBytes 샘플 수 ÷ 초(기간을 초로 계산)'이라고 가정하겠습니다.

이 경우, CloudWatch 지표 정보는 다음과 같습니다.

ID	사용 가능한 지표	통계	Period
m1	• DataReadIOBytes • DataWriteIOBytes	샘플 수	1분

ID	사용 가능한 지표	통계	Period
	• MetadataIOBytes • TotalIOBytes		

지표 수식 ID와 표현식은 다음과 같습니다.

ID	표현식
e1	m1/PERIOD(m1)

지표 수식: IOPS(%)

특정 기간의 여러 I/O 유형(DataReadIOBytes, DataWriteIOBytes, MetadataIOBytes)의 초당 IOPS(%)를 계산하려면 먼저 샘플 수 통계에 100을 곱합니다. 그런 다음 같은 기간에 해당 값을 TotalIOBytes의 샘플 수 통계로 나눕니다.

예가 되는 로직이 '(MetadataIOBytes 샘플 수 x 100 (퍼센트로 변환)) ÷ TotalIOBytes 샘플 수'라고 가정하겠습니다.

이 경우, CloudWatch 지표 정보는 다음과 같습니다.

ID	사용 가능한 지표	통계	Period
m1	• TotalIOBytes	샘플 수	1분
m2	• DataReadIOBytes • DataWriteIOBytes • MetadataIOBytes	샘플 수	1분

지표 수식 ID와 표현식은 다음과 같습니다.

ID	표현식
e1	(m2*100)/m1

지표 수식: 평균 I/O 크기(KiB)

특정 기간에 대한 평균 I/O 크기(KiB)를 계산하려면 `DataReadIOBytes`, `DataWriteIOBytes`, `MetadataIOBytes` 지표의 합계 통계를 해당 지표의 샘플 수 통계로 나눕니다.

예가 되는 로직이 '(`DataReadIOBytes` 합계 ÷ 1024 (KiB로 변환)) ÷ `DataReadIOBytes` 샘플 수'라고 가정하겠습니다.

이 경우, CloudWatch 지표 정보는 다음과 같습니다.

ID	사용 가능한 지표	통계	Period
m1	• `DataReadIOBytes` • `DataWriteIOBytes` • `MetadataIOBytes`	합계	1분
m2	• `DataReadIOBytes` • `DataWriteIOBytes` • `MetadataIOBytes`	샘플 수	1분

지표 수식 ID와 표현식은 다음과 같습니다.

ID	표현식
e1	`(m1/1024)/m2`

Amazon EFS용 CloudFormation 템플릿을 통한 지표 수학 사용

CloudFormation 템플릿을 통해 지표 수학 표현식을 생성할 수도 있습니다. GitHub의 Amazon EFS 자습서에서 사용할 템플릿을 다운로드 받아 사용자 지정할 수 있습니다. CloudFormation 템플릿 사용에 대한 자세한 내용은 사용 설명서의 CloudFormation 템플릿 작업을 참조하세요. AWS CloudFormation

탑재 시도 성공 또는 실패 모니터링

Amazon CloudWatch Logs를 사용하면 클라이언트에 로그인하지 않고도 EFS 파일 시스템에 대한 탑재 시도의 성공 또는 실패를 원격으로 모니터링하고 보고할 수 있습니다. 다음 절차를 사용하여 CloudWatch Logs를 사용하여 파일 시스템 탑재 시도의 성공 또는 실패를 모니터링하도록 EC2 인스턴스를 구성합니다.

CloudWatch 로그에서 탑재 시도 성공 또는 실패 알림을 활성화하려면

1. EC2 인스턴스에 `amazon-efs-utils`를 설치하여 파일 시스템을 탑재합니다. 자세한 내용은 를 사용하여 Amazon EFS 클라이언트 자동 설치 또는 업데이트 AWS Systems Manager 또는 Amazon EFS 클라이언트 수동 설치을 참조하세요.
2. 파일 시스템을 탑재할 EC2 인스턴스에 `botocore` 설치합니다. 자세한 내용은 botocore 설치 및 업그레이드 단원을 참조하십시오.
3. `amazon-efs-utils`에서 CloudWatch Logs 기능을 활성화합니다. AWS Systems Manager 를 사용하여를 설치하고 구성하면 `amazon-efs-utils` CloudWatch 로깅이 자동으로 수행됩니다. `amazon-efs-utils` 패키지를 수동으로 설치하는 경우 `cloudwatch-log` 섹션에서 해당 `# enabled = true` 줄의 주석을 제거하여 `/etc/amazon/efs/efs-utils.conf` 구성 파일을 수동으로 업데이트해야 합니다. 다음 명령 중 하나를 사용하여 CloudWatch Logs를 수동으로 활성화합니다.

   Linux 인스턴스의 경우:

   ```
   sudo sed -i -e '/\[cloudwatch-log\]/{N;s/# enabled = true/enabled = true/}' /etc/
   amazon/efs/efs-utils.conf
   ```

   macOS 인스턴스의 경우:

   ```
   EFS_UTILS_VERSION= efs-utils-version
   sudo sed -i -e '/\[cloudwatch-log\]/{N;s/# enabled = true/enabled = true/;}' /usr/
   local/Cellar/amazon-efs-utils/${EFS_UTILS_VERSION}/libexec/etc/amazon/efs/efs-
   utils.conf
   ```

Mac2 인스턴스의 경우:

```
EFS_UTILS_VERSION= efs-utils-version
sudo sed -i -e '/\[cloudwatch-log\]/{N;s/# enabled = true/enabled = true/;}' /opt/
homebrew/Cellar/amazon-efs-utils/${EFS_UTILS_VERSION}/libexec/etc/amazon/efs/efs-
utils.conf
```

4. 선택적으로 CloudWatch Logs 그룹 이름을 구성하고 `efs-utils.conf` 파일에 로그 보존 일수를 설정할 수 있습니다. CloudWatch에서 탑재된 각 파일 시스템에 대해 별도의 로그 그룹을 만들려면 다음과 같이 `efs-utils.conf` 파일의 `log_group_name` 필드 끝에 `/{fs_id}`를 추가하세요.

   ```
   [cloudwatch-log]
   log_group_name = /aws/efs/utils/{fs_id}
   ```

5. `AmazonElasticFileSystemsUtils` AWS 관리형 정책을 EC2 인스턴스에 연결한 IAM 역할 또는 인스턴스에 구성된 AWS 자격 증명에 연결합니다. Systems Manager를 사용하여 이를 수행할 수 있습니다. 자세한 내용은 1단계: 필요한 권한으로 IAM 인스턴스 프로파일 구성 섹션을 참조하세요.

다음은 탑재 시도 상태 로그 항목의 예입니다.

```
Successfully mounted fs-12345678.efs.us-east-1.amazonaws.com at /home/ec2-user/efs
Mount failed, Failed to resolve "fs-01234567.efs.us-east-1.amazonaws.com"
```

CloudWatch Logs에서 탑재 상태를 보려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 로그 그룹을 선택합니다.
3. /aws/efs/utils 로그 그룹을 선택합니다. 각 Amazon EC2 인스턴스 및 EFS 파일 시스템 조합에 대한 로그 스트림이 표시됩니다.
4. 로그 스트림을 선택하면 탑재 시도 성공 또는 실패 상태를 비롯한 특정 로그 이벤트를 볼 수 있습니다.

Amazon EFS 모니터링을 위한 CloudWatch 경보 생성

경보로 인해 상태가 변경되면 Amazon SNS 메시지를 보내는 CloudWatch 경보를 생성할 수 있습니다. 경보는 지정한 기간 동안 단일 지표를 감시합니다. 이 경보는 여러 기간에 대해 주어진 임계값과 지표 값을 비교하여 하나 이상의 작업을 수행합니다. 이 작업은 Amazon SNS 주제 또는 Auto Scaling 정책으로 전송되는 알림입니다.

경보는 지속적인 상태 변경에 대해서만 작업을 호출합니다. CloudWatch 경보는 특정 상태에 있다는 이유만으로는 작업을 호출하지 않습니다. 상태가 변경되고 지정한 기간 동안 유지되어야 합니다.

Amazon EFS에 CloudWatch 경보를 사용하는 중요한 용도 중 하나는 파일 시스템에 유휴 암호화를 적용하는 것입니다. 생성할 때 Amazon EFS 파일 시스템에 유휴 상태의 암호화를 활성화할 수 있습니다. Amazon EFS 파일 시스템에 유휴 데이터 암호화 정책을 적용하려면 Amazon CloudWatch 및 AWS CloudTrail 을 사용해 파일 시스템 생성을 감지하고, 유휴 암호화가 활성화되어 있는지 확인합니다.

> **Note**
>
> 현재는 전송 중 암호화를 적용할 수 없습니다.

다음 절차에서는 Amazon EFS에 대한 경보를 만드는 방법을 간략하게 설명합니다.

콘솔 사용

CloudWatch 콘솔을 사용하여 경보를 설정하려면

1. 에 로그인 AWS Management Console 하고 https://console.aws.amazon.com/cloudwatch/ CloudWatch 콘솔을 엽니다.
2. 경보 생성을 선택합니다. 그러면 Create Alarm Wizard가 시작됩니다.
3. EFS 지표를 선택하고 Amazon EFS 지표를 스크롤하여 경보를 생성할 지표를 찾습니다. 이 대화 상자에서 Amazon EFS 지표를 표시하려면 파일 시스템의 파일 시스템 ID를 검색합니다. 경보를 생성할 지표를 선택하고 다음을 선택합니다.
4. 지표에 대한 이름, 설명, 다음 경우 항상 값을 입력합니다.
5. 경보 상태 도달 시 CloudWatch에서 이메일을 보내게 하려면, 이 경우 언제든 경보: 필드에서 경보 상태를 선택합니다. 알림 보내기: 필드에서 기존 SNS 주제를 선택합니다. 주제 생성을 선택한 경우 새 이메일 구독 목록에 대한 명칭 및 이메일 주소를 설정할 수 있습니다. 이 목록은 향후 경보를 위해 필드에 저장되고 표시됩니다.

> **Note**
>
> 새 Amazon SNS 주제를 생성하기 위해 주제 생성을 사용할 경우 이메일 주소는 알림을 받기 전에 검증되어야 합니다. 이메일은 경보가 경보 상태에 입력될 때만 전송됩니다. 이러한 경보 상태 변경이 이메일이 검증되기 전에 발생할 경우에는 알림을 받지 못합니다.

6. 이제 경보 미리 보기 영역에서 생성할 경보를 미리 볼 수 있습니다. 경보 생성을 선택합니다.

사용 AWS CLI

를 사용하여 경보를 설정하려면 AWS CLI

- `put-metric-alarm`를 호출합니다. 자세한 내용은 AWS CLI 명령 참조를 참조하세요.

CloudWatch API 사용

CloudWatch API를 사용하여 경보 설정

- `PutMetricAlarm`을 호출합니다. 자세한 내용은 CloudWatch API 참조를 참조하세요.

를 사용하여 Amazon EFS API 호출 로깅 AWS CloudTrail

Amazon EFS는 Amazon EFS에서 사용자 AWS CloudTrail, 역할 또는 서비스가 수행한 작업에 대한 레코드를 제공하는 AWS 서비스와 통합됩니다. CloudTrail은 Amazon EFS 콘솔의 직접 호출과 Amazon EFS API 작업에 대한 코드 직접 호출을 포함하여 Amazon EFS에 대한 모든 API 직접 호출을 이벤트로 캡처합니다.

추적을 생성하면 Amazon EFS 이벤트를 포함한 CloudTrail 이벤트를 지속적으로 Amazon S3 버킷에 배포할 수 있습니다. 추적을 구성하지 않은 경우에도 이벤트 기록에서 CloudTrail 콘솔의 최신 이벤트를 볼 수 있습니다. CloudTrail에서 수집한 정보를 사용하여 Amazon EFS에 수행된 요청, 요청이 수행된 IP 주소, 요청을 수행한 사람, 요청이 수행된 시간 및 추가 세부 정보를 확인할 수 있습니다.

자세한 내용은 AWS CloudTrail 사용 설명서의 What is AWS CloudTrail?를 참조하세요.

CloudTrail의 Amazon EFS 정보

CloudTrail은 계정을 생성할 AWS 계정 때에서 활성화됩니다. Amazon EFS에서 활동이 발생하면 해당 활동이 이벤트 기록의 다른 AWS 서비스 이벤트와 함께 CloudTrail 이벤트에 기록됩니다. 에서 최근

이벤트를 보고 검색하고 다운로드할 수 있습니다 AWS 계정. 자세한 설명은 CloudTrail 이벤트 기록 작업을 참조하세요.

Amazon EFS에 대한 이벤트를 AWS 계정포함하여에 이벤트를 지속적으로 기록하려면 추적을 생성합니다. CloudTrail은 추적을 사용하여 Amazon S3 버킷으로 로그 파일을 전송할 수 있습니다. 기본적으로 콘솔에서 추적을 생성하면 추적이 모든 AWS 리전에 적용됩니다. 추적은 AWS 파티션 AWS 리전 의 모든에서 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 내용은AWS CloudTrail 사용 설명서에서 다음 주제를 참조하세요.

- AWS 계정에 대한 추적 생성
- AWS CloudTrail 로그와의 서비스 통합
- CloudTrail에 대한 Amazon SNS 알림 구성
- 여러 리전에서 CloudTrail 로그 파일 수신 및 여러 계정에서 CloudTrail 로그 파일 수신

모든 Amazon EFS Amazon EFS API는 CloudTrail에 의해 로깅됩니다. 예를 들어, `CreateFileSystem`, `CreateMountTarget` 및 `CreateTags` 작업에 대한 직접 호출은 CloudTrail 로그 파일의 항목을 생성합니다.

모든 이벤트 또는 로그 항목에는 요청을 생성했던 사용자에 대한 정보가 포함됩니다. ID 정보를 이용하면 다음을 쉽게 판단할 수 있습니다.

- 요청이 루트 사용자 또는 AWS Identity and Access Management (IAM) 사용자 자격 증명으로 이루어졌는지 여부입니다.
- 역할 또는 페더레이션 사용자의 임시 자격 증명을 사용하여 요청이 생성되었는지 여부.
- 요청이 다른 AWS 서비스에서 이루어졌는지 여부입니다.

자세한 내용은 AWS CloudTrail 사용 설명서의 CloudTrail userIdentity 요소를 참조하세요.

Amazon EFS 로그 파일 항목 이해

추적이란 지정한 Amazon S3 버킷에 이벤트를 로그 파일로 입력할 수 있게 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.

다음 예는 콘솔에서 파일 시스템의 태그를 만든 경우 CreateTags 작업을 실행하는 CloudTrail 로그 항목을 보여줍니다.


```
{
 "eventVersion": "1.06",
 "userIdentity": {
  "type": "Root",
  "principalId": "111122223333",
  "arn": "arn:aws:iam::111122223333:root",
  "accountId": "111122223333",
  "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
  "sessionContext": {
   "attributes": {
    "mfaAuthenticated": "false",
    "creationDate": "2017-03-01T18:02:37Z"
   }
  }
 },
 "eventTime": "2017-03-01T19:25:47Z",
 "eventSource": "elasticfilesystem.amazonaws.com",
 "eventName": "CreateTags",
 "awsRegion": "us-west-2",
 "sourceIPAddress": "192.0.2.0",
 "userAgent": "console.amazonaws.com",
 "requestParameters": {
  "fileSystemId": "fs-00112233",
  "tags": [{
    "key": "TagName",
    "value": "AnotherNewTag"
   }
  ]
 },
 "responseElements": null,
 "requestID": "dEXAMPLE-feb4-11e6-85f0-736EXAMPLE75",
 "eventID": "eEXAMPLE-2d32-4619-bd00-657EXAMPLEe4",
 "eventType": "AwsApiCall",
 "apiVersion": "2015-02-01",
 "recipientAccountId": "111122223333"
}
```


다음 예는 콘솔에서 파일 시스템의 태그를 삭제한 경우 DeleteTags 작업을 실행하는 CloudTrail 로그 항목을 보여줍니다.

```
{
 "eventVersion": "1.06",
 "userIdentity": {
  "type": "Root",
  "principalId": "111122223333",
  "arn": "arn:aws:iam::111122223333:root",
  "accountId": "111122223333",
  "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
  "sessionContext": {
   "attributes": {
    "mfaAuthenticated": "false",
    "creationDate": "2017-03-01T18:02:37Z"
   }
  }
 },
 "eventTime": "2017-03-01T19:25:47Z",
 "eventSource": "elasticfilesystem.amazonaws.com",
 "eventName": "DeleteTags",
 "awsRegion": "us-west-2",
 "sourceIPAddress": "192.0.2.0",
 "userAgent": "console.amazonaws.com",
 "requestParameters": {
  "fileSystemId": "fs-00112233",
  "tagKeys": []
 },
 "responseElements": null,
 "requestID": "dEXAMPLE-feb4-11e6-85f0-736EXAMPLE75",
 "eventID": "eEXAMPLE-2d32-4619-bd00-657EXAMPLEe4",
 "eventType": "AwsApiCall",
 "apiVersion": "2015-02-01",
 "recipientAccountId": "111122223333"
}
```


EFS 서비스 연결 역할에 대한 로그 항목

Amazon EFS 서비스 연결 역할은 AWS 리소스에 대한 API 호출을 수행합니다. EFS 서비스 연결 역할에서 수행한 직접 호출의 경우 CloudTrail 로그 항목에 `username: AWSServiceRoleForAmazonElasticFileSystem`이 표시됩니다. EFS 및 서비스 연결 역할에 대한 자세한 내용은 Amazon EFS에 대해 서비스 연결 역할 사용 섹션을 참조하세요.

다음 예제에서는 에서 AWSServiceRoleForAmazonElasticFileSystem 서비스 연결 역할을 생성할 때 `CreateServiceLinkedRole` 작업을 보여 주는 CloudTrail 로그 항목을 표시합니다.


```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "111122223333",
        "arn": "arn:aws:iam::111122223333:user/user1",
        "accountId": "111122223333",
        "accessKeyId": "A111122223333",
        "userName": "user1",
        "sessionContext": {
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2019-10-23T22:45:41Z"
            }
        },
        "invokedBy": "elasticfilesystem.amazonaws.com”
    },
    "eventTime": "2019-10-23T22:45:41Z",
    "eventSource": "iam.amazonaws.com",
    "eventName": "CreateServiceLinkedRole",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "user_agent",
    "requestParameters": {
        "aWSServiceName": "elasticfilesystem.amazonaws.com”
    },
    "responseElements": {
        "role": {
            "assumeRolePolicyDocument":
 "111122223333-10-111122223333Statement111122223333Action111122223333AssumeRole111122223333Effe
%22%3A%20%22Allow%22%2C%20%22Principal%22%3A%20%7B%22Service%22%3A%20%5B%22
 elasticfilesystem.amazonaws.com%22%5D%7D%7D%5D%7D",
            "arn": "arn:aws:iam::111122223333:role/aws-service-role/
elasticfilesystem.amazonaws.com/AWSServiceRoleForAmazonElasticFileSystem",
            "roleId": "111122223333",
            "createDate": "Oct 23, 2019 10:45:41 PM",
            "roleName": "AWSServiceRoleForAmazonElasticFileSystem",
            "path": "/aws-service-role/elasticfilesystem.amazonaws.com/“
        }
    },
```

```
    "requestID": "11111111-2222-3333-4444-abcdef123456",
    "eventID": "11111111-2222-3333-4444-abcdef123456",
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

다음 예제에서는 sessionContext에 표시된 AWSServiceRoleForAmazonElasticFileSystem 서비스 연결 역할에서 생성된 CreateNetworkInterface 작업을 보여 주는 CloudTrail 로그 항목을 표시합니다.

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AIDACKCEVSQ6C2EXAMPLE",
        "arn": "arn:aws:sts::0123456789ab:assumed-role/
AWSServiceRoleForAmazonElasticFileSystem/0123456789ab",
        "accountId": "0123456789ab",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AIDACKCEVSQ6C2EXAMPLE",
                "arn": "arn:aws:iam::0123456789ab:role/aws-service-role/
elasticfilesystem.amazonaws.com/AWSServiceRoleForAmazonElasticFileSystem",
                "accountId": "0123456789ab",
                "userName": "AWSServiceRoleForAmazonElasticFileSystem"
            },
            "webIdFederationData": {},
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2019-10-23T22:50:05Z"
            }
        },
        "invokedBy": "AWS Internal"
    },
    "eventTime": "20You 19-10-23T22:50:05Z",
    "eventSource": "ec2.amazonaws.com",
    "eventName": "CreateNetworkInterface",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "elasticfilesystem.amazonaws.com”,
    "userAgent": "elasticfilesystem.amazonaws.com",
    "requestParameters": {
        "subnetId": "subnet-71e2f83a",
```

```
        "description": "EFS mount target for fs-1234567 (fsmt-1234567)",
        "groupSet": {},
        "privateIpAddressesSet": {}
    },
    "responseElements": {
        "requestId": "0708e4ad-03f6-4802-b4ce-4ba987d94b8d",
        "networkInterface": {
            "networkInterfaceId": "eni-0123456789abcdef0",
            "subnetId": "subnet-12345678",
            "vpcId": "vpc-01234567",
            "availabilityZone": "us-east-1b",
            "description": "EFS mount target for fs-1234567 (fsmt-1234567)",
            "ownerId": "666051418590",
            "requesterId": "0123456789ab",
            "requesterManaged": true,
            "status": "pending",
            "macAddress": "00:bb:ee:ff:aa:cc",
            "privateIpAddress": "192.0.2.0",
            "privateDnsName": "ip-192-0-2-0.ec2.internal",
            "sourceDestCheck": true,
            "groupSet": {
                "items": [
                    {
                        "groupId": "sg-c16d65b6",
                        "groupName": "default"
                    }
                ]
            },
            "privateIpAddressesSet": {
                "item": [
                    {
                        "privateIpAddress": "192.0.2.0",
                        "primary": true
                    }
                ]
            },
            "tagSet": {}
        }
    },
    "requestID": "11112222-3333-4444-5555-666666777777",
    "eventID": "aaaabbbb-1111-2222-3333-444444555555",
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
```

```
}
```

EFS 인증을 위한 로그 항목

NFS 클라이언트에 대한 Amazon EFS 인증은 `NewClientConnection` 및 `UpdateClientConnection` CloudTrail 이벤트를 생성합니다. 초기 연결 및 재연결 직후에 연결이 승인되면 `NewClientConnection` 이벤트가 발생합니다. 연결이 재승인되고 허용된 작업 목록이 변경되면 `UpdateClientConnection` 이 발생합니다. 허용된 작업의 새 목록에 `ClientMount`가 포함되지 않은 경우에도 이벤트가 발생합니다. EFS 권한 부여에 대한 자세한 내용은 IAM을 사용하여 파일 시스템에 대한 액세스 제어 섹션을 참조하세요.

다음 예제는 `NewClientConnection` 이벤트를 보여 주는 CloudTrail 로그 항목입니다.

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AIDACKCEVSQ6C2EXAMPLE",
        "arn": "arn:aws:sts::0123456789ab:assumed-role/abcdef0123456789",
        "accountId": "0123456789ab",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE ",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AIDACKCEVSQ6C2EXAMPLE",
                "arn": "arn:aws:iam::0123456789ab:role/us-east-2",
                "accountId": "0123456789ab",
                "userName": "username"
            },
            "webIdFederationData": {},
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2019-12-23T17:50:16Z"
            },
            "ec2RoleDelivery": "1.0"
        }
    },
    "eventTime": "2019-12-23T18:02:12Z",
    "eventSource": "elasticfilesystem.amazonaws.com",
    "eventName": "NewClientConnection",
    "awsRegion": "us-east-2",
    "sourceIPAddress": "AWS Internal",
```

```
    "userAgent": "elasticfilesystem",
    "requestParameters": null,
    "responseElements": null,
    "eventID": "27859ac9-053c-4112-aee3-f3429719d460",
    "readOnly": true,
    "resources": [
        {
            "accountId": "0123456789ab",
            "type": "AWS::EFS::FileSystem",
            "ARN": "arn:aws:elasticfilesystem:us-east-2:0123456789ab:file-system/
fs-01234567"
        },
        {
            "accountId": "0123456789ab",
            "type": "AWS::EFS::AccessPoint",
            "ARN": "arn:aws:elasticfilesystem:us-east-2:0123456789ab:access-point/
fsap-0123456789abcdef0"
        }
    ],
    "eventType": "AwsServiceEvent",
    "recipientAccountId": "0123456789ab",
    "serviceEventDetails": {
        "permissions": {
            "ClientRootAccess": true,
            "ClientMount": true,
            "ClientWrite": true
        },
        "sourceIpAddress": "10.7.3.72"
    }
}
```


파일 시스템 유휴 암호화용 Amazon EFS 로그 파일 항목

Amazon EFS는 전송 중 암호화와 유휴 암호화라는 두 가지 파일 시스템 암호화 사용 옵션을 제공합니다. 자세한 내용은 Amazon EFS의 데이터 암호화 단원을 참조하십시오.

Amazon EFS는 데이터 키를 생성하고 Amazon EFS 데이터를 복호화하기 위한 AWS KMS API 요청을 할 때 암호화 컨텍스트를 전송합니다. 파일 시스템 ID는 유휴 암호화가 적용된 모든 파일 시스템에 대한 암호화 컨텍스트입니다. CloudTrail 로그 항목의 `requestParameters` 필드에서 암호화 컨텍스트는 다음과 비슷하게 표시됩니다.


```
"EncryptionContextEquals": {}
```

```
"aws:elasticfilesystem:filesystem:id" : "fs-4EXAMPLE"
```

AWS 결제 및 사용 보고서

Amazon EFS를 사용하는 경우 사용하는 리소스에 대해서만 비용을 지불하며 최소 요금이나 설정 요금은 없습니다. 다른 AWS 서비스와 마찬가지로 요금은 월별로 인용되지만 사용량은 시간 단위로 비례 할당되며 한 달 동안의 평균 사용량에 대해 요금이 청구됩니다.

AWS는 Amazon EFS에 대해 다음과 같은 보고서를 제공합니다.

- 결제 보고서 — Amazon EFS를 포함하여 사용 중인 AWS 서비스의 모든 활동에 대한 개괄적인 보기를 제공하는 여러 보고서입니다. AWS는 항상 EFS 파일 시스템 소유자에게 Amazon EFS 요금을 청구합니다. 결제 보고서에 대한 자세한 내용은 Amazon EFS에 대한 AWS Billing 보고서 섹션을 참조하세요.
- 사용 보고서 – 특정 서비스에 대한 활동을 시간, 일 또는 월 기준으로 요약한 보고서입니다. 포함시킬 사용 유형과 작업을 선택할 수 있습니다. 데이터가 축적되는 방식을 선택할 수도 있습니다. 자세한 내용은 Amazon EFS에 대한 AWS 사용 보고서 섹션을 참조하세요.

  Amazon EFS에 대한 결제 및 사용 보고에 관한 내용은 다음 주제를 참조하세요.

주제

- 비용 할당 EFS 파일 시스템 태그 사용
- Amazon EFS에 대한 AWS Billing 보고서
- Amazon EFS에 대한 AWS 사용 보고서
- Amazon EFS에 대한 결제 및 사용 보고서 이해

비용 할당 EFS 파일 시스템 태그 사용

개별 프로젝트나 프로젝트 그룹에 대해 스토리지 비용이나 기타 기준을 추적하려면 비용 할당 태그를 사용하여 EFS 파일 시스템에 레이블을 지정합니다. 비용 할당 태그는 파일 시스템과 연관된 키-값 페어입니다. 비용 할당 태그를 활성화하고 나면 AWS가 이 태그를 사용하여 비용 할당 보고서에서 리소스 비용을 구성합니다.

비용 할당 보고서에는 계정에 대한 AWS 사용이 제품 범주 및 연결된 계정 사용자별로 나열됩니다. 이 보고서에는 세부 결제 보고서(Amazon EFS에 대한 결제 및 사용 보고서 이해 참조)와 동일한 행 항목 및 태그 키에 대한 추가 열이 포함되어 있습니다.

AWS는 AWS 생성 태그와 사용자 정의 태그라는 두 가지 유형의 비용 할당 태그를 제공합니다. AWS는 Amazon EFS CreateFileSystem 이벤트 발생 후 AWS 생성 `aws:createdBy` 태그 키를 자동으로 정의, 생성 및 적용합니다. 사용자는 파일 시스템에 사용자 정의 태그를 정의, 생성 및 적용합니다.

Billing and Cost Management 콘솔에서 개별적으로 두 유형의 태그를 모두 활성화해야만 결제 보고서에 표시가 됩니다. AWS 생성 태그에 대한 자세한 내용은 AWS Billing 사용 설명서의 AWS 비용 할당 태그를 사용한 비용 구성 및 추적을 참조하세요.

- EFS 리소스에 대한 사용자 정의 태그 생성 방법에 대한 자세한 내용은 EFS 리소스에 태그 지정을 참조하세요.
- Billing and Cost Management 콘솔에서 사용자 정의 태그를 활성화하는 방법에 대한 자세한 내용은 AWS Billing 사용 설명서의 사용자 정의 비용 할당 태그 활성화를 참조하세요.
- AWS 생성 태그를 Billing and Cost Management 콘솔에서 활성화하는 방법에 대한 자세한 내용은 AWS Billing 사용 설명서의 AWS 생성 비용 할당 태그 활성화를 참조하세요.

사용자 정의 비용 할당 태그

사용자 정의 비용 할당 태그는 다음과 같은 요소로 구성되어 있습니다.

- 태그 키. 태그 키는 태그의 이름입니다. 예를 들어, project/Trinity 태그에서 프로젝트가 키입니다. 태그 키는 대/소문자 구분 문자열로 유니코드 문자 1 ~ 128자를 포함할 수 있습니다.

- 태그 값. 태그 값은 필수 문자열입니다. 예를 들어, project/Trinity 태그에서 Trinity가 값입니다. 태그 값은 대/소문자 구분 문자열로 유니코드 문자 0 ~ 256자를 포함할 수 있습니다.

사용자 정의 태그에 대한 자세한 내용은 AWS Billing 사용 설명서의 사용자 정의 비용 할당 태그 사용을 참조하세요.

Amazon EFS에 대한 AWS Billing 보고서

AWS에서 보내는 월 청구서에는 AWS 서비스 및 기능에 따라 사용 정보와 비용이 구분되어 있습니다. 월별 보고서, 비용 할당 보고서, 세부 결제 보고서 등 몇 가지 AWS 결제 보고서가 제공되고 있습니다. 자세한 내용은 AWS Billing 사용 설명서의 청구서 이해 섹션을 참조하세요.

Amazon EFS 스토리지 사용에 대한 보다 상세한 보고서를 보려면 AWS Data Exports를 사용하여 AWS Cost and Usage Report(AWS CUR) 내보내기를 생성합니다. AWS CUR 2.0을 사용하면 AWS 결제 보고서를 소유한 Amazon Simple Storage Service(Amazon S3) 버킷에 게시할 수 있습니다.

Amazon EFS 사용 보고서에는 사용 유형 및 AWS 리전별로 작업이 나열됩니다. 자세한 내용은 AWS Data Exports 사용 설명서의 Cost and Usage Report(CUR) 2.0을 참조하세요.

EFS 파일 시스템은 다음 사용 범주에 따라 요금이 부과됩니다.

- 스토리지(기가바이트당[GB]) - EFS 파일 시스템에 월간 저장된 데이터 양으로, 스토리지 클래스(EFS Standard, EFS Infrequent Access(IA), EFS Archive)에 따라 가격이 다릅니다.
- 처리량(GB당 또는 초당 메비바이트[MiBps]) - 파일 시스템의 처리량에 따라 월간 전송된 데이터(읽기/쓰기 작업) 또는 프로비저닝된 양을 초과한 데이터의 양입니다.
- 데이터 액세스 및 전환(MiBps당) - 매월 EFS Infrequent Access(IA) 및 EFS Archive 스토리지 클래스로 읽기 및 전송된 데이터의 양입니다.
- 백업 스토리지(GB당) - 백업에서 매월 사용하는 스토리지 공간의 양입니다. 백업 보존 기간을 늘리거나 사용자가 시작한 볼륨 백업을 추가로 수행하면 파일 시스템에서 사용하는 백업 스토리지의 양이 늘어납니다. 자세한 내용은 EFS 파일 시스템 백업 섹션을 참조하세요.
- 복제(GB당) - 소스 및 대상 EFS 파일 시스템의 일반적인 스토리지 및 읽기, 쓰기, 계층화 활동 요금과 월별 데이터 전송 활동에 적용되는 요금입니다.

Amazon EFS 요금에 대한 자세한 내용은 Amazon EFS 요금을 참조하세요.

Amazon EFS에 대한 결제 및 사용 보고서에서 사용되는 코드와 약어에 대한 내용은 Amazon EFS에 대한 결제 및 사용 보고서 이해 섹션을 참조하세요.

Amazon EFS에 대한 AWS 사용 보고서

사용 보고서를 다운로드하면 사용 데이터를 시간, 일 또는 월 기준으로 집계할 수 있습니다. Amazon EFS 사용 보고서에는 사용 유형 및 AWS 리전별로 작업이 나열됩니다. Amazon EFS 스토리지 사용에 대한 자세한 보고서를 보려면 동적으로 생성된 AWS 사용 보고서를 다운로드합니다. 포함시킬 사용 유형, 작업 및 기간을 선택할 수 있습니다. 데이터가 축적되는 방식을 선택할 수도 있습니다. 사용 보고서 다운로드에 대한 자세한 내용은 AWS Data Exports 사용 설명서의 AWS 사용 보고서 다운로드를 참조하세요.

Amazon EFS 사용 보고서에는 다음 정보가 포함됩니다.

- 서비스 - Amazon EFS
- UsageType – 다음 값 중 하나입니다.
  - 스토리지의 유형을 식별하는 코드

  - 요청의 유형을 식별하는 코드
  - 데이터 전송의 유형을 식별하는 코드입니다.
  - 처리량 모드를 식별하는 코드
  - 백업 사용량을 식별하는 코드

  Amazon EFS 사용 유형에 대한 자세한 설명은 Amazon EFS에 대한 결제 및 사용 보고서 이해 섹션을 참조하세요.
- 리소스 – 나열된 사용과 관련된 리소스의 이름입니다.
- StartTime – 사용이 적용되는 날짜의 시작 시간으로, UTC(협정 세계시)를 따릅니다.
- EndTime – 사용이 적용되는 날짜의 종료 시간으로, UTC(협정 세계시)를 따릅니다.
- UsageValue – 다음 볼륨 값 중 하나입니다. 데이터의 일반적인 측정 단위는 기가바이트(GB)입니다. 그러나 서비스 및 보고서에 따라 TB(테라바이트)가 대신 나타날 수 있습니다.
  - 지정된 기간 동안 요청 수
  - 전송된 데이터의 양
  - 지정된 시간에 저장된 데이터의 양

Amazon EFS에 대한 결제 및 사용 보고서에서 사용되는 코드와 약어에 대한 내용은 Amazon EFS에 대한 결제 및 사용 보고서 이해 섹션을 참조하세요.

Amazon EFS에 대한 결제 및 사용 보고서 이해

Amazon EFS 결제 및 사용 보고서는 코드와 약어를 사용합니다. 다음 테이블의 사용 유형에 대해 *region*을 이 목록의 약어로 바꿉니다.

- APE1: 아시아 태평양(홍콩)
- APN1: 아시아 태평양(도쿄)
- APN2: 아시아 태평양(서울)
- APN3: 아시아 태평양(오사카)
- APS1: 아시아 태평양(싱가포르)
- APS2: 아시아 태평양(시드니)
- APS3: 아시아 태평양(뭄바이)
- APS4: 아시아 태평양(자카르타)
- APS5: 아시아 태평양(하이데라바드)

- APS6: 아시아 태평양(멜버른)
- CAN1: 캐나다(중부)
- CAN2: 캐나다 서부(캘거리)
- CNN1: 중국(베이징)
- CNW1: 중국(닝샤)
- AFS1: 아프리카(케이프타운)
- EUC2: 유럽(취리히)
- EUN1: 유럽(스톡홀름)
- EUS2: 유럽(스페인)
- EUC1: 유럽(프랑크푸르트)
- EU: 유럽(아일랜드)
- EUS1: 유럽(밀라노)
- EUW2: 유럽(런던)
- EUW3: 유럽(파리)
- ILC1: 이스라엘(텔아비브)
- MEC1: 중동(UAE)
- MES1: 중동(바레인)
- SAE1: 남아메리카(상파울루)
- UGW1: AWS GovCloud(미국 서부)
- UGE1: AWS GovCloud(미국 동부)
- USE1(또는 접두사 없음): 미국 동부(버지니아 북부)
- USE2: 미국 동부(오하이오)
- USW1: 미국 서부(캘리포니아 북부)
- USW2: 미국 서부(오레곤)

요금에 대한 자세한 내용은 Amazon EFS 요금을 AWS 리전참조하세요.

다음 표의 첫째 열에는 결제 및 사용 보고서에 나타나는 사용 유형이 나열됩니다. 데이터의 일반적인 측정 단위는 기가바이트(GB)입니다. 그러나 프로비저닝된 처리량의 경우 초당 메비바이트(MiBps)가 대신 사용됩니다.

Amazon EFS의 사용 유형

사용 유형	CloudWatch 지표	단위	Granularity	설명
region-ArchiveDataAccess-Bytes	해당 사항 없음	GB	해당 사항 없음	EFS Archive 스토리지로 전송되거나 EFS Archive 스토리지에서 액세스된 데이터의 양입니다.
region-ArchiveEarlyDelete-ByteHrs	해당 사항 없음	GB-Mo	해당 사항 없음	EFS Archive 스토리지에서 최소 90일 약정 기간이 종료되기 전에 삭제된 파일의 비례 배분된 스토리지 사용량입니다.
region-ArchiveEarlyDelete-SmallFiles	해당 사항 없음	GB-Mo	해당 사항 없음	EFS 아카이브 스토리지에서 최소 90일 약정이 종료되기 전에 삭제된 작은 파일(128KB 미만)에 대한 비례 배분된 스토리지 사용량입니다.
region-ArchiveTimedStorage-ByteHrs	StorageBytes.Archive	GB-Mo	시간당	EFS Archive 스토리지에서 데이터가 저장된 GB-월 수입니다.
region-ArchiveTimedStorage-SmallFiles	StorageBytes.ArchiveSizeOverhead	GB-Mo	시간당	EFS Archive 스토리지에 작은 객체(128KB 미만)가 저장된 GB-월 수입니다.
region-ETDataAccess-Bytes	TotalIOBytes	GB	시간당	탄력적 처리량을 사용하여 파일에 대해 쓰거나 읽은 데이터의 양입니다.

사용 유형	CloudWatch 지표	단위	Granularity	설명
region-IADataAccess-Bytes	해당 사항 없음	GB	시간당	EFS Infrequent Access(IA) 스토리지로 전송되거나 액세스된 데이터의 양입니다.
region-IATimedStorage-ByteHrs	StorageBytes.IA	GB-Mo	시간당	EFS IA 스토리지에서 데이터가 저장된 GB-월 수입니다.
region-IATimedStorage-ET-ByteHrs	StorageBytes.IA	GB-Mo	시간당	탄력적 처리량을 사용하는 파일의 데이터가 EFS IA 스토리지에 저장된 GB-월 수입니다.
region-IATimedStorage-ET-SmallFiles	StorageBytes.IASizeOverhe	GB-Mo	시간당	탄력적 처리량을 사용하는 파일의 경우, EFS IA 스토리지에서 저장된 작은 파일(128KB 미만)의 GB-월 수입니다.
region-IATimedStorage-SmallFiles	StorageBytes.IASizeOverhe	GB-Mo	시간당	EFS IA 스토리지에 작은 파일(128KB 미만)가 저장된 GB-월 수입니다.
region-IATimedStorage-Z-ByteHrs	StorageBytes.IA	GB-Mo	시간당	One Zone 파일 시스템의 EFS IA 스토리지에 저장된 데이터의 실제 크기입니다.
region-IATimedStorage-Z-SmallFiles	StorageBytes.IASizeOverhe	GB-Mo	시간당	One Zone 파일 시스템의 EFS IA 스토리지에 저장된 작은 파일(128KB 미만)의 GB-월 수입니다.

사용 유형	CloudWatch 지표	단위	Granularity	설명
region-IncludedTP-MiBpsHrs	PermittedThroughput	MiBps-Mo	시간당	파일 시스템이 구동할 수 있는 최대 처리량입니다.
region-ProvisionedTP-MiBpsHrs	PermittedThroughput	MiBps-Mo	시간당	프로비저닝된 처리량을 사용하는 파일 시스템의 경우 프로비저닝된 처리량 또는 허용되는 최대 처리량입니다.
region-TimedStorage-ByteHrs	StorageBytes.Standard	GB-Mo	시간당	리전 파일 시스템을 위해 EFS Standard 스토리지에 저장된 데이터의 실제 총량입니다.
region-TimedStorage-Z-ByteHrs	StorageBytes.Standard	GB-Mo	시간당	One Zone 파일 시스템을 위해 EFS Standard 스토리지에 저장된 데이터의 실제 총량입니다.

참고

- EFS Archive 스토리지는 데이터가 90일 이전에 삭제되더라도 최소 90일의 스토리지 기간에 대해 요금이 부과됩니다.
- 계층화 요금은 EFS Standard 스토리지에서 EFS Infrequent Access(IA) 스토리지로 계층화된 데이터와 EFS IA 스토리지에서 EFS Standard 스토리지로 계층화된 데이터에 적용됩니다.
- EFS IA 및 EFS Archive 스토리지의 청구 가능한 파일 크기는 최소 128KiB입니다. 128KiB보다 작은 파일은 이러한 스토리지 클래스로 계층화할 수 있지만 적절한 스토리지 클래스 요금으로 128KiB의 스토리지에 대해 요금이 부과됩니다.
- 탄력적 처리량을 사용하는 파일 시스템의 경우 메타데이터 작업은 처음 4KiB 이후 1KiB 단위로 측정되고 데이터 작업은 처음 32KiB 이후 1KiB 단위로 측정됩니다.
- 스토리지 사용량은 이진 기가바이트(GB) 단위로 계산됩니다. 여기서 1GB는 2^{30}바이트입니다. Amazon EFS 청구서를 계산하기 위해 스토리지 및 처리량 사용량을 측정하는 방법에 Amazon EFS에서 파일 시스템 및 객체 크기를 보고하는 방법 대한 자세한 내용은 섹션을 참조하세요.

- GB-월은 한 달 동안의 총 GB-시간을 집계하여 총 GB-시간 수를 구한 다음 해당 월의 시간 수로 나누어 구합니다.

사용 보고서의 추적 작업

작업은 지정된 사용 유형에 따라 EFS 파일 시스템에 대해 취해진 조치를 설명합니다. 작업은 `Read` 또는 `Delete` 같이 이름에 설명이 포함된 코드로 표시가 됩니다. 이러한 코드를 사용하여 파일 시스템에 대한 어떤 조치가 특정한 사용 유형을 생성했는지 확인합니다. 사용 보고서를 생성할 때 선택에 따라 보고서에 모든 작업 또는 `Write` 같은 특정 작업을 포함시킬 수 있습니다.

Amazon EFS 성능 사양

다음 섹션에서는 Amazon EFS 성능에 대한 개요와 파일 시스템 구성이 주요 성능 차원에 미치는 영향을 설명합니다. 또한 파일 시스템의 성능을 최적화하기 위한 몇 가지 중요한 팁과 권장 사항도 제공합니다.

주제

- 성능 요약
- 스토리지 클래스
- 성능 모드
- 처리량 모드
- Amazon EFS 성능 팁
- Amazon EFS 성능 문제 해결
- AMI 및 커널 문제 해결

성능 요약

파일 시스템 성능은 일반적으로 지연 시간, 처리량, 초당 입출력 작업 처리량(IOPS)의 차원을 사용하여 측정됩니다. 이러한 차원에서의 Amazon EFS 성능은 파일 시스템의 구성에 따라 달라집니다. 다음 구성은 Amazon EFS 파일 시스템 성능에 영향을 미칩니다.

- 파일 시스템 유형 - Regional 또는 One Zone
- 성능 모드 - 범용 또는 최대 I/O

> **Important**
>
> 최대 I/O 성능 모드는 범용 성능 모드보다 작업당 지연 시간이 더 깁니다. 더 빠른 성능을 위해 항상 범용 성능 모드를 사용하는 것이 좋습니다. 자세한 내용은 성능 모드 단원을 참조하십시오.

- 처리량 모드 - 탄력적, 프로비저닝 또는 버스팅

다음 표에는 범용 성능 모드를 사용하는 파일 시스템의 성능 사양과 파일 시스템 유형 및 처리량 모드의 가능한 다양한 조합이 요약되어 있습니다.

범용 성능 모드를 사용하는 파일 시스템의 성능 사양

스토리지 및 처리량 구성		지연 시간 [1]		최대 IOPS		최대 처리량		
파일 시스템 유형	처리량 모드	읽기 작업	쓰기 작업	읽기 작업	쓰기 작업	파일 시스템별 읽기[2]	파일 시스템별 쓰기[2]	클라이언트별 읽기/쓰기
리전	Elastic	~1밀리초(ms)	~2.7 milliseconds (ms)	900,000–2,500,000[3]	500,000[3]	초당 20~60기가바이트 (GiBps)	1~5Gibps	초당 1,500메비바이트 (MiBps)[4]
리전	Provisioned	~1ms	~2.7 ms	55,000	25,000	3~10GiBps	1~3.33GiBps	500 MiBps
리전	Bursting	~1ms	~2.7 ms	35,000	7,000	3~5Gibps	1~3Gibps	500 MiBps
One Zone	Elastic, Provisioned, Bursting	~1ms	~1.6ms	35,000	7,000	3GiBps[5]	1GiBps[5]	500 MiBps

1. 표시된 지연 시간 값은 최적의 조건에서 최상의 성능을 나타냅니다. 실제 결과는 네트워크, 워크로드 및 시스템 요인에 따라 달라질 수 있습니다.
2. 최대 읽기 및 쓰기 처리량은 AWS 리전에 따라 달라집니다. 처리량이 AWS 리전의 최대 처리량을 초과하려면 처리량 할당량을 늘려야 합니다. Amazon EFS 서비스 팀은 모든 추가 처리량 요청을 건별로 고려합니다. 승인은 워크로드 유형에 따라 달라질 수 있습니다. 할당량 증가 요청에 대해 자세히 알아보려면 Amazon EFS 할당량 섹션을 참조하세요.
3. 기본적으로 Elastic 처리량을 사용하는 파일 시스템은 접근 빈도가 낮은 데이터에 대해 최대 90,000개의 읽기 IOPS, 접근 빈도가 높은 데이터에 대해 최대 250,000개의 읽기 IOPS, 최대 50,000개의 쓰기 IOPS를 제공합니다. 워크로드에 더 많은 IOPS가 필요한 경우 이러한 수의 최대 10배 증가를 요청할 수 있습니다. 자세한 내용은 늘릴 수 있는 Amazon EFS 할당량 단원을 참조하십시오. 최대

IOPS를 달성하기 위한 추가 권장 사항이 적용됩니다. 자세한 내용은 the section called “높은 처리량과 IOPS를 요구하는 워크로드 최적화” 단원을 참조하십시오.

4. 탄력적 처리량을 사용하고 버전 2.0 이상의 Amazon EFS 클라이언트(amazon-efs-utils 버전) 또는 Amazon EFS CSI 드라이버(aws-efs-csi-driver)를 사용하여 탑재된 파일 시스템의 경우 최대 읽기 및 쓰기 결합 처리량은 1,500MiBps입니다. 다른 모든 파일 시스템의 경우 처리량 한도는 500MiBps입니다. Amazon EFS 클라이언트에 대한 자세한 내용은 Amazon EFS 클라이언트 수동 설치 섹션을 참조하세요.
5. 버스팅 처리량을 사용하는 One Zone 파일 시스템은 버스팅 처리량을 사용하는 리전 파일 시스템과 동일한 파일 시스템별 읽기 및 쓰기 처리량(최대 읽기 5GiBps, 최대 쓰기 3GiBps)을 구동할 수 있습니다.

스토리지 클래스

Amazon EFS 스토리지 클래스는 사용 사례에 따라 가장 효과적인 스토리지를 제공하도록 설계되었습니다.

- EFS Standard 스토리지 클래스에서 솔리드 스테이트 드라이브(SSD) 스토리지를 사용하여 자주 액세스하는 파일에 대해 최저 수준의 지연 시간을 제공합니다. 이 스토리지 클래스는 읽기의 경우 최소 1밀리초, 쓰기의 경우 2.7밀리초의 1바이트 지연 시간을 제공합니다.
- EFS Infrequency Access(IA) 및 EFS Archive 스토리지 클래스는 액세스되지 않는 데이터를 저장하며, 이러한 데이터는 자주 액세스되는 데이터가 요구하는 대기 시간 성능을 필요로 하지 않습니다. 이러한 스토리지 클래스는 수십 밀리초의 첫 번째 바이트 지연 시간을 제공합니다.

EFS 스토리지 클래스에 대한 자세한 내용은 the section called “EFS 스토리지 클래스” 섹션을 참조하세요.

성능 모드

Amazon EFS는 범용 및 최대 I/O라는 두 가지 성능 모드를 제공합니다.

- 범용 모드는 작업당 지연 시간이 가장 낮으며 파일 시스템의 기본 성능 모드입니다. One Zone 파일 시스템은 항상 범용 성능 모드를 사용합니다. 더 빠른 성능을 위해 항상 범용 성능 모드를 사용하는 것이 좋습니다.
- 최대 I/O 모드는 범용 모드보다 긴 지연 시간을 견딜 수 있는 고도로 병렬화된 워크로드용으로 설계된 이전 세대 성능 유형입니다. One Zone 파일 시스템 또는 탄력적 처리량을 사용하는 파일 시스템에서는 최대 I/O 모드가 지원되지 않습니다.

> **Important**
>
> 최대 I/O에서는 작업당 지연 시간이 길어지기 때문에 모든 파일 시스템에 범용 성능 모드를 사용하는 것이 좋습니다.

범용 성능 모드를 사용하는 파일 시스템에서 사용할 수 있는 IOPS 한도 내에서 워크로드가 유지되도록 하려면 `PercentIOLimit` CloudWatch 지표를 모니터링할 수 있습니다. 자세한 내용은 Amazon EFS에 대한 CloudWatch 지표 단원을 참조하십시오.

애플리케이션은 성능 모드와 관련된 한도까지 IOPS를 탄력적으로 확장할 수 있습니다. IOPS는 별도로 청구되지 않으며, IOPS는 파일 시스템의 처리량 계산에 포함됩니다. 모든 네트워크 파일 시스템(NFS) 요청은 4킬로바이트(KB) 의 처리량 또는 실제 요청 및 응답 크기 중 큰 쪽을 기준으로 계산됩니다.

처리량 모드

파일 시스템의 처리량 모드에 따라 파일 시스템에서 사용할 수 있는 처리량이 결정됩니다. Amazon EFS는 탄력적, 프로비저닝, 버스팅이라는 세 가지 처리량 모드를 제공합니다. 쓰기 처리량보다 높은 읽기 처리량을 제공할 수 있도록 읽기 처리량이 할인됩니다. 각 처리량 모드에서 사용할 수 있는 최대 처리량은 AWS 리전에 따라 달라집니다. 지역별 최대 파일 시스템 처리량에 대한 자세한 내용은 Amazon EFS 할당량을 참조하세요.

파일 시스템은 읽기 및 쓰기 처리량을 합쳐 100%를 달성할 수 있습니다. 예를 들어 파일 시스템이 읽기 처리량 한도의 33%를 사용하는 경우 그 파일 시스템은 동시에 쓰기 처리량 한도의 최대 67%까지 달성할 수 있습니다. 콘솔의 파일 시스템 세부 정보 페이지에 있는 처리량 사용률(%) 그래프에서 파일 시스템의 처리량 사용량을 모니터링할 수 있습니다. 자세한 내용은 처리량 성능 모니터링 단원을 참조하십시오.

파일 시스템의 올바른 처리량 모드 선택

파일 시스템에 적합한 처리량 모드를 선택하는 것은 워크로드의 성능 요구 사항에 따라 달라집니다.

- 탄력적 처리량(권장) - 급증하거나 예측하기 어려운 워크로드 및 성능 요구 사항이 있거나 애플리케이션에서 평균 대비 피크 처리량 비율이 5% 이하인 경우 기본 탄력적 처리량을 사용합니다. 자세한 내용은 탄력적 처리량 단원을 참조하십시오.

- 프로비저닝된 처리량 - 워크로드의 성능 요구 사항을 알고 있거나 애플리케이션에서 평균 대비 피크 처리량 비율이 5% 이상인 경우 프로비저닝된 처리량을 사용합니다. 자세한 내용은 프로비저닝된 처리량 단원을 참조하십시오.
- 버스팅 처리량 - 파일 시스템의 저장 용량에 따라 확장되는 처리량을 원할 때 버스팅 처리량을 사용합니다.

  버스팅 처리량을 사용한 후 애플리케이션에 처리량 제약이 있는 경우(예: 허용된 처리량의 80% 이상을 사용하거나 버스트 크레딧을 모두 사용한 경우), 탄력적 처리량 또는 프로비저닝된 처리량을 사용해야 합니다. 자세한 내용은 버스팅 처리량 단원을 참조하십시오.

Amazon EFS 지표에 대한 자세한 내용은 Amazon EFS에 대한 CloudWatch 지표 섹션을 참조하세요.

탄력적 처리량

탄력적 처리량을 사용하는 파일 시스템의 경우 Amazon EFS는 워크로드 활동의 요구에 맞게 처리량 성능을 자동으로 늘리거나 줄입니다. 탄력적 처리량은 워크로드가 급증하거나 예측할 수 없으며 성능 요구 사항을 예측하기 어려운 경우 또는 애플리케이션에서 평균 대비 피크 처리량(평균 대비 피크 비율)이 5% 이하인 경우에 가장 적합한 처리량 모드입니다.

탄력적 처리량을 사용하는 파일 시스템의 처리량 성능은 자동으로 확장되므로 애플리케이션 요구 사항에 맞게 처리량 용량을 지정하거나 프로비저닝할 필요가 없습니다. 탄력적 처리량을 사용하면 메타데이터 및 읽거나 쓴 데이터 양에 대해서만 비용을 지불하며, 버스트 크레딧이 발생하거나 소비되지 않습니다.

> **Note**
>
> 탄력적 처리량은 처리량에 따라 탄력적으로 확장되도록 설계되었지만 운영 모범 사례의 일환으로 CloudWatch(MeteredIOBytes) 및 사용량 알림을 사용하여 지표를 모니터링하여 적절한 거버넌스를 구현하는 것이 좋습니다. 이를 통해 최적의 리소스 사용률을 유지하고 계획된 운영 파라미터를 유지할 수 있습니다. 자세한 내용은 Amazon CloudWatch를 사용한 지표 모니터링 단원을 참조하십시오.

리전별 탄력적 처리량 한도에 대한 자세한 내용은 늘릴 수 있는 Amazon EFS 할당량 단원을 참조하세요.

프로비저닝된 처리량

프로비저닝된 처리량을 사용하면 파일 시스템의 크기 또는 버스트 크레딧 밸런스에 관계없이 파일 시스템에서 처리할 수 있는 처리량 수준을 지정할 수 있습니다. 워크로드의 성능 요구 사항을 알고 있거나 애플리케이션에서 평균 대비 피크 처리량 비율이 5% 이상인 경우 프로비저닝된 처리량을 사용합니다.

프로비저닝된 처리량을 사용하는 파일 시스템의 경우 파일 시스템에 설정된 처리량에 따라 요금이 부과됩니다. 한 달에 청구되는 처리량은 Standard 스토리지에서 파일 시스템에 포함된 기준 처리량을 초과하여 프로비저닝된 처리량을 기준으로 하며, AWS 리전의 일반적인 버스팅 기준 처리량 한도까지 초과하여 청구됩니다.

파일 시스템의 기준 처리량이 프로비저닝된 처리량을 초과하는 경우 파일 시스템에 허용되는 버스팅 처리량을 자동으로 사용합니다(해당 기준 처리량 한도까지 AWS 리전).

리전별 프로비저닝 처리량 한도에 대한 자세한 내용은 늘릴 수 있는 Amazon EFS 할당량을 참조하세요.

버스팅 처리량

파일 시스템의 스토리지 용량에 따라 처리량을 확장해야 하는 워크로드에는 버스팅 처리량 모드를 사용하는 것이 좋습니다. 버스팅 처리량을 사용하면 기본 처리량은 Standard 스토리지 클래스에서 파일 시스템의 크기에 비례하며, 스토리지의 각 GiB당 50KiBps의 비율로 계산됩니다. 버스트 크레딧은 파일 시스템이 기본 처리량 이하로 소비할 때 누적되고, 처리량이 기본 속도를 초과하면 차감됩니다.

버스트 크레딧이 사용 가능한 경우, 파일 시스템은 AWS 리전 제한 범위 내에서 표준 스토리지 기준 TiB당 최대 100MiBps(GiB당 50KiBps)의 처리량을 달성할 수 있으며, 최소 처리량은 100MiBps입니다. 버스트 크레딧을 사용할 수 없는 경우 파일 시스템은 스토리지 TiB당 최대 50MiBps(최소 1MiBps)를 구동할 수 있습니다.

리전별 버스팅 처리량에 대한 자세한 내용은 General resource quotas that cannot be changed 단원을 참조하세요.

Amazon EFS 버스트 크레딧에 대한 이해

버스팅 처리량을 사용하면 각 파일 시스템은 EFS Standard 스토리지 클래스에 저장된 파일 시스템의 크기에 따라 결정되는 기준 속도로 시간이 지남에 따라 버스트 크레딧을 지급받습니다. 기준 속도는 스토리지 테비바이트(TiB)당 50MiBps입니다(스토리지 GiB당 50KiBps와 동일). Amazon EFS는 쓰기 작업 속도의 최대 3분의 1까지 읽기 작업을 측정하므로 파일 시스템은 읽기 처리량 GiB당 최대 150KiBps 또는 쓰기 처리량 GiB당 50KiBps의 기준 속도를 제공할 수 있습니다.

파일 시스템은 기준 측정 속도로 지속적으로 처리량을 높일 수 있습니다. 파일 시스템은 비활성 상태이거나 처리량이 기준 측정 속도 이하로 떨어질 때마다 버스트 크레딧을 누적합니다. 누적된 버스트 크레딧은 파일 시스템에 처리량을 기준 속도 이상으로 끌어올릴 수 있는 기능을 제공합니다.

예를 들어 Standard 스토리지에 100GiB의 측정된 데이터가 있는 파일 시스템의 기준 처리량은 5MiBpS입니다. 24시간 동안 사용하지 않는 동안 파일 시스템은 432,000MiB 상당의 크레딧(5MiB × 86,400초 = 432,000MiB)을 획득하며, 이 크레딧을 사용하여 72분 동안 100MiBps의 속도로 버스트하는 데 사용할 수 있습니다(432,000MiB ÷ 100MiBps = 72분).

1TiB보다 큰 파일 시스템은 나머지 50%의 시간 동안 비활성 상태이면 최대 50%의 시간 동안 항상 버스트할 수 있습니다.

다음 표는 버스팅 동작의 예를 보여 줍니다.

파일 시스템 크기	버스트 처리량	기준 처리량
Standard 스토리지에 있는 100GiB의 측정된 데이터	• 하루 최대 72분 동안 읽기 전용으로 300(MiBps)까지 버스트하거나 • 하루 최대 72분 동안 쓰기 전용으로 100MiBps까지 버스트할 수 있습니다.	• 최대 15MiBPS의 읽기 전용으로 연속 구동 • 최대 5MiBPS의 쓰기 전용으로 연속 구동
Standard 스토리지에 있는 1TiB의 측정된 데이터	• 하루 12시간 동안 읽기 전용으로 300MiBps까지 버스트하거나 • 하루 12시간 동안 쓰기 전용으로 100MiBps까지 버스트할 수 있습니다.	• 150MiBPS의 읽기 전용으로 연속 구동 • 50MiBPS의 쓰기 전용으로 연속 구동
Standard 스토리지에 있는 10TiB의 측정된 데이터	• 하루 12시간 동안 읽기 전용으로 3GiBPS로 버스트하거나 • 하루 12시간 동안 쓰기 전용으로 1GiBPS로 버스트할 수 있습니다.	• 1.5GiBps의 읽기 전용으로 연속 구동 • 500MiBPS의 쓰기 전용으로 연속 구동
일반적으로 규모가 더 큰 파일 시스템	• 하루 12시간 동안 스토리지 TiB당 읽기 전용으로 300MiBps까지 버스트하거나	• 스토리지 TiB당 150MiBps의 읽기 전용으로 연속 구동 • 스토리지 TiB당 50MiBps의 쓰기 전용으로 연속 구동

파일 시스템 크기	버스트 처리량	기준 처리량
	• 하루 12시간 동안 스토리지 TiB당 쓰기 전용으로 100MiBps까지 버스트할 수 있습니다.	

> **Note**
>
> Amazon EFS는 기준 속도가 더 낮더라도 모든 파일 시스템에 1MiBPS의 측정된 처리량을 제공합니다.
> 기준 속도 및 버스트 속도를 결정할 때 사용되는 파일 시스템 크기는 DescribeFileSystems API 작업을 통해 사용 가능한 `ValueInStandard`로 측정된 크기와 동일합니다.
> 파일 시스템이 획득할 수 있는 최대 크레딧 잔고는 1TiB보다 작은 파일 시스템의 경우 2.1TiB이고, 1TiB보다 큰 파일 시스템의 경우 저장된 TiB당 2.1TiB입니다. 즉, 파일 시스템은 최대 12시간 동안 연속으로 버스트하는 데 충분한 크레딧을 축적할 수 있습니다.

처리량 전환 및 프로비저닝량 변경에 대한 제한 사항

기존 파일 시스템의 처리량 모드를 전환하고 처리량을 변경할 수 있습니다. 하지만 처리량 모드를 프로비저닝된 처리량으로 전환하거나 프로비저닝된 처리량을 변경한 후에는 24시간 동안 다음 작업이 제한됩니다.

- 프로비저닝된 처리량 모드에서 탄력적 또는 버스팅 처리량 모드로 전환
- 프로비저닝된 처리량 감소

Amazon EFS 성능 팁

Amazon EFS를 사용할 때는 다음 성능 팁을 명심하세요.

평균 I/O 크기

Amazon EFS의 분산 특성 덕분에 높은 수준의 가용성, 내구성 및 확장성을 구현할 수 있습니다. 이러한 분산 아키텍처로 인해 각 파일 작업에서 약간의 지연 오버헤드가 생기는데, 이렇게 작업당 지연 시간이 있으므로 평균 I/O 크기가 늘어남에 따라 전체 처리량도 대개 증가합니다. 대량의 데이터에 대해 오버헤드가 분할 사용되기 때문입니다.

높은 처리량과 IOPS를 요구하는 워크로드 최적화

높은 처리량과 IOPS가 필요한 워크로드의 경우 범용 성능 모드 및 탄력적 처리량으로 구성된 Regional 파일 시스템을 사용합니다.

> **Note**
>
> 자주 액세스하는 데이터에 대해 최대 읽기 IOPS를 달성하려면 파일 시스템에서 탄력적 처리량을 사용해야 합니다.

최고 수준의 성능을 달성하려면 다음과 같이 애플리케이션 또는 워크로드를 구성하여 병렬화를 활용해야 합니다.

1. 최소한 사용 중인 클라이언트 수와 동일한 수의 디렉터리를 사용하여 모든 클라이언트와 디렉터리에 워크로드를 균등하게 분배하십시오.
2. 개별 스레드를 별개의 데이터 세트 또는 파일에 맞게 조정하여 경합을 최소화합니다.
3. 단일 탑재 대상에서 클라이언트당 최소 64개의 스레드를 사용하여 10개 이상의 NFS 클라이언트에 워크로드를 분산합니다.

동시 연결

최대 수천 개의 Amazon EC2 및 기타 AWS 컴퓨팅 인스턴스에 Amazon EFS 파일 시스템을 동시에 탑재할 수 있습니다. 더 많은 인스턴스에서 병렬화할 수 있는 애플리케이션인 경우, 인스턴스 간 집계를 통해 파일 시스템의 처리량을 더 높은 수준으로 끌어올릴 수 있습니다.

요청 모델

파일 시스템에 대한 비동기 쓰기를 활성화하면, 보류 중인 쓰기 작업은 Amazon EC2 인스턴스에서 버퍼링된 다음 Amazon EFS에 비동기식으로 기록됩니다. 비동기 쓰기는 일반적으로 지연 시간이 더 짧습니다. 비동기 쓰기를 수행하는 경우 커널은 캐싱에 추가 메모리를 사용합니다.

비동기 쓰기를 활성화한 파일 시스템 또는 캐시 우회 옵션(예: `O_DIRECT`)을 사용하여 파일을 여는 파일 시스템에서는 Amazon EFS에 비동기 요청을 생성합니다. 모든 작업은 클라이언트와 Amazon EFS를 왕복합니다.

> **Note**
>
> 선택한 요청 모델은 일관성(여러 Amazon EC2 인스턴스를 사용하는 경우)과 속도를 절충합니다. 동기 쓰기를 사용하면 다음 요청을 처리하기 전에 각 쓰기 요청 트랜잭션을 완료하여 데이터 일관성을 높일 수 있습니다. 비동기 쓰기를 사용하면 보류 중인 쓰기 작업을 버퍼링하여 처리량을 높일 수 있습니다.

NFS 클라이언트 탑재 설정

EFS 파일 시스템 탑재과 Linux의 탑재 고려 사항에 설명된 대로 권장 탑재 옵션을 사용하고 있는지 확인하세요.

Amazon EC2 인스턴스에 파일 시스템을 탑재하는 경우, Amazon EFS에서는 네트워크 파일 시스템 버전 4.0 및 4.1(NFSv4) 프로토콜을 지원합니다. NFSv4.1은 NFSv4.0(초당 파일 1,000개 미만)에 비해 소규모 파일 병렬 읽기 작업(초당 파일 10,000개 이상)에서 더 나은 성능을 제공합니다. macOS Big Sur를 실행하는 Amazon EC2 macOS 인스턴스의 경우 NFSv4.0만 지원됩니다.

다음 탑재 옵션은 사용하지 마세요.

- `noac`, `actimeo=0`, `acregmax=0`, `acdirmax=0` - 이 옵션은 성능에 매우 큰 영향을 미치는 속성 캐시를 비활성화합니다.
- `lookupcache=pos`, `lookupcache=none` - 이 옵션은 성능에 매우 큰 영향을 미치는 파일 이름 조회 캐시를 비활성화합니다.
- `fsc`- 이 옵션은 로컬 파일 캐싱을 활성화하지만 NFS 캐시 일관성을 변경하지 않으며 지연 시간을 줄이지 않습니다.

> **Note**
>
> 파일 시스템을 탑재할 때 NFS 클라이언트의 읽기 및 쓰기 버퍼 크기를 1MB로 늘리는 것을 고려 해 보세요.

소규모 파일 성능 최적화

파일 다시 열기를 최소화하고, 병렬 처리를 증가시키고, 가능한 경우 참조 파일을 번들링하여 소규모 파일의 성능을 개선할 수 있습니다.

- 서버로의 왕복 횟수를 최소화하세요.

  나중에 워크플로에서 필요할 경우 파일을 불필요하게 닫지 마세요. 파일 설명자를 열어 두면 캐시에 있는 로컬 사본에 직접 액세스할 수 있습니다. 파일 열기, 닫기 및 메타데이터 작업은 일반적으로 비동기적으로 또는 파이프라인을 통해 수행할 수 없습니다.

  소규모 파일을 읽거나 쓸 때는 두 번의 추가 왕복이 중요합니다.

  각 왕복(파일 열기, 파일 닫기)에는 메가바이트 단위의 대량 데이터를 읽거나 쓰는 시간만큼 걸릴 수 있습니다. 컴퓨팅 작업을 시작할 때 입력 또는 출력 파일을 한 번 열고 전체 작업 기간 동안 열어 두는 것이 더 효율적입니다.
- 병렬 처리를 사용하면 왕복 시간이 미치는 영향을 줄일 수 있습니다.
- 참조 파일을 `.zip` 파일로 번들링합니다. 일부 애플리케이션은 크기가 작고 대부분 읽기 전용인 참조 파일을 대량으로 사용합니다. 이러한 파일을 하나의 `.zip` 파일로 번들링하면 한 번의 열고 닫기 왕복으로 여러 파일을 읽을 수 있습니다.

  `.zip` 형식을 사용하면 개별 파일에 무작위로 액세스할 수 있습니다.

디렉터리 성능 최적화

동시에 수정되는 매우 큰 디렉터리(10만 개 이상의 파일)에 대해 목록(ls) 작업을 수행하는 경우 Linux NFS 클라이언트가 응답을 반환하지 않고 중단될 수 있습니다. 이 문제는 Amazon Linux 2 커널 4.14, 5.4, 5.10으로 이식된 커널 5.11에서 수정되었습니다.

가능하면 파일 시스템의 디렉터리 수를 10,000개 미만으로 유지하는 것이 좋습니다. 중첩된 하위 디렉터리를 최대한 많이 사용하세요.

디렉터리를 목록을 작성할 때 필요하지 않은 파일 특성은 디렉터리 자체에 저장되지 않으므로 가져오지 마세요.

NFS read_ahead_kb 크기 최적화

NFS `read_ahead_kb` 속성은 Linux 커널이 순차적 읽기 작업 중에 미리 읽거나 이리 가져올 킬로바이트 수를 정의합니다.

Linux 커널 버전 5.4.* 이전의 경우 `read_ahead_kb` 값은 값 `rsize`(탑재 옵션에 설정된 클라이언트 구성 읽기 버퍼 크기)에 `NFS_MAX_READAHEAD`를 곱하여 설정됩니다. 권장 탑재 옵션을 사용할 경우 이 공식은 `read_ahead_kb`을 15MB로 설정합니다.

> **Note**
>
> 리눅스 커널 버전 5.4.*부터 리눅스 NFS 클라이언트는 read_ahead_kb 기본값인 128KB를 사용합니다. 이 값을 15MB로 늘리는 것이 좋습니다.

amazon-efs-utils 버전 1.33.2 이상에서 사용할 수 있는 Amazon EFS 탑재 도우미는 파일 시스템을 탑재한 후 자동으로 read_ahead_kb 값을 15* rsize 또는 15MB로 수정합니다.

Linux 커널 5.4 이상의 경우 탑재 도우미를 사용하여 파일 시스템을 탑재하지 않는 경우 성능 향상을 위해 수동으로 read_ahead_kb를 15MB로 설정하는 것이 좋습니다. 파일 시스템을 탑재한 후 다음 명령을 사용하여 read_ahead_kb 값을 재설정할 수 있습니다. 이 명령을 사용하기 전에 다음 값을 교체하세요.

- *read-ahead-value-kb*를 원하는 크기(킬로바이트)로 교체합니다.
- *efs-mount-point*를 파일 시스템의 탑재 포인트를 교체합니다.

```
device_number=$(stat -c '%d' efs-mount-point)
((major = ($device_number & 0xFFF00) >> 8))
((minor = ($device_number & 0xFF) | (($device_number >> 12) & 0xFFF00)))
sudo bash -c "echo read-ahead-value-kb > /sys/class/bdi/$major:$minor/read_ahead_kb"
```

다음 예제는 read_ahead_kb 크기를 15MB로 설정합니다.

```
device_number=$(stat -c '%d' efs)
((major = ($device_number & 0xFFF00) >> 8))
((minor = ($device_number & 0xFF) | (($device_number >> 12) & 0xFFF00)))
sudo bash -c "echo 15000 > /sys/class/bdi/$major:$minor/read_ahead_kb"
```

Amazon EFS 성능 문제 해결

일반적으로 Amazon EFS에 문제가 발생해 문제를 해결하는 경우 최신 Linux 커널을 사용하고 있는지 확인하세요. 엔터프라이즈 Linux 배포판을 사용하는 경우 다음을 권장합니다.

- Amazon Linux 2(커널 4.3 이상)
- Amazon Linux 2015.09 이상
- RHEL 7.3 이상

- Ubuntu 16.04 버전 모두
- 커널 3.13.0-83 이상의 Ubuntu 14.04
- SLES 12 Sp2 이상

다른 배포판이나 사용자 지정 커널을 사용하고 있는 경우 커널 버전 4.3 이상을 권장합니다.

> **Note**
>
> RHEL 6.9는 병렬로 많은 파일을 열 때 성능 불량으로 인해 특정 워크로드에 대해서는 차선일 수 있습니다.

주제

- EFS 파일 시스템을 생성할 수 없음
- NFS 파일 시스템에서 허용된 파일에 대한 액세스가 거부되었습니다.
- Amazon EFS 콘솔에 액세스할 때 발생하는 오류
- Amazon EC2 인스턴스 중단
- 대용량 데이터를 쓰는 애플리케이션이 중단됨
- 병렬로 많은 파일을 열 때 성능 불량
- 사용자 정의 NFS 설정으로 인해 쓰기 지연 발생
- Oracle Recovery Manager로 백업을 생성하는 속도가 느림

EFS 파일 시스템을 생성할 수 없음

EFS 파일 시스템 생성 요청이 실패하고 다음 메시지가 표시됩니다.

```
User: arn:aws:iam::111122223333:user/username is not authorized to
perform: elasticfilesystem:CreateFileSystem on the specified resource.
```

취할 조치

AWS Identity and Access Management (IAM) 정책을 확인하여 지정된 리소스 조건으로 EFS 파일 시스템을 생성할 권한이 있는지 확인합니다. 자세한 내용은 Amazon EFS의 자격 증명 및 액세스 관리 단원을 참조하십시오.

NFS 파일 시스템에서 허용된 파일에 대한 액세스가 거부되었습니다.

16개 이상의 액세스 그룹 ID(GID)가 할당된 사용자가 NFS 파일 시스템에서 작업을 수행하려고 하면 파일 시스템에서 허용된 파일에 대한 액세스가 거부될 수 있습니다. 이 문제는 NFS 프로토콜이 사용자당 최대 16개의 GID를 지원하고 RFC 5531에 정의된 대로 NFS 클라이언트 요청에서 추가 GID가 잘리기 때문에 발생합니다.

취할 조치

각 사용자에게 16개 이하의 액세스 그룹(GID)이 할당되지 않도록 NFS 사용자 및 그룹 매핑을 재구성하세요.

Amazon EFS 콘솔에 액세스할 때 발생하는 오류

이 섹션에서는 Amazon EFS 관리 콘솔에 액세스할 때 사용자가 겪을 수 있는 오류를 설명합니다.

`ec2:DescribeVPCs`에 대한 보안 인증 정보를 인증하는 중 오류가 발생했습니다.

Amazon EFS 콘솔에 액세스할 때 다음 오류 메시지가 표시됩니다.

```
AuthFailure: An error occurred authenticating your credentials for ec2:DescribeVPCs.
```

이 오류는 로그인 보안 인증이 Amazon EC2 서비스를 통해 성공적으로 인증되지 않았음을 나타냅니다. Amazon EFS 콘솔은 사용자가 선택한 VPC에서 EFS 파일 시스템을 생성할 때 사용자 대신 Amazon EC2 서비스를 직접 호출합니다.

취할 조치

Amazon EFS 콘솔에 액세스하는 클라이언트의 시간이 올바르게 설정되었는지 확인하세요.

Amazon EC2 인스턴스 중단

파일 시스템을 먼저 탑재 해제하지 않고 파일 시스템 탑재 대상을 삭제했기 때문에 Amazon EC2 인스턴스가 중단될 수 있습니다.

취할 조치

파일 시스템 탑재 대상을 삭제하기 전에 파일 시스템의 탑재를 해제합니다. Amazon EFS 파일 시스템 탑재 해제에 대한 자세한 내용은 파일 시스템 탑재 해제 섹션을 참조하세요.

대용량 데이터를 쓰는 애플리케이션이 중단됨

Amazon EFS에 대용량 데이터를 쓰는 애플리케이션이 중단되어 인스턴스가 재부팅됩니다.

취할 조치

애플리케이션이 Amazon EFS에 데이터를 전부 쓰는 데 시간이 너무 오래 걸리는 경우, 프로세스가 응답 없음 상태로 보이기 때문에 Linux가 재부팅될 수 있습니다. 이러한 동작은 두 가지 커널 구성 파라미터, 즉 `kernel.hung_task_panic` 및 `kernel.hung_task_timeout_secs`로 정의합니다.

다음 예에서는 `ps` 명령과 `D`를 사용해 인스턴스가 재부팅되기 전에 중단 프로세스의 상태를 보고함으로써 이 프로세스가 I/O 대기 중임을 알립니다.

```
$ ps aux | grep large_io.py
root 33253 0.5 0.0 126652 5020 pts/3 D+ 18:22 0:00 python large_io.py
/efs/large_file
```

재부팅되지 않도록 하려면 제한 시간을 늘리거나 중단 작업이 감지되면 커널 패닉을 비활성화하도록 합니다. 다음 명령은 대부분의 Linux 시스템에서 중단 작업 커널 패닉을 비활성화합니다.

```
$ sudo sysctl -w kernel.hung_task_panic=0
```

병렬로 많은 파일을 열 때 성능 불량

애플리케이션이 여러 파일을 병렬로 열 때 I/O 병렬화를 통해 예상된 성능 향상을 보이지 않습니다.

취할 조치

이 문제는 네트워크 파일 시스템 버전 4(NFSv4) 클라이언트 및 NFSv4.1을 사용하는 RHEL 6 클라이언트가 NFS 열기 및 닫기 작업을 직렬화하기 때문에 발생합니다. NFS 프로토콜 버전 4.1과 제안한 Linux 배포판 중 하나를 사용하면 이 문제가 발생하지 않습니다.

NFSv4.1을 사용할 수 없는 경우에는 Linux NFSv4.0 클라이언트가 사용자 ID 및 그룹 ID별로 열기 및 닫기 요청을 직렬화합니다. 이러한 직렬화는 여러 프로세스 또는 여러 스레드가 동시에 요청을 발급하는 경우에도 발생합니다. 모든 ID가 일치하는 경우 클라이언트는 NFS 서버로 열기 또는 닫기 작업을 한 번에 하나만 보낼 수 있습니다. 이러한 문제를 해결하려면 다음 중 한 가지 방법을 시도하면 됩니다.

- 동일한 Amazon EC2 인스턴스의 다른 사용자 ID에서 각 프로세스를 실행할 수 있습니다.
- 모든 열기 요청에서 사용자 ID를 동일하게 남겨 두고 대신 그룹 ID 세트를 수정할 수 있습니다.

- 별도의 Amazon EC2 인스턴스에서 각 프로세스를 실행할 수 있습니다.

사용자 정의 NFS 설정으로 인해 쓰기 지연 발생

사용자 정의 NFS 클라이언트 설정이 있고, Amazon EC2 인스턴스가 다른 Amazon EC2 인스턴스의 파일 시스템에 대해 수행되는 쓰기 작업을 확인하는 데 최대 3초가 걸립니다.

취할 조치

이 문제가 발생하면 다음 중 한 가지 방법으로 해결할 수 있습니다.

- 데이터를 읽는 Amazon EC2 인스턴스의 NFS 클라이언트에 속성 캐싱이 활성화되어 있으면 파일 시스템을 탑재 해제합니다. 그런 다음 `noac` 옵션을 사용해 탑재 해제해 속성 캐싱을 비활성화합니다. NFSv4.1의 속성 캐싱은 기본적으로 활성화되어 있습니다.

  > **Note**
  >
  > 클라이언트 측 캐싱을 비활성화하면 애플리케이션의 성능이 잠재적으로 저하될 수 있습니다.

- 또한 NFS 프로시저와 호환되는 프로그래밍 언어를 사용하여 요청 시 속성 캐시를 지울 수도 있습니다. 이렇게 하려면 읽기 요청 직전에 `ACCESS` 프로시저 요청을 보낼 수 있습니다.

  예를 들어 Python 프로그래밍 언어를 사용하여 다음 호출을 구성할 수 있습니다.

```
# Does an NFS ACCESS procedure request to clear the attribute cache, given a path to
 the file
import os
os.access(path, os.W_OK)
```

Oracle Recovery Manager로 백업을 생성하는 속도가 느림

Oracle Recovery Manager가 120초 동안 중지된 후 백업 작업을 시작하는 경우 Oracle Recovery Manager를 사용하여 백업을 생성하는 속도가 느릴 수 있습니다.

취할 조치

이런 문제에 직면한 경우 Oracle 도움말 센터의 NFS의 직접 NFS 클라이언트 제어 활성화 및 비활성화에 설명된 것처럼 Oracle Direct NFS를 비활성화합니다.

> **Note**
>
> Amazon EFS는 Oralce Direct NFS를 지원하지 않습니다.

AMI 및 커널 문제 해결

아래에는 Amazon EC2 인스턴스에서 Amazon EFS를 사용하는 경우 특정 Amazon Machine Image(AMI) 또는 커널 버전과 관련된 문제 해결 정보가 나와 있습니다.

주제

- 소유권을 변경할 수 없음
- 클라이언트 버그로 인해 파일 시스템이 계속해서 작업을 반복함
- 교착 상태에 빠진 클라이언트
- 큰 디렉터리의 파일을 나열하는 데 시간이 오래 걸림

소유권을 변경할 수 없음

Linux chown 명령을 사용하여 파일/디렉터리의 소유권을 변경할 수 없습니다.

이 버그가 있는 커널 버전

2.6.32

취할 조치

다음을 수행하여 이런 오류를 해결할 수 있습니다.

- EFS 루트 디렉터리의 소유권을 변경하는 데 필요한 1회 설정 단계에 대해 chown을 수행하면 최신 커널을 실행하는 인스턴스에서 chown 명령을 실행할 수 있습니다. 예를 들어, Amazon Linux 최신 버전을 사용합니다.
- chown이 프로덕션 워크플로의 일부인 경우 chown을 사용할 수 있도록 커널 버전을 업데이트해야 합니다.

클라이언트 버그로 인해 파일 시스템이 계속해서 작업을 반복함

클라이언트 버그로 인해 파일 시스템이 계속해서 작업을 반복합니다.

취할 조치

클라이언트 소프트웨어를 최신 버전으로 업데이트합니다.

교착 상태에 빠진 클라이언트

클라이언트가 교착 상태에 빠집니다.

이 버그가 있는 커널 버전

- 커널 Linux 3.10.0-229.20.1.el7.x86_64가 설치된 CentOS-7
- 커널 Linux 4.2.0-18-generic이 설치된 Ubuntu 15.10

취할 조치

다음 중 하나를 수행하세요.

- 최신 커널 버전으로 업그레이드합니다. CentOS-7의 경우 커널 버전 Linux 3.10.0-327 이상에 수정 사항이 포함되어 있습니다.
- 이전 커널 버전으로 다운그레이드합니다.

큰 디렉터리의 파일을 나열하는 데 시간이 오래 걸림

이 문제는 NFS 클라이언트가 나열 작업을 완료하기 위해 디렉터리를 반복해서 오가는 동안 디렉터리가 변경되는 경우 발생할 수 있습니다. 이렇게 반복하는 중 디렉터리의 콘텐츠가 변경되었음이 감지될 때마다 NFS 클라이언트는 처음부터 다시 반복하기 시작합니다. 따라서 파일 변경이 빈번하게 일어나는 큰 디렉터리에서 ls 명령을 완료하는 데 시간이 오래 걸릴 수 있습니다.

이 버그가 있는 커널 버전

CentOS 및 RHEL 커널 버전 2.6.32-696.el6 이하

취할 조치

이 문제를 해결하려면 최신 커널 버전으로 업그레이드하세요.

Amazon EFS에서 데이터 백업 및 복제

데이터를 보호하기 위해 Amazon EFS는 EFS 파일 시스템을 자동으로 백업합니다. 복원력 및 데이터 보호를 강화하기 위해 EFS 파일 시스템을 AWS 리전에 복제할 수 있습니다. EFS 파일 시스템을 백업하고 복제하면 EFS 파일 시스템 데이터에 문제가 발생할 경우에도 지속적인 운영 또는 서비스를 제공할 수 있습니다. 예를 들어, 데이터 손상 또는 데이터 손실이 발생한 경우입니다.

주제

- EFS 파일 시스템 백업
- EFS 파일 시스템 복제

EFS 파일 시스템 백업

Amazon EFS는 기본적으로 Amazon EFS의 데이터를 보호하기 위해 백업 정책을 생성하고 관리하는 데 사용할 수 있는 AWS Backup완전 관리형 정책 기반 서비스인와 통합됩니다. Amazon EFS 콘솔을 사용하여 생성하는 파일 시스템은 AWS Backup 기본적으로에 의해 자동으로 백업됩니다. CLI 또는 API를 사용하여 파일 시스템을 생성하는 경우 One Zone 파일 시스템에만 자동 백업이 기본적으로 활성화됩니다.

Amazon EFS AWS Backup 용를 사용하여 다음 작업을 수행할 수 있습니다.

- 백업 계획을 구성하여 자동 백업 예약 및 보존을 관리합니다. 백업 빈도, 백업 시기, 백업 보관 기간, 백업의 수명 주기 정책을 지정할 수 있습니다.
- Amazon EFS 데이터의 백업을 복원합니다. 파일 시스템 데이터는 새 파일 시스템이나 기존 파일 시스템으로 복원할 수 있습니다. 전체 복원을 수행할지, 항목 수준 복원을 수행할지 여부를 선택할 수도 있습니다.

사용에 대한 자세한 내용은 AWS Backup 개발자 안내서의 시작하기 AWS Backup를 AWS Backup참조하세요.

주제

- 에서 Amazon EFS를 AWS Backup 사용하는 방법
- 필수 IAM 권한
- 백업 성능
- EFS 파일 시스템의 자동 백업 관리

에서 Amazon EFS를 AWS Backup 사용하는 방법

Amazon EFS 콘솔을 사용하여 생성하는 파일 시스템은 AWS Backup 기본적으로 자동으로 백업됩니다. CLI 또는 API를 사용하여 파일 시스템을 생성하는 경우 One Zone 파일 시스템에만 자동 백업이 기본적으로 활성화됩니다. AWS CLI 또는 API를 사용하여 EFS 파일 시스템을 생성한 후 자동 백업을 켤 수 있습니다. 기본 EFS 백업 계획은 자동 백업, 즉 보존 기간이 35일인 일일 백업에 AWS Backup 권장되는 설정을 사용합니다. 기본 EFS 백업 계획을 사용하여 생성된 백업은 사용자를 대신하여 Amazon EFS가 생성하는 기본 EFS 백업 볼트에 저장됩니다. 기본 백업 계획 및 백업 볼트는 삭제할 수 없습니다.

EFS 파일 시스템의 모든 데이터는 데이터의 스토리지 클래스에 관계없이 백업됩니다. Lifecycle Management가 활성화되고 Infrequent Access(IA) 또는 Archive 스토리지 클래스에 데이터가 있는 EFS 파일 시스템을 백업할 때는 데이터 액세스 요금이 부과되지 않습니다. 복구 시점을 복원하면 모든 파일이 Standard 스토리지 클래스에 복원됩니다.

증분 백업

AWS Backup 는 EFS 파일 시스템의 증분 백업을 수행합니다. 초기 백업 중에는 전체 파일 시스템의 복사본이 생성됩니다. 해당 파일 시스템의 후속 백업 중에는 변경, 추가 또는 제거된 파일 및 디렉터리만 복사됩니다. 각 증분 백업에서는 전체 복원을 허용하는 데 필요한 참조 데이터를 AWS Backup 유지합니다. 이 접근 방법을 사용하면 백업을 완료하는 데 필요한 시간이 최소화되며 데이터를 복제하지 않으므로 스토리지 비용이 절약됩니다.

백업 일관성

Amazon EFS는 가용성이 뛰어나도록 설계되었습니다. AWS Backup에서 백업이 진행되는 동안 EFS 파일 시스템에 액세스하고 이를 수정할 수 있습니다. 그러나 백업이 수행되는 동안 파일 시스템을 수정하면 중복된 데이터, 편향된 데이터 또는 제외된 데이터 등의 불일치가 발생할 수 있습니다. 이러한 수정 사항에는 쓰기, 이름 변경, 이동 또는 삭제 작업이 포함됩니다. 일관된 백업을 위해서는 백업 프로세스가 진행되는 동안 파일 시스템을 수정하는 애플리케이션 또는 프로세스를 일시 중지하는 것이 좋습니다. 또는 파일 시스템이 수정되지 않는 기간에 백업이 수행되도록 예약합니다.

백업 완료 창

원할 경우 백업의 완료 기간을 지정할 수 있습니다. 이 기간은 백업을 완료해야 하는 기간을 정의합니다. 완료 기간을 지정하는 경우 파일 시스템의 크기 및 구조, 예상 성능을 반드시 고려해야 합니다. 이렇게 하면 기간 중에 백업을 완료할 수 있습니다.

지정된 기간 중에 완료되지 않은 백업은 미완료 상태로 플래그가 지정됩니다. 예약된 다음 백업 중에는 중단한 지점에서 AWS Backup 재개합니다. AWS Backup 관리 콘솔에서 모든 백업의 상태를 볼 수 있습니다.

백업 완료 기간의 모범 사례

EFS 파일 시스템에 대한 백업 완료 기간을 구성할 때는 안정적인 백업 작업을 보장하기 위해 다음 모범 사례를 고려하세요.

- 백업 기간 모니터링 - 백업이 장기간(예: 20일 이상)이 소요되는 경우 완료 기간을 3일로 줄이는 것이 좋습니다. 이 접근 방식을 사용하면 AWS Backup 가 단일 확장 창에서 전체 백업을 완료하려고 하지 않고 더 짧은 부분 백업을 여러 번 수행할 수 있습니다.
- 파일 시스템 특성 고려 - 수백만 개의 파일 또는 중요한 데이터 볼륨이 있는 대규모 파일 시스템은 완료 기간이 짧으면 도움이 될 수 있습니다. 3일 기간 권장 사항은 백업 작업이 성공적으로 완료되고 필요한 경우 효율적으로 재개되도록 하는 데 도움이 됩니다.
- 증분 진행 계획 - 더 짧은 완료 기간을 사용하는 경우는 다음 예약된 백업 중에 중단된 위치에서 AWS Backup 자동으로 재개됩니다. 이 증분 접근 방식은 하나의 긴 창에서 모든 것을 완료하려고 시도하는 것보다 대규모 파일 시스템에서 더 안정적일 수 있습니다.
- 백업 전략 테스트 - 프로덕션 환경에서 백업 구성에 의존하기 전에 특정 파일 시스템 크기 및 특성으로 백업 구성을 테스트하여 사용 사례에 가장 적합한 완료 기간을 결정합니다.

> **Note**
>
> 완료 기간이 짧다고 해서 대용량 파일 시스템을 처리하는 기능이 제한 AWS Backup되는 것은 아닙니다. 대신 다양한 파일 시스템 크기 및 네트워크 조건에 맞게 조정할 수 있는 보다 예측 가능하고 관리 가능한 백업 프로세스를 제공합니다.

온디맨드 백업

를 사용하면 단일 리소스를 백업 볼트에 온디맨드로 저장할 AWS Backup수 있습니다. 예약된 백업과 달리 백업 계획을 생성하여 온디맨드 백업을 시작할 필요가 없습니다. 수명 주기를 백업에 계속 할당하여 복구 시점을 콜드 스토리지 티어로 자동 이동하고 삭제할 시기를 명시할 수 있습니다.

또한는 최신 웜 백업에 더 이상 존재하지 않는 데이터에 대해서만 데이터를 콜드 스토리지로 AWS Backup 자동 전환합니다. 백업을 생성할 때 파일 시스템에 100개의 파일이 있었고 백업을 생성한 다음 날에 2개의 파일을 삭제한 경우를 예로 들어보겠습니다(100개 파일 - 2개 파일 = 둘째 날 98개 파일).

데이터를 콜드 스토리지로 전환하면 삭제된 파일 2개만 콜드 스토리지로 이동하고 나머지 98개 파일은 웜 스토리지로 청구됩니다.

동시 백업

AWS Backup 는 백업을 리소스당 하나의 동시 백업으로 제한합니다. 따라서 백업 작업이 이미 진행 중인 경우에는 예약된 백업 또는 온디맨드 백업이 실패할 수 있습니다. AWS Backup 한도에 대한 자세한 내용은 AWS Backup 개발자 설명서에서 AWS Backup 할당량을 참조하세요.

백업 삭제

기본 EFS 백업 볼트 액세스 정책은 복구 지점 삭제를 거부하도록 설정되어 있습니다. EFS 파일 시스템의 기존 백업을 삭제하려면 볼트 액세스 정책을 변경해야 합니다. 볼트 액세스 정책을 수정하지 않고 EFS 복구 시점을 삭제하려고 하면 다음 오류 메시지가 나타납니다.

```
"Access Denied: Insufficient privileges to perform this action. Please consult with the
 account administrator for necessary permissions."
```

기본 백업 볼트 액세스 정책을 편집하려면 정책을 편집할 권한이 있어야 합니다. 자세한 내용은 IAM 사용 설명서의 모든 IAM 작업 허용(관리자 액세스) 을 참조하세요.

필수 IAM 권한

AWS Backup 는 계정에서 사용자를 대신하여 서비스 연결 역할을 생성합니다. 이 역할에는 Amazon EFS 백업을 수행하는 데 필요한 권한이 있습니다.

`elasticfilesystem:backup` 및 `elasticfilesystem:restore` 작업을 사용하여 IAM 엔터티(사용자, 그룹, 역할 등)가 EFS 파일 시스템에 대한 백업을 생성하거나 복원하도록 허용하거나 거부합니다. 파일 시스템 정책 또는 자격 증명 기반 IAM 정책에서 이러한 작업을 사용할 수 있습니다. 자세한 내용은 Amazon EFS의 자격 증명 및 액세스 관리 및 IAM을 사용하여 파일 시스템에 대한 액세스 제어 섹션을 참조하세요.

백업 성능

일반적으로를 사용하면 다음과 같은 백업 및 복원 속도를 기대할 수 있습니다 AWS Backup. 대용량 파일 또는 디렉터리가 포함된 워크로드 등 일부 워크로드의 경우 속도가 더 낮을 수 있습니다.

- 백업 속도는 초당 2,000개 파일 또는 초당 400메가바이트(MBps) 중 더 느린 값입니다.
- 복원 속도는 초당 1,500개 또는 200MBps 중 더 느린 값입니다.

에서 백업 작업의 최대 기간은 30일 AWS Backup 입니다.

AWS Backup 를 사용하면 누적된 버스트 크레딧이 소비되지 않으며 범용 성능 모드 파일 작업 제한에 포함되지 않습니다. 자세한 내용은 Amazon EFS 파일 시스템 할당량 단원을 참조하십시오.

EFS 파일 시스템의 자동 백업 관리

Amazon EFS 콘솔을 사용하여 파일 시스템을 생성하면 자동 백업이 기본적으로 활성화됩니다. AWS CLI 또는 API를 사용하여 파일 시스템을 생성한 후 자동 백업을 켤 수 있습니다.

AWS Backup 콘솔을 사용하여 기본 백업 계획 설정을 편집할 수 있습니다. 자세한 정보는 AWS Backup 개발자 안내서에서 백업 계획 관리를 참조하세요. AWS Backup 콘솔을 사용하여 모든 자동 백업을 확인하고 기본 EFS 백업 계획 설정을 편집할 수 있습니다.

Amazon EFS는 자동 백업이 활성화된 경우 값이 `enabled`인 `aws:elasticfilesystem:default-backup` 시스템 태그 키를 EFS 파일 시스템에 적용합니다.

파일 시스템을 생성한 후 콘솔, AWS CLI또는 EFS API를 사용하여 자동 백업을 켜거나 끌 수 있습니다.

콘솔 사용

1. Amazon Elastic File System 콘솔(https://console.aws.amazon.com/efs/)을 엽니다.
2. 파일 시스템 페이지에서 자동 백업을 켜거나 끌 파일 시스템을 선택하고 파일 시스템 세부 정보 페이지를 표시합니다.
3. 일반 설정 탭에서 편집을 선택하세요.
4.
   - 자동 백업을 켜려면 자동 백업 사용을 선택합니다.
   - 자동 백업을 끄려면 자동 백업 사용을 선택 해제합니다.
5. 변경 사항 저장을 선택합니다.

사용 AWS CLI

- `put-backup-policy` CLI 명령(해당 API 작업은 PutBackupPolicy)을 사용하여 기존 파일 시스템에 대해 자동 백업을 켜거나 끕니다.
  - 다음 명령어를 사용하여 자동 백업을 켭니다.

```
$ aws efs put-backup-policy --file-system-id fs-01234567 \
```

```
--backup-policy Status="ENABLED"
```

Amazon EFS는 새 백업 정책으로 응답합니다.

```
{
   "BackupPolicy": {
      "Status": "ENABLING"
   }
}
```

- 다음 명령어를 사용하여 자동 백업을 끕니다.

```
$ aws efs put-backup-policy --file-system-id fs-01234567 \
--backup-policy Status="DISABLED"
```

Amazon EFS는 새 백업 정책으로 응답합니다.

```
{
   "BackupPolicy": {
      "Status": "DISABLING"
   }
}
```

EFS 파일 시스템 복제

복원력 및 데이터 보호를 확장하기 위해 EFS 파일 시스템을 AWS 리전에 복제할 수 있습니다. EFS 파일 시스템에서 복제를 활성화하면 Amazon EFS는 소스 파일 시스템의 데이터와 메타데이터를 대상 파일 시스템에 자동으로 투명하게 복제합니다. 재해가 발생하거나 게임데이 연습을 수행할 때 복제 파일 시스템으로 장애 조치할 수 있습니다. 작업을 재개하려면 기본 파일 시스템으로 다시 장애 복구할 수 있습니다.

대상 파일 시스템을 생성하고 소스 파일 시스템과 동기화된 상태로 유지하는 프로세스를 관리하기 위해 Amazon EFS는 복제 구성을 사용합니다.

복제 구성을 생성하면 Amazon EFS는 소스 및 대상 파일 시스템을 자동으로 동기화합니다. 소스 파일 시스템의 변경 사항은 특정 시점에서 일관된 방식으로 대상 파일 시스템에 전송되지 않습니다. 대신 복제에 대한 마지막 동기화 시간을 기준으로 전송됩니다. 마지막 동기화 시간은 소스와 대상 간에 마지막으로 성공적으로 동기화가 완료된 시간을 나타냅니다. 마지막 동기화 시간을 기준으로 소스 파일 시스템에서 변경한 내용은 대상 파일 시스템에 복제되지만, 마지막 동기화 시간 이후에 소스 파일 시스템에

서 변경된 내용은 복제되지 않을 수 있습니다. 자세한 내용은 애플리케이션 세부 정보 보기 단원을 참조하십시오.

복제는 Amazon EFS를 사용할 수 있는 모든 AWS 리전 에서 사용할 수 있습니다. 기본적으로 비활성화되어 있는 리전에서 EFS 파일 시스템을 복제하려면 먼저 리전에 옵트인해야 합니다. 자세한 내용은 AWS 일반 참조 안내서의 계정에서 사용할 수 있는 AWS 리전 지정을 참조하세요. 나중에 리전을 옵트아웃하면 Amazon EFS는 해당 리전에 대한 모든 복제 활동을 일시 중지합니다. 리전에 대한 복제 활동을 재개하려면에 AWS 리전 다시 옵트인합니다.

> **Note**
>
> Replication은 ABAC(속성 기반 액세스 제어)에 대한 태그 사용을 지원하지 않습니다.

주제

- 비용
- 복제 성능
- 필수 IAM 권한
- 새 EFS 파일 시스템으로 복제 구성
- 기존 EFS 파일 시스템으로 복제 구성
- AWS 계정 간에 EFS 파일 시스템 복제
- 애플리케이션 세부 정보 보기
- 복제 구성 삭제
- 복제본 사용

비용

복제를 용이하게 하기 위해 Amazon EFS는 대상 파일 시스템에 숨겨진 디렉터리와 메타데이터를 생성합니다. 이는 청구 대상 측정 데이터로 약 12MiB에 해당합니다. 파일 시스템 스토리지 측정에 대한 자세한 내용은 Amazon EFS에서 파일 시스템 및 객체 크기를 보고하는 방법 섹션을 참조하세요.

복제 성능

페일백 프로세스 중에 새 복제를 생성하거나 기존 복제의 방향을 바꾸면 Amazon EFS는 복제를 지원하는 일련의 일회성 설정 작업을 포함하는 초기 동기화를 수행합니다. 복제된 데이터는 초기 동기화가

완료된 후에만 대상 파일 시스템에서 액세스할 수 있습니다. 초기 동기화가 완료되는 데 걸리는 시간은 소스 파일 시스템의 크기 및 그 내부의 파일 수에 따라 다릅니다.

초기 복제가 완료된 후 Amazon EFS는 대부분의 파일 시스템의 Recovery Point Objective(RPO)를 15분으로 유지합니다. 그러나 소스 파일 시스템에 자주 변경되는 파일이 있고 1억 개 이상의 파일 또는 100GB보다 큰 파일이 있는 경우 복제에 15분 이상 걸릴 수 있습니다. 마지막 복제가 성공적으로 완료된 시점 모니터링에 대한 자세한 내용은 애플리케이션 세부 정보 보기을 참조하세요.

콘솔, AWS Command Line Interface (AWS CLI), API 및 Amazon CloudWatch를 사용하여 마지막으로 동기화에 성공한 시기를 모니터링할 수 있습니다. CloudWatch에서는TimeSinceLastSync EFS 지표를 사용합니다. 자세한 내용은 애플리케이션 세부 정보 보기 단원을 참조하십시오.

필수 IAM 권한

Amazon EFS는 `AWSServiceRoleForAmazonElasticFileSystem`이라는 이름의 EFS 서비스 연결 역할 또는 사용자가 지정한 IAM 역할을 사용하여 소스 및 대상 파일 시스템 간에 복제를 동기화합니다. IAM 역할을 제공하려면 복제 구성을 생성하는 IAM 사용자 또는 역할에 `iam:PassRole` 권한이 있어야 합니다. 자세한 내용은 사용 AWS Identity and Access Management 설명서의 AWS 서비스에 역할을 전달할 수 있는 사용자 권한 부여를 참조하세요.

- `iam:CreateServiceLinkedRole`에 대한 자세한 내용은 Amazon EFS에 대해 서비스 연결 역할 사용의 예제를 참조하세요.
- 사용자 지정 IAM 역할에 대한 자세한 내용은 사용자 지정 신뢰 정책으로 IAM 역할 생성 섹션을 참조하세요.

> **Note**
>
> 교차 계정 복제를 수행하는 경우 복제 구성을 생성할 때 IAM 역할을 제공해야 합니다. 서비스 연결 역할을 사용하는 것은 허용되지 않습니다. 자세한 내용은 AWS 계정 간에 EFS 파일 시스템 복제 단원을 참조하십시오.

복제 구성을 생성할 때 제공하는 서비스 연결 역할 또는 IAM 역할에는 복제에 대한 다음 권한이 있어야 합니다.

- `elasticfilesystem:DescribeFileSystems`
- `elasticfilesystem:CreateFileSystem`
- `elasticfilesystem:CreateReplicationConfiguration`

- `elasticfilesystem:DeleteReplicationConfiguration`
- `elasticfilesystem:DescribeReplicationConfigurations`

`AmazonElasticFileSystemFullAccess` 관리형 정책을 사용하면 필요한 모든 EFS 권한을 자동으로 가져올 수 있습니다. 자세한 내용은 AWS 관리형 정책: AmazonElasticFileSystemFullAccess 단원을 참조하십시오.

새 EFS 파일 시스템으로 복제 구성

Amazon EFS는 자동으로 새 파일 시스템을 생성하고 소스 파일 시스템의 데이터와 메타데이터를 선택한의 새 읽기 전용 대상 파일 시스템에 복사 AWS 리전 합니다. 새 파일 시스템에 복제할 때 암호화에 사용할 파일 시스템 유형과 AWS Key Management Service (AWS KMS) 키를 선택합니다. 또한 Amazon EFS는 대상 파일 시스템을 생성할 때 탑재 대상을 생성하지 않습니다. 복제 구성을 생성한 후에는 대상 파일 시스템을 탑재할 탑재 대상을 하나 이상 생성해야 합니다.

> **Note**
>
> 파일 시스템은 하나의 복제 구성에만 포함될 수 있습니다. 다른 복제 구성에서는 대상 파일 시스템을 소스 파일 시스템으로 사용할 수 없습니다.

- 파일 시스템 유형 - 파일 시스템 유형은 AWS 리전내 Amazon EFS 파일 시스템이 데이터를 저장하는 데 사용하는 가용성 및 내구성을 결정합니다.
  - Regional을 선택하면 AWS 리전내 모든 가용 영역에 걸쳐 데이터와 메타데이터를 중복으로 저장하는 파일 시스템을 만들 수 있습니다.
  - One Zone을 선택하면 단일 가용 영역 내에 데이터와 메타데이터를 중복 저장하는 파일 시스템을 생성할 수 있습니다.

  파일 시스템 유형에 대한 자세한 내용은 EFS 파일 시스템 유형 섹션을 참조하십시오.
- 암호화 - 모든 대상 파일 시스템은 저장 중 암호화가 활성화된 상태로 생성됩니다. 대상 파일 시스템을 암호화하는 데 사용되는 AWS KMS 키를 지정할 수 있습니다. KMS 키를 지정하지 않으면 Amazon EFS의 서비스 관리형 KMS 키가 사용됩니다.

> **Important**
>
> 대상 파일 시스템이 생성되면 KMS 키를 변경할 수 없습니다.

대상 파일 시스템은 소스 파일 시스템을 기반으로 기본 설정을 사용하여 생성됩니다. 생성 후 추가 설정을 변경할 수 있습니다.

- 자동 백업 - One Zone 스토리지를 사용하는 대상 파일 시스템의 경우 자동 백업이 기본적으로 활성화됩니다. 파일 시스템이 생성되면 자동 백업 설정을 변경할 수 있습니다. 자세한 내용은 EFS 파일 시스템의 자동 백업 관리 단원을 참조하십시오.
- 성능 모드 - 대상 파일 시스템이 One Zone 스토리지를 사용하지 않는 한, 대상 파일 시스템의 성능 모드는 소스 파일 시스템의 성능 모드와 일치합니다. 이 경우에는 범용 모드가 사용됩니다. 성능 모드는 변경할 수 없습니다.
- 처리량 모드 - 대상 파일 시스템의 처리량 모드가 소스 파일 시스템의 처리량 모드와 일치합니다. 파일 시스템이 생성된 후 모드를 변경할 수 있습니다.

  소스 파일 시스템의 처리량 모드가 프로비저닝된 처리량인 경우, 소스 파일의 프로비저닝된 양이 대상 파일 시스템의 리전 한도를 초과하지 않는 한, 대상 파일 시스템의 프로비저닝된 처리량은 소스 파일 시스템의 처리량과 일치합니다. 소스 파일 시스템의 프로비저닝량이 대상 파일 시스템의 리전 한도를 초과하는 경우 대상 파일 시스템의 프로비저닝 처리량은 리전 한도가 됩니다. 자세한 내용은 늘릴 수 있는 Amazon EFS 할당량 단원을 참조하십시오.

- 수명 주기 관리 - 대상 파일 시스템에서는 수명 주기 관리가 활성화되어 있지 않습니다. 대상 파일 시스템이 생성되면 이를 활성화할 수 있습니다. 자세한 내용은 스토리지 수명 주기 관리 단원을 참조하십시오.

1단계: 복제 구성 생성

새 파일 시스템에 복제하는 첫 번째 단계는 복제 구성을 생성하는 것입니다. 대상 파일 시스템에 복제된 데이터는 초기 동기화가 완료된 후에만 액세스할 수 있습니다. 초기 동기화 소요 시간은 소스 파일 시스템의 크기와 그 안에 포함된 파일 수와 같은 요소에 따라 달라집니다. 복제 성능에 대한 자세한 내용은 복제 성능 섹션을 참조하세요.

콘솔 사용

1. 에 로그인 AWS Management Console 하고 https://console.aws.amazon.com/efs/ Amazon EFS 콘솔을 엽니다.
2. 복제할 파일 시스템을 엽니다.

   a. 왼쪽 탐색 창에서 파일 시스템을 선택합니다.

b. 파일 시스템 목록에서 복제하려는 파일 시스템을 선택합니다. 선택한 파일 시스템은 기존 복제 구성의 소스 또는 대상 파일 시스템이 될 수 없습니다.

3. 복제 탭을 선택합니다.
4. 복제 섹션에서 복제 생성을 선택합니다.
5. 복제 설정 섹션에서 복제 설정을 정의합니다.

   a. 복제 구성에서 새 파일 시스템으로 복제할지 기존 파일 시스템으로 복제할지 여부를 선택합니다.

   b. 대상 AWS 리전 AWS 리전 에서 파일 시스템을 복제할를 선택합니다.

6. 대상 파일 시스템 설정 섹션에서 대상 파일 시스템 설정을 정의합니다.

   a. 파일 시스템 유형에서 파일 시스템의 스토리지 옵션을 선택합니다.

   - 내의 지리적으로 분리된 여러 가용 영역에 데이터를 중복 저장하는 파일 시스템을 생성하려면 리전을 AWS 리전선택합니다.
   - 의 단일 가용 영역 내에 데이터를 중복 저장하는 파일 시스템을 생성하려면 One Zone을 AWS 리전선택한 다음 가용 영역을 선택합니다.

   자세한 내용은 EFS 파일 시스템 유형 단원을 참조하십시오.

   > **Note**
   >
   > Amazon EFS를 사용할 수 있는 AWS 리전 의 모든 가용 영역에서는 One Zone 파일 시스템을 사용할 수 없습니다.

   b. 암호화의 경우 대상 파일 시스템에서 저장 데이터의 암호화가 자동으로 활성화됩니다. 기본적으로 Amazon EFS는 AWS Key Management Service (AWS KMS) 서비스 키()를 사용합니다`aws/elasticfilesystem`. 다른 KMS 키를 사용하려면 KMS 키를 선택하거나 키의 Amazon 리소스 이름(ARN)을 입력합니다.

   > **Important**
   >
   > 파일 시스템이 생성된 후에는 KMS 키를 변경할 수 없습니다.

복제 구성 생성(AWS CLI)

이 단원에서는 `create-replication-configuration` 명령을 사용하여 AWS CLI 에서 복제 구성을 생성하는 예제를 제공합니다. 동등한 API 명령은 CreateReplicationConfiguration입니다.

Example: Regional 대상 파일 시스템의 복제 구성을 생성합니다.

다음 예제에서는 파일 시스템 *fs-0123456789abcdef1*에 대한 복제 구성을 생성합니다. 이 예제에서는 `Region` 파라미터를 사용하여에서 대상 파일 시스템을 생성합니다*eu-west-2* AWS 리전. `KmsKeyId` 파라미터는 대상 파일 시스템을 암호화할 때 사용할 KMS 키 ID를 지정합니다.

```
aws efs create-replication-configuration \
--source-file-system-id fs-0123456789abcdef1 \
--destinations "[{\"Region\":\"eu-west-2\", \"KmsKeyId\":\"arn:aws:kms:us-
east-2:111122223333:key\/abcd1234-ef56-ab78-cd90-1111abcd2222\"}]"
```

는 다음과 같이 AWS CLI 응답합니다.

```
        {
    "SourceFileSystemArn": "arn:aws:elasticfilesystem:us-east-1:111122223333:file-
system/fs-0123456789abcdef1",
    "SourceFileSystemRegion": "us-east-1",
    "Destinations": [
        {
            "Status": "ENABLING",
            "FileSystemId": "fs-0123456789abcde22",
            "Region": "eu-west-2"
        }
    ],
    "SourceFileSystemId": "fs-0123456789abcdef1",
    "CreationTime": 1641491892.0,
    "OriginalSourceFileSystemArn": "arn:aws:elasticfilesystem:us-
east-1:111122223333:file-system/fs-0123456789abcdef1"
}
```

Example: One Zone 대상 파일 시스템의 복제 구성을 생성합니다.

다음 예제에서는 파일 시스템 *fs-0123456789abcdef1*에 대한 복제 구성을 생성합니다. 이 예제에서는 `AvailabilityZoneName` 파라미터를 사용하여 *us-west-2a* 가용 영역에서 One Zone 대상 파

일 시스템을 생성합니다. KMS 키가 지정되지 않았으므로 대상 파일 시스템은 계정의 기본 AWS KMS 서비스 키(aws/elasticfilesystem)를 사용하여 암호화됩니다.

```
aws efs create-replication-configuration \
--source-file-system-id fs-0123456789abcdef1 \
--destinations AvailabilityZoneName=us-west-2a
```

2단계: 대상 파일 시스템 탑재

Amazon EFS는 대상 파일 시스템을 생성할 때 탑재 대상을 생성하지 않습니다. 대상 파일 시스템을 탑재하려면 탑재 대상을 하나 이상 생성해야 합니다. 자세한 내용은 EFS 파일 시스템 탑재 단원을 참조하십시오.

기존 EFS 파일 시스템으로 복제 구성

Amazon EFS는 소스 파일 시스템의 데이터와 메타데이터를 대상 파일 시스템과 사용자가 선택한 대상 파일 시스템에 복제 AWS 리전 합니다. 복제 중에 Amazon EFS는 파일 시스템 간의 데이터 차이를 식별하여 대상 파일 시스템에 차이를 적용합니다.

기존 파일 시스템으로 복제하려면 다음 단계를 수행합니다.

주제

- 1단계: 파일 시스템의 복제 덮어쓰기 보호 비활성화
- 2단계: 복제 구성 생성

> **Note**
>
> 파일 시스템은 하나의 복제 구성에만 포함될 수 있습니다. 다른 복제 구성에서는 대상 파일 시스템을 소스 파일 시스템으로 사용할 수 없습니다.

1단계: 파일 시스템의 복제 덮어쓰기 보호 비활성화

Amazon EFS 파일 시스템을 생성할 때 복제 덮어쓰기 보호가 기본적으로 활성화됩니다. 복제 덮어쓰기 보호 기능은 파일 시스템이 복제 구성의 대상으로 사용되는 것을 방지합니다. 복제 구성에서 파일 시스템을 대상으로 사용하려면 먼저 보호를 비활성화해야 합니다. 복제 구성을 삭제하면 파일 시스템의 복제 덮어쓰기 보호가 다시 활성화되고 파일 시스템은 쓰기 가능해집니다.

Amazon EFS 파일 시스템의 복제 덮어쓰기 보호 상태는 다음 표에 설명된 상태 값 중 하나를 가질 수 있습니다.

파일 시스템 상태	설명
ENABLED	파일 시스템은 복제 구성의 대상 파일 시스템으로 사용할 수 없습니다. 파일 시스템은 쓰기 가능합니다. 복제 덮어쓰기 보호는 기본적으로 ENABLED 상태입니다.
DISABLED	파일 시스템은 복제 구성의 대상 파일 시스템으로 사용할 수 있습니다.
복제	파일 시스템은 복제 구성의 대상 파일 시스템으로 사용 중입니다. 파일 시스템은 읽기 전용이며 복제 중 Amazon EFS를 통해서만 수정됩니다.

필수 권한

복제 덮어쓰기 보호를 비활성화하려면 `elasticfilesystem:UpdateFileSystemProtection` 작업에 대한 권한이 필요합니다. 자세한 내용은 AWS 관리형 정책: AmazonElasticFileSystemFullAccess 단원을 참조하십시오.

콘솔 사용

1. 에 로그인 AWS Management Console 하고 https://console.aws.amazon.com/efs/ Amazon EFS 콘솔을 엽니다.
2. 왼쪽 탐색 창에서 파일 시스템을 선택합니다.
3. 파일 시스템 목록에서 복제 구성에서 대상 파일 시스템으로 사용하려는 Amazon EFS 파일 시스템을 선택합니다.
4. 파일 시스템 보호 섹션에서 복제 덮어쓰기 보호를 끕니다.

복제 덮어쓰기 보호 비활성화(AWS CLI)

다음 예제에서는 `update-file-system-protection` CLI 명령이 지정된 파일 시스템에 대한 복제 덮어쓰기 보호를 비활성화합니다. 동일한 API 명령은 UpdateFileSystemProtection입니다.

```
aws efs update-file-system-protection
 --file-system-id fs-0a8b2be428114d97c
 --replication-overwrite-protection DISABLED
```

는 다음과 같이 AWS CLI 응답합니다.

```
{
    "ReplicationOverwriteProtection": "DISABLED"
}
```

2단계: 복제 구성 생성

대상 파일 시스템에서 복제 덮어쓰기 보호를 비활성화한 후 복제 구성을 생성할 수 있습니다. 기존 파일 시스템에 복제할 때 대상 파일 시스템은 소스 파일 시스템과 동일한 계정 또는 다른 계정에 있을 수 있습니다.

Amazon EFS에 대한 복제 구성을 생성하기 전에 다음 중요 요구 사항 및 고려 사항을 검토하세요.

- 소스 파일 시스템이 암호화된 경우 대상 파일 시스템도 암호화해야 합니다. 또한 소스 파일이 암호화되지 않고 대상 파일 시스템이 암호화된 경우 장애 조치를 수행한 후 소스 대상으로 페일백할 수 없습니다. 암호화에 대한 자세한 내용은 Amazon EFS의 데이터 암호화 섹션을 참조하세요.
- 기존 파일 시스템에 대한 복제를 처음 구성할 때 Amazon EFS는 원본 파일 시스템의 데이터와 일치하도록 대상 파일 시스템에 데이터를 쓰거나 대상 파일 시스템에서 기존 데이터를 제거합니다. 대상 파일 시스템의 데이터를 변경하지 않으려면 대신 새 파일 시스템에 복제해야 합니다. 자세한 내용은 새 EFS 파일 시스템으로 복제 구성 단원을 참조하십시오.
- 대상 파일 시스템으로 복제된 데이터는 초기 동기화가 완료된 후에만 액세스할 수 있습니다. 동기화 소요 시간은 소스 파일 시스템의 크기와 그 안에 포함된 파일 수와 같은 요소에 따라 달라집니다. 복제 성능에 대한 자세한 내용은 복제 성능 섹션을 참조하세요.

사전 조건

사용하려는 대상 파일 시스템 ID(동일 계정 복제용) 또는 대상 파일 시스템 ARN(교차 계정 복제용)의 사본이 있어야 합니다.

대상 파일 시스템이 AWS 계정 소스 파일 시스템과 다른에 있는 경우 Amazon EFS가 복제를 수행하고 파일 시스템에 리소스 정책을 할당할 수 있도록 허용하는 IAM 역할을 생성합니다. 자세한 내용은 AWS 계정 간에 EFS 파일 시스템 복제 단원을 참조하십시오.

콘솔 사용

1. 에 로그인 AWS Management Console 하고 https://console.aws.amazon.com/efs/ Amazon EFS 콘솔을 엽니다.
2. 복제할 파일 시스템을 엽니다.

a. 왼쪽 탐색 창에서 파일 시스템을 선택합니다.

b. 파일 시스템 목록에서 복제하려는 Amazon EFS 파일 시스템을 선택합니다. 선택한 파일 시스템은 기존 복제 구성의 소스 또는 대상 파일 시스템이 될 수 없습니다.

3. 복제 탭을 선택합니다.
4. 복제 섹션에서 복제 생성을 선택합니다.
5. 복제 구성에서 기존 파일 시스템을 선택합니다.
6. 대상 파일 시스템을 선택합니다.

- AWS 계정 소스 파일 시스템과 동일한에 있는 파일 시스템에 복제하려면:
  1. 이 계정에서 파일 시스템 선택을 선택하고 대상 AWS 리전에서 파일 시스템을 복제할를 선택합니다 AWS 리전 .
  2. EFS 찾아보기를 선택한 다음 파일 시스템을 선택합니다. 대상 파일 시스템의 경로가 대상 상자에 표시됩니다.

- AWS 계정 소스 파일 시스템과 다른에 있는 파일 시스템에 복제하려면:
  1. 다른 계정에서 파일 시스템 지정을 선택합니다.
  2. 대상 파일 시스템 ARN에 대상 파일 시스템의 Amazon 리소스 이름(ARN)을 입력합니다.

> **Note**
>
> 파일 시스템에서 복제 덮어쓰기 보호가 활성화된 경우 경고가 표시됩니다. 보호 기능 비활성화를 선택하여 파일 시스템을 새 탭에서 열고 복제 덮어쓰기 보호를 끕니다. 보호를 비활성화한 후 복제 생성 탭으로 돌아가서 새로 고침 버튼을 클릭하여 메시지를 지웁니다.

7. IAM 역할에 Amazon EFS가 대상 파일 시스템에 복제할 수 있도록 허용하는 IAM 역할의 ARN을 입력합니다. 이는 동일한 계정 복제의 경우 선택 사항이지만 교차 계정 복제에는 필요합니다. 자세한 내용은 AWS 계정 간에 EFS 파일 시스템 복제 단원을 참조하십시오.
8. 복제 생성을 선택하고 확인 메시지 입력 상자에 확인을 입력한 다음 복제 생성을 선택합니다. 복제 섹션에 복제 세부 정보가 표시됩니다.

복제 구성 생성(AWS CLI)

이 단원에서는 `create-replication-configuration` 명령을 사용하여 AWS CLI 에서 복제 구성을 생성하는 예제를 제공합니다. 동등한 API 명령은 CreateReplicationConfiguration입니다.

Example다른 리전의 기존 대상 파일 시스템으로 복제 구성 생성

다음 예제에서는 파일 시스템 ID*fs-0123456789abcdef1*가의 파일 시스템 ID *fs-0a8b2be428114d97c*에 복제되는 복제 구성을 생성합니다*eu-west-2* AWS 리전.

```
aws efs create-replication-configuration \
--source-file-system-id fs-0123456789abcdef1 \
--destinations "[{\"Region\":\"eu-west-2\",\"FileSystemId\":\"fs-0a8b2be428114d97c\"}]"
```

는 다음과 같이 AWS CLI 응답합니다.

```
{
    "SourceFileSystemId": "fs-0123456789abcdef1",
    "SourceFileSystemRegion": "us-east-1",
    "SourceFileSystemArn": "arn:aws:elasticfilesystem:us-east-1:111122223333:file-
system/fs-0123456789abcdef1",
    "OriginalSourceFileSystemArn": "arn:aws:elasticfilesystem:us-
east-1:111122223333:file-system/fs-0123456789abcdef1",
    "CreationTime": "2024-10-20T20:40:13+00:00",
    "Destinations": [
        {
            "Status": "ENABLING",
            "FileSystemId": "fs-0a8b2be428114d97c",
            "Region": "eu-west-2",
            "OwnerId": "123456789012,
         }
    ],
    "SourceFileSystemOwnerId": "123456789012"
}
```

Example: 교차 계정 복제 구성 생성

다음 예제에서는 소스 파일 시스템과 대상 파일 시스템이 서로 다른 복제 구성을 생성합니다 AWS 계정. 계정 *555666777888* 내의 소스 파일 시스템 ID *fs-0123456789abcdef1*이 계정 *123456789012* 내의 파일 시스템 ID *fs-0a8b2be428114d97c*로 복제됩니다. 이 예제에서는 대상 파일 시스템의 Amazon 리소스 이름(ARN)과 Amazon EFS가 해당 계정을 대신하여 복제를 수행할 수 있도록 허용하는 소스 계정의 IAM 역할 ARN을 지정합니다. KMS 키가 지정되지 않았으므로 대상 파일 시스템은 계정의 기본 AWS KMS 서비스 키()를 사용하여 암호화됩니다`aws/elasticfilesystem`.

```
aws efs
--region $REGION
--endpoint $ENDPOINT create-replication-configuration
```

```
--source-file-system-id fs-0123456789abcdef1
--destinations Region=eu-west-2,FileSystemId=arn:aws:elasticfilesystem:eu-
west-2:123456789012:file-system/
fs-0a8b2be428114d97c,RoleArn=arn:aws:iam::555666777888:role/cross-account-replication
```

는 다음과 같이 AWS CLI 응답합니다.

```
{
    "SourceFileSystemId": "fs-0123456789abcdef1",
    "SourceFileSystemRegion": "us-east-1",
    "SourceFileSystemArn": "arn:aws:elasticfilesystem:us-east-1:555666777888:file-
system/fs-0123456789abcdef1",
    "OriginalSourceFileSystemArn": "arn:aws:elasticfilesystem:us-
east-1:555666777888:file-system/fs-0123456789abcdef1",
    "CreationTime": "2024-10-20T20:40:13+00:00",
    "Destinations": [
        {
            "Status": "ENABLING",
            "FileSystemId": "fs-0a8b2be428114d97c",
            "Region": "eu-west-2",
            "OwnerId": "123456789012,
            "RoleArn": "arn:aws:iam::555666777888:role/cross-account-replication"
        }
    ],
    "SourceFileSystemOwnerId": "555666777888"
}
```

AWS 계정 간에 EFS 파일 시스템 복제

AWS 계정에서 EFS 파일 시스템을 복제할 수 있습니다. 계정 간에 복제하면 재해 복구(DR) 전략의 전반적인 복원력과 신뢰성이 향상되고 기업 규정 준수 요구 사항을 충족하는 데 도움이 될 수 있습니다.

예를 들어 규정 준수 정책에 따라 환경(예: 프로덕션, 스테이징 및 재해 복구(DR))마다 다른 계정을 사용해야 할 수 있습니다. 또는 서로 다른 간에 복제하면 더 강력한 격리, 권한 및 액세스 정책에 대한 더 세분화된 제어, 리소스에 대한 더 간단한 감사를 AWS 계정 제공할 수 있습니다. 프로덕션 계정이 손상된 경우(예: 보안 침해, 잘못된 구성 또는 내부자 위협) 별도의 계정에 DR 서버를 배치하면 공격자가 액세스하지 못하게 하고, 보안 인시던트의 폭발 반경을 줄이며, 무단 변경의 위험을 최소화할 수 있습니다.

간에 복제하려면 추가 보안 및 정책 설정이 AWS 계정 필요합니다. 서비스 연결 역할을 사용하여 교차 계정 복제를 수행하는 대신 대상 계정에서 복제를 수행할 수 있는 권한을 Amazon EFS에 부여하는

IAM 역할을 생성해야 합니다. 또한 계정 간에 공유하려는 파일 시스템에 정책을 생성해야 합니다. IAM 역할 및 파일 시스템 정책이 생성된 후 복제 구성을 생성합니다.

주제

- 사용자 지정 신뢰 정책으로 IAM 역할 생성
- 소스 및 대상 파일 시스템에 정책 생성
- 복제 구성 생성

사용자 지정 신뢰 정책으로 IAM 역할 생성

Amazon EFS가 소스 계정을 대신하여 교차 계정 복제를 수행하려면 소스 계정에 IAM 역할을 생성해야 합니다. 해당 역할에는 Amazon EFS가 역할을 인수하고 서비스 보안 주체로 활동할 수 있도록 허용하는 `elasticfilesystem.amazonaws.com` 신뢰 정책이 있어야 합니다. 역할에는 복제를 수행하는 데 필요한 모든 IAM 권한이 포함되어야 하며(필수 IAM 권한 참조) 대상 계정의 파일 시스템에 복제할 명시적 권한을 부여해야 합니다.

사전 조건

소스 계정에 대한 IAM 역할을 생성하려면 먼저 복제 구성에서 소스 파일 시스템과 대상 파일 시스템을 모두 생성해야 합니다. Amazon EFS는 복제 중에 대상 파일 시스템을 생성할 수 없습니다. 또한 각 파일 시스템의 Amazon 리소스 이름(ARN)을 알고 제공해야 합니다.

교차 계정 복제를 위한 IAM 역할 생성

다음은 Amazon EFS를 사용한 교차 계정 복제를 위한 사용자 지정 신뢰 정책을 사용하여 IAM 역할을 생성하는 일반적인 단계입니다. IAM 역할 생성 단계별 지침은 AWS Identity and Access Management 사용 설명서의 사용자 지정 신뢰 정책을 사용하여 역할 생성을 참조하세요.

1. 소스 계정의 AWS Identity and Access Management 콘솔에서 다음 신뢰 정책을 사용하는 IAM 역할을 생성합니다. 자세한 지침은 AWS Identity and Access Management 사용 설명서의 사용자 지정 신뢰 정책을 사용하여 역할 생성을 참조하세요.

   JSON

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

```
            "Effect": "Allow",
            "Principal": {
                "Service": "elasticfilesystem.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

2. 역할을 생성한 후, 해당 역할에 다음 권한을 할당합니다. `arn:aws:elasticfilesystem:us-east-1:111122223333:file-system/fs-0123456789abcdef1`을 대상 파일 시스템의 ARN으로 바꾸고, `arn:aws:elasticfilesystem:us-east-1:444455556666:file-system/fs-5678910112hijkqr1`을 소스 파일 시스템의 ARN으로 바꿉니다. 역할에 권한을 할당하는 방법에 대한 지침은 JSON 편집기를 사용하여 정책 생성을 참조하세요.

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action":[
                "elasticfilesystem:DescribeFileSystems",
                "elasticfilesystem:CreateReplicationConfiguration",
                "elasticfilesystem:DescribeReplicationConfigurations",
                "elasticfilesystem:DeleteReplicationConfiguration",
                "elasticfilesystem:ReplicationWrite"
            ],
            "Resource": "arn:aws:elasticfilesystem:us-east-1:111122223333:file-system/fs-0123456789abcdef1"
        },
        {
            "Effect": "Allow",
            "Action": [
                "elasticfilesystem:ReplicationRead",
                "elasticfilesystem:DescribeFileSystems"
            ],
            "Resource": "arn:aws:elasticfilesystem:us-east-1:444455556666:file-system/fs-5678910112hijkqr1"
        }
    ]
```

```
}
```

3. IAM 역할의 ARN을 복사하거나 기록해 둡니다. 복제 구성 생성 시 ARN을 제공해야 합니다.

소스 및 대상 파일 시스템에 정책 생성

Amazon EFS에서 파일 시스템 교차 계정을 공유하려면 대상 파일 시스템과 소스 파일 시스템 모두에 정책을 할당해야 합니다. 정책은 계정 간에 적용되는 파일 시스템에 대한 액세스 권한을 부여하거나 제한합니다. 파일 시스템을 편집할 권한이 있는 계정 소유자만 계정의 파일 시스템에 정책을 할당할 수 있습니다.

정책은 계정 간 접근을 허용하거나 제한하는 것 외에도, 클라이언트가 파일 시스템과 작업하는 데 필요한 다른 권한(예: `elasticfilesystem:ClientMount`)을 부여해야 합니다. 그렇지 않으면 클라이언트가 파일 시스템에 액세스할 수 없게 될 수 있습니다.

> **Important**
>
> TLS 연결을 통한 리소스 액세스를 제한할 수 없습니다. 문에 `"aws:SecureTransport": "false"` 조건을 포함하면 NFS 클라이언트 연결이 실패합니다.

대상 파일 시스템에 대한 정책

소스 계정이 대상 파일 시스템으로 복제하고 대상 계정에서 복제 구성을 삭제할 수 있도록 하려면 대상 파일 시스템에 다음 정책을 생성해야 합니다. `arn:aws:iam::`*`444455556666`*`:root`를 소스 파일 시스템을 소유한 계정의 ID로 바꿉니다. `arn:aws:elasticfilesystem:`*`us-east-1`*`:`*`111122223333`*`:file-system/fs-0123456789abcdef1`를 대상 파일 시스템의 ARN으로 바꿉니다.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
     {
        "Sid": "AllowSourceAccountReplicationActions",
        "Effect": "Allow",
        "Principal": {
        "AWS": "arn:aws:iam::444455556666:root"
```

```
      },
    "Action": [
        "elasticfilesystem:DescribeFileSystems",
        "elasticfilesystem:CreateReplicationConfiguration",
        "elasticfilesystem:DescribeReplicationConfigurations",
        "elasticfilesystem:DeleteReplicationConfiguration",
        "elasticfilesystem:ReplicationWrite"
        ],
     "Resource": "arn:aws:elasticfilesystem:us-east-1:111122223333:file-system/
fs-0123456789abcdef1"
     },
     {
        "Sid": "AllowReadOnlyClientAccess",
        "Effect": "Allow",
        "Principal": {
            "AWS": "arn:aws:iam::111122223333:role/EfsReadOnly"
        },
        "Action": [
            "elasticfilesystem:ClientMount"
        ],
        "Resource": "arn:aws:elasticfilesystem:us-east-1:111122223333:file-
system/fs-0123456789abcdef1"
     }
   ]
}
```

소스 파일 시스템에 대한 정책

대상 계정 권한이 소스 계정에서 복제 구성을 삭제하도록 허용하려면 소스 파일 시스템에 다음 정책을 할당해야 합니다. `arn:aws:iam::111122223333:root`를 대상 파일 시스템을 소유한 계정의 ID로 바꿉니다. `arn:aws:elasticfilesystem:us-east-1:444455556666:file-system/fs-5678910112hijkqr1`를 소스 파일 시스템의 ARN으로 바꿉니다.

JSON

```
{
    "Version":"2012-10-17",
    "Id": "efs-policy-wizard-15ad9567-2546-4bbb-8168-5541b6fc0e55",
    "Statement": [
        {
            "Sid": "AllowDestinationAccountToDeleteReplication",
```

```
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:root"
            },
            "Action": "elasticfilesystem:DeleteReplicationConfiguration",
            "Resource": "arn:aws:elasticfilesystem:us-east-1:444455556666:file-
system/fs-5678910112hijkqr1"
        },
        {
            "Sid": "AllowClientAccess",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:role/EfsReadOnly"
            },
            "Action": [
                "elasticfilesystem:ClientMount",
                "elasticfilesystem:ClientWrite",
                "elasticfilesystem:ClientRootAccess"
            ],
            "Resource": "arn:aws:elasticfilesystem:us-east-1:444455556666:file-
system/fs-5678910112hijkqr1",
            "Condition": {
               "Bool": {
                   "elasticfilesystem:AccessedViaMountTarget": "true"
               }
           }
        }
    ]
}
```

파일 시스템 정책 생성

이전 섹션의 정책을 사용하여 대상 및 소스 파일 시스템 모두에 대해 다음 단계를 수행합니다.

1. 파일 시스템을 소유한 AWS Management Console 계정으로에 로그인한 다음 Amazon EFS 콘솔에서 Amazon EFS 콘솔을 엽니다.
2. 다음 방법으로 파일 시스템을 엽니다.
   a. 왼쪽 탐색 창에서 파일 시스템을 선택합니다.
   b. 파일 시스템 목록에서 파일 시스템을 선택합니다.
3. 파일 시스템 정책 탭에서 편집을 선택합니다.

4. 정책 편집기 {Json}에 정책을 붙여넣은 후 저장을 선택합니다.

복제 구성 생성

IAM 역할을 생성하고 소스 및 대상 파일 시스템에 파일 시스템 정책을 추가한 후, 기존 EFS 파일 시스템으로 복제 구성의 지침에 따라 복제 구성을 생성합니다.

애플리케이션 세부 정보 보기

복제 구성에서 마지막으로 동기화에 성공한 시간을 모니터링할 수 있습니다. 이 시간 이전에 발생한 소스 파일 시스템의 모든 데이터 변경 사항은 대상 파일 시스템에 성공적으로 복제되었습니다. 이 시간 이후에 발생한 모든 변경 사항은 완전히 복제되지 않을 수 있습니다. 마지막 복제가 성공적으로 완료된 시기를 모니터링하려면 Amazon EFS 콘솔, AWS CLI API 또는 Amazon CloudWatch를 사용할 수 있습니다.

- EFS 콘솔 - 파일 시스템 세부 정보 > 복제 섹션의 마지막 동기화 속성은 소스와 대상 간에 동기화가 마지막으로 성공적으로 완료된 시간을 표시합니다.
- AWS CLI 또는 API - `Destination` 객체의 `LastReplicatedTimestamp` 속성에는 마지막으로 성공한 동기화가 완료된 시간이 표시됩니다. 이 속성에 액세스하려면 `describe-replication-configurations` CLI 명령을 사용합니다. DescribeReplicationConfigurations는 동등한 API 작업입니다.
- CloudWatch에서 - Amazon EFS의 `TimeSinceLastSync` CloudWatch 지표는 마지막으로 성공적으로 동기화가 완료된 이후 경과된 시간을 보여줍니다. 자세한 내용은 Amazon EFS에 대한 CloudWatch 지표 단원을 참조하십시오.

복제 구성에는 다음 표에 설명된 상태 값 중 하나가 있을 수 있습니다.

복제 상태	설명
`ENABLED`	복제 구성이 정상 상태이며 사용할 수 있습니다.
`ENABLING`	Amazon EFS에서 복제 구성을 생성하는 중입니다.
`DELETING`	Amazon EFS는 사용자가 시작한 삭제 요청에 대한 응답으로 복제 구성을 삭제합니다.
`PAUSING`	Amazon EFS가 복제를 일시 중지하는 중입니다.

복제 상태	설명
PAUSED	구성 문제로 인해 복제가 일시 중지되었습니다. 문제에 대한 추가 정보가 제공됩니다. 복제가 일시 중지되는 몇 가지 문제는 다음과 같습니다. • 권한 부여 오류. 권한 문제로 인해 Amazon EFS가 복제를 수행할 수 없습니다. 복제 구성을 생성하는 데 사용되는 IAM 역할에 복제를 수행할 권한이 있는지 확인합니다. 또한 파일 시스템 정책이 올바른지 확인합니다. • AWS 계정을 사용할 수 없습니다. 소스 및 대상 계정이 적절한에 옵트인 AWS 리전 되고 일시 중지되지 않았는지 확인합니다. 자세한 내용은 AWS 일반 참조 안내서의 계정에서 사용할 수 있는 지정을 참조 AWS 리전하세요. • 소스 또는 대상 파일 시스템의 KMS 키에 액세스할 수 없습니다. 각 파일 시스템에 할당된 KMS 키에 액세스할 수 있는지 확인합니다. 자세한 내용은 Amazon EFS에 AWS KMS 키 사용 단원을 참조하십시오.
ERROR	복제 구성이 실패 상태이며 복구할 수 없습니다. 복제 구성을 삭제하고 새로 생성해야 합니다. 문제에 대한 추가 정보가 제공됩니다. 교차 계정 또는 교차 리전 복제의 경우 복제 구성이 다른 AWS 계정 또는에서 삭제되었기 때문에 오류가 발생할 수 있습니다 AWS 리전.

콘솔 사용

1. Amazon Elastic File System 콘솔(https://console.aws.amazon.com/efs/)을 엽니다.
2. 왼쪽 탐색 창에서 파일 시스템을 선택합니다.
3. 목록에서 파일 시스템을 선택합니다.
4. 복제 탭을 선택하여 복제 섹션을 표시합니다.

   복제 섹션에서 복제 구성에 대한 다음 정보를 볼 수 있습니다.

   • 복제 상태는 활성화 중, 활성화됨, 삭제 중, 일시 중지 중, 일시 중지됨 또는 오류일 수 있습니다. Amazon EFS는 일시 중지됨 또는 오류 상태의 원인에 대한 세부 정보를 표시합니다.
   • 복제 방향은 데이터가 복제되는 방향을 나타냅니다. 나열된 첫 번째 파일 시스템이 소스이고 해당 데이터가 나열된 두 번째 파일 시스템(대상)으로 복제되고 있습니다.

- 마지막 동기화는 대상 파일 시스템에서 마지막으로 성공적으로 동기화된 시간을 나타냅니다. 이 시간 이전에 발생한 소스 파일 시스템의 모든 데이터 변경 사항은 대상 파일 시스템에 성공적으로 복제되었습니다. 이 시간 이후에 발생한 모든 변경 사항은 완전히 복제되지 않을 수 있습니다.
- 복제 파일 시스템은 복제 구성의 각 파일 시스템을 파일 시스템 ID, 복제 구성의 역할(소스 또는 대상), 파일 시스템이 AWS 리전 위치한 , 권한을 기준으로 나열합니다. 소스 파일 시스템에는 쓰기 가능 권한이 있고 대상 파일 시스템에는 읽기 전용 권한이 있습니다.

사용 AWS CLI

복제 구성을 보려면 `describe-replication-configurations` 명령을 사용합니다. 특정 파일 시스템의 복제 구성 또는의 특정에 대한 모든 복제 구성을 볼 수 AWS 계정 있습니다 AWS 리전. 동등한 API 명령은 DescribeReplicationConfigurations입니다.

복제 구성의 상태가 `PAUSED` 또는 `ERROR`인 경우 문제의 원인과 수정 방법에 대한 정보가 `StatusMessage` 파라미터에 반환됩니다.

Example: 특정 파일 시스템의 복제 구성 보기

다음 예제에서는 파일 시스템 `fs-0123456789abcdef1`의 복제 구성을 생성합니다.

```
aws efs describe-replication-configurations --file-system-id fs-0123456789abcdef1
```

는 다음과 같이 AWS CLI 응답합니다.

```
{
    "Replications": [
        {
            "SourceFileSystemArn": "arn:aws:elasticfilesystem:eu-
west-1:111122223333:file-system/fs-abcdef0123456789a",
            "CreationTime": 1641491892.0,
            "SourceFileSystemRegion": "eu-west-1",
            "OriginalSourceFileSystemArn": "arn:aws:elasticfilesystem:eu-
west-1:111122223333:file-system/fs-abcdef0123456789a",
            "SourceFileSystemId": "fs-abcdef0123456789a",
            "Destinations": [
                {
                    "Status": "ENABLED",
                    "FileSystemId": "fs-0123456789abcdef1",
                    "Region": "us-east-1"
```

```
                }
            ]
        }
    ]
}
```

Example: 계정 내 모든 복제 구성에 대한 복제 구성 보기

다음 예제에서는 파일 시스템의 계정에 대한 모든 복제 구성 AWS 리전 에 대한 복제 구성을 설명합니다.

```
aws efs describe-replication-configurations
```

는 다음과 같이 AWS CLI 응답합니다.

```
 {
    "Replications": [
        {
            "SourceFileSystemArn": "arn:aws:elasticfilesystem:eu-
west-1:555555555555:file-system/fs-0123456789abcdef1",
            "CreationTime": 1641491892.0,
            "SourceFileSystemRegion": "eu-west-1",
            "OriginalSourceFileSystemArn": "arn:aws:elasticfilesystem:eu-
west-1:555555555555:file-system/fs-0123456789abcdef1",
            "SourceFileSystemId": "fs-0123456789abcdef1",
            "Destinations": [
                {
                    "Status": "ENABLED",
                    "FileSystemId": "fs-abcdef0123456789a",
                    "Region": "us-east-1",
                    "LastReplicatedTimestamp": 1641491802.375
                }
            ]
        },
        {
            "SourceFileSystemArn": "arn:aws:elasticfilesystem:eu-
west-1:555555555555:file-system/fs-021345abcdef6789a",
            "CreationTime": 1641491822.0,
```

```
            "SourceFileSystemRegion": "eu-west-1",
            "OriginalSourceFileSystemArn": "arn:aws:elasticfilesystem:eu-
west-1:555555555555:file-system/fs-021345abcdef6789a",
            "SourceFileSystemId": "fs-021345abcdef6789a",
            "Destinations": [
                {
                    "Status": "ENABLED",
                    "FileSystemId": "fs-012abc3456789def1",
                    "Region": "us-east-1",
                    "LastReplicatedTimestamp": 1641491823.575
                }
            ]
        }
    ]
}
```

복제 구성 삭제

대상 파일 시스템으로 장애 조치해야 하는 경우 해당 시스템이 구성원으로 속해 있는 복제 구성을 삭제하세요. 복제 구성을 삭제하면 대상 파일 시스템이 쓰기 가능 상태가 되고 복제 덮어쓰기 보호가 다시 활성화됩니다. 자세한 내용은 복제본 사용 단원을 참조하십시오.

복제 구성을 삭제하고 대상 파일 시스템을 쓰기 가능으로 변경하는 작업을 완료하는 데 몇 분 정도 걸릴 수 있습니다. 구성을 삭제한 후 Amazon EFS는 다음 명명 규칙을 사용하여 대상 파일 시스템의 루트 디렉터리에 있는 `lost+found` 디렉터리에 일부 데이터를 쓸 수 있습니다.

```
efs-replication-lost+found-source-file-system-id-TIMESTAMP
```

> **Note**
>
> 복제 구성의 일부인 파일 시스템은 삭제할 수 없습니다. 파일 시스템을 삭제하기 전에 복제 구성을 삭제해야 합니다.

Amazon EFS 콘솔 AWS CLI, 또는 API를 사용하여 소스 또는 대상 파일 시스템에서 기존 복제 구성을 삭제할 수 있습니다.

교차 계정 또는 교차 리전 복제의 경우 Amazon EFS는 소스 및 대상 계정 또는 리전 모두에서 복제 구성을 삭제합니다. Amazon EFS가 양쪽에서 복제 구성을 삭제하지 못하게 하는 구성 또는 권한 문제가

있는 경우 로컬 측(삭제가 수행되는 계정 또는 리전)에서만 구성을 삭제할 수 있습니다. 로컬 구성을 삭제하면 구성을 다른 계정 또는 리전에서 복구할 수 없습니다.

콘솔 사용

1. Amazon Elastic File System 콘솔(https://console.aws.amazon.com/efs/)을 엽니다.
2. 왼쪽 탐색 창에서 파일 시스템을 선택합니다.
3. 복제 구성에서 삭제하려는 소스 또는 대상 파일 시스템을 선택합니다.
4. 복제 탭을 선택하여 복제 섹션을 표시합니다.
5. 복제 구성을 삭제하려면 복제 삭제를 선택합니다. 확인 메시지가 나타나면 선택을 확인합니다.

   교차 계정 복제 구성을 삭제하고 소스 및 대상 측 모두에서 구성을 삭제하지 못하는 문제가 있는 경우 해당 파일 시스템의 구성만 삭제하는 옵션을 선택할 수 있습니다.

   > **Note**
   >
   > Amazon EFS가 소스 및 대상 계정 또는 리전 모두에서 복제 구성을 삭제할 수 없는 경우에만 파일 시스템의 구성을 삭제합니다. 로컬 구성을 삭제하면 구성을 다른 계정 또는 리전에서 복구할 수 없습니다.

사용 AWS CLI

복제 구성을 검색하려면 `delete-replication-configuration` CLI를 사용합니다. 동등한 API 명령은 DeleteReplicationConfiguration입니다.

다음 예제에서는 파일 시스템 `fs-0123456789abcdef1`의 복제 구성을 생성합니다.

```
aws efs --region us-west-2 delete-replication-configuration \
--source-file-system-id fs-0123456789abcdef1
```

Amazon EFS가 양쪽에서 복제 구성을 삭제하지 못하게 하는 구성 또는 권한 문제가 있는 경우 로컬 측(삭제가 수행되는 계정 또는 리전)에서만 구성을 삭제할 수 있습니다. 로컬 구성을 삭제하면 구성을 다른 계정 또는 리전에서 복구할 수 없습니다. 동등한 API 파라미터는 `DeletionMode`이고 값은 `LOCAL_CONFIGURATION_ONLY`입니다.

다음 예제에서는 파일 시스템 *fs-0123456789abcdef1*의 복제 구성을 생성합니다.

```
aws efs --region us-west-2 delete-replication-configuration \
```

```
--source-file-system-id fs-0123456789abcdef1
--deletion-mode LOCAL_CONFIGURATION_ONLY
```

복제본 사용

재해가 발생하거나 게임데이 연습을 수행할 때 복제 구성을 삭제하여 복제 파일 시스템으로 장애 조치할 수 있습니다. 복제 구성이 삭제되면 복제본이 쓰기 가능해지며, 애플리케이션 워크플로에서 이를 사용할 수 있습니다. 재해가 완화되거나 게임데이 연습이 끝나면 복제본을 기본 파일 시스템으로 계속 사용하거나 장애 복구를 수행하여 원래 기본 파일 시스템에서 작업을 재개할 수 있습니다.

페일백 프로세스 중에 복제본 파일 시스템의 변경 내용을 취소하거나 기본 시스템에 다시 복사하여 보존할 수 있습니다.

- 장애 조치 중에 복제본에 대한 변경 내용을 무시하려면 기본 파일 시스템에서 원래 복제 구성을 다시 생성하십시오. 기본 파일 시스템에서는 복제본 파일 시스템이 복제 대상입니다. 복제 중에 Amazon EFS는 복제본 파일 시스템의 데이터를 기본 데이터와 일치하도록 업데이트하여 파일 시스템을 동기화합니다.
- 장애 조치 중에 복제본에 대한 변경 내용을 복제하려면 복제본 파일 시스템에서 복제 구성을 생성하십시오. 복제본 파일 시스템에서는 기본 파일 시스템이 복제 대상입니다. 복제 중에 Amazon EFS는 복제본 파일 시스템의 차이점을 식별하여 기본 파일 시스템으로 다시 전송합니다. 복제가 완료되면 원래 복제 구성을 다시 생성하거나 새 구성을 생성하여 기본 파일 시스템 복제를 재개할 수 있습니다.

Amazon EFS에서 복제 프로세스를 완료하는 데 걸리는 시간은 파일 시스템의 크기 및 그 내부의 파일 수와 같은 요인에 따라 달라집니다. 자세한 내용은 복제 성능 단원을 참조하십시오.

Amazon EFS에서 데이터 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 File System에 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 EFS 사용 시 공동 책임 모델을 적용하는 방법을 이해하는 데 도움이 됩니다. 다음 주제에서는 보안 및 규정 준수 목표에 맞게 Amazon EFS를 구성하는 방법을 보여줍니다. 또한 Amazon EFS 리소스를 모니터링하고 보호하는 데 도움이 되는 다른 AWS 서비스를 사용하는 방법을 알아봅니다.

주제

- Amazon EFS의 데이터 보호
- Amazon EFS의 자격 증명 및 액세스 관리
- IAM을 사용하여 파일 시스템에 대한 액세스 제어
- Amazon EFS에 대한 규정 준수 확인
- Amazon EFS의 복원성
- NFS 클라이언트용 EFS 파일 시스템에 대한 네트워크 액세스 제어
- NFS(Network File System) 수준 사용자, 그룹 및 권한 사용
- 액세스 포인트 작업
- EFS 파일 시스템에 대한 퍼블릭 액세스 차단
- Amazon EFS용 네트워크 격리

Amazon EFS의 데이터 보호

AWS 공동 책임 모델 Amazon EFS의 데이터 보호에 적용됩니다. 이 모델에 설명된 대로 AWS 는 모든 를 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 관한 자세한 내용은 데이터 프라이버시 FAQ를 참조하세요. 유럽의 데이터 보호에 대한 자세한 내용은 AWS 보안 블로그의 AWS 공동 책임 모델 및 GDPR 블로그 게시물을 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 EFS 또는 기타 AWS 서비스 에서 콘솔 AWS CLI, API 또는 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 보안 인증 정보를 URL에 포함시켜서는 안 됩니다.

주제

- Amazon EFS의 데이터 암호화
- 인터네트워크 개인 정보 보호

Amazon EFS의 데이터 암호화

Amazon EFS는 저장 데이터와 전송 중 데이터를 보호할 수 있는 포괄적인 암호화 기능을 제공합니다.

- 유휴 시 암호화 - 파일 시스템에 저장된 데이터를 암호화합니다.
- 전송 중 암호화 - 클라이언트와 파일 시스템 간에 이동하는 데이터를 암호화합니다.

조직이 데이터 및 메타데이터 암호화를 요구하는 기업이나 규제 정책을 준수해야 하는 경우, 전송 중 데이터 암호화를 사용하여 파일 시스템을 탑재하고 유휴 시 암호화된 파일 시스템을 생성하는 것이 좋습니다.

주제

- 저장 데이터 암호화
- 전송 중 데이터 암호화
- Amazon EFS에 AWS KMS 키 사용
- 암호화 문제 해결

저장 데이터 암호화

유휴 시 암호화는 EFS 파일 시스템에 저장된 데이터를 암호화합니다. 이를 통해 규정 준수 요구 사항을 충족하고 민감한 데이터를 무단 액세스로부터 보호할 수 있습니다. 조직에서 특정 분류를 충족하거나 특정 애플리케이션이나 워크로드, 환경과 연결된 모든 데이터를 암호화해야 할 수 있습니다.

> **Note**
>
> AWS 키 관리 인프라는 FIPS(Federal Information Processing Standards) 140-3 승인 암호화 알고리즘을 사용합니다. 이 인프라는 미국 국립 표준 기술 연구소(NIST) 800-57 표준의 권장 사항에 부합됩니다.

Amazon EFS 콘솔을 사용하여 파일 시스템을 생성하면 유휴 시 암호화가 기본적으로 활성화됩니다. AWS CLI, API 또는 SDKs를 사용하여 파일 시스템을 생성할 때는 암호화를 명시적으로 활성화해야 합니다.

EFS 파일 시스템을 생성한 후에는 암호화 설정을 변경할 수 없습니다. 즉, 암호화되지 않은 파일 시스템을 수정하여 암호화할 수 없습니다. 대신 파일 시스템을 복제하여 암호화되지 않은 파일 시스템의 데

이터를 암호화된 새 파일 시스템으로 복사합니다. 자세한 내용은 기존 EFS 파일 시스템에 대해 유휴 시 암호화를 켜려면 어떻게 해야 합니까?를 참조하세요.

유휴 시 암호화 작동 방식

암호화된 파일 시스템에서 데이터 및 메타데이터는 스토리지에 기록되기 전에 기본적으로 암호화되며 읽을 때 자동으로 해독됩니다. Amazon EFS는 해당 프로세스를 투명하게 처리하기 때문에 애플리케이션을 수정할 필요가 없습니다.

Amazon EFS는 다음과 같이 키 관리에 AWS KMS 를 사용합니다.

- 파일 데이터 암호화 - 파일 콘텐츠는 지정한 KMS 키를 사용하여 암호화됩니다. 이는 다음 중 하나일 수 있습니다.
  - Amazon EFS AWS 소유 키 용 (`aws/elasticfilesystem`) - 기본 옵션, 추가 요금 없음.
  - 고객이 생성 및 관리하는 고객 관리형 키 – 추가적인 제어 및 감사 기능을 제공합니다.
- 메타데이터 암호화 - 파일 이름, 디렉터리 이름 및 디렉터리 내용은 Amazon EFS가 내부적으로 관리하는 키를 사용하여 암호화됩니다.

암호화 프로세스

파일 시스템이 생성되거나 동일 계정의 파일 시스템으로 복제될 때, Amazon EFS는 호출자의 자격 증명을 사용하여 KMS 호출을 수행하기 위해 Forward Access Session(FAS)을 사용합니다. CloudTrail 로그에서 `kms:CreateGrant` 직접 호출은 파일 시스템 또는 복제를 생성한 것과 동일한 사용자 ID에 의해 수행된 것으로 표시됩니다. CloudTrail에서 값이 `elasticfilesystem.amazonaws.com`인 `invokedBy` 필드를 확인하여 Amazon EFS 서비스 직접 호출을 식별할 수 있습니다. KMS 키의 리소스 정책은 FAS가 호출을 수행할 수 있도록 `CreateGrant` 작업을 허용해야 합니다.

> **Important**
>
> 권한 부여에 대한 제어를 관리하고 언제든지 취소할 수 있습니다. 권한 부여를 취소하면 Amazon EFS가 향후 작업을 위해 KMS 키에 액세스할 수 없습니다. 자세한 내용은 AWS Key Management Service 개발자 안내서의 권한 부여 중지 및 취소를 참조하세요.

고객 관리형 KMS 키를 사용하는 경우 리소스 정책은 Amazon EFS 서비스 보안 주체도 허용하고 특정 서비스 엔드포인트에 대한 액세스를 제한하는 `kms:ViaService` 조건을 포함해야 합니다. 예제:

```
"kms:ViaService":
```

```
    "elasticfilesystem.us-east-2.amazonaws.com"
```

Amazon EFS는 저장 데이터 및 메타데이터 암호화에 업계 표준인 AES-256 암호화 알고리즘을 사용합니다.

Amazon EFS의 KMS 키 정책에 대한 자세한 내용은 Amazon EFS에 AWS KMS 키 사용 섹션을 참조하세요.

새 파일 시스템에 유휴 시 데이터 암호화 적용

AWS Identity and Access Management (IAM)의 ID 기반 정책에서 `elasticfilesystem:Encrypted` IAM 조건 키를 사용하여 사용자가 EFS 파일 시스템을 생성할 때 유휴 시 암호화를 강제 적용할 수 있습니다. 조건 키 사용에 관한 자세한 내용은 예: 암호화된 파일 시스템 생성 적용 섹션을 참조하세요.

또한 내부에서 서비스 제어 정책(SCPs 정의 AWS Organizations 하여 조직의 모든 AWS 계정 에 대해 Amazon EFS 암호화를 적용할 수 있습니다. 의 서비스 제어 정책에 대한 자세한 내용은 AWS Organizations 사용 설명서의 서비스 제어 정책을 AWS Organizations참조하세요.

전송 중 데이터 암호화

Amazon EFS는 Transport Layer Security(TLS)를 통한 전송 중 데이터 암호화를 지원합니다. 전송 중 데이터 암호화가 EFS 파일 시스템의 탑재 옵션으로 선언되면 Amazon EFS는 파일 시스템을 탑재할 때 EFS 파일 시스템과 안전한 TLS 연결을 설정합니다. 모든 NFS 트래픽은 이 암호화된 연결을 통해 라우팅됩니다.

전송 중 암호화 작동 방식

EFS 탑재 도우미를 사용하여 파일 시스템을 탑재하는 것이 좋습니다. NFS `mount`로 탑재하는 것보다 탑재 프로세스가 간단하기 때문입니다. EFS 탑재 도우미는 efs-proxy(efs-utils 버전 2.0.0 이상) 또는 stunnel(efs-utils 이전 버전)을 사용하여 EFS 파일 시스템과의 안전한 TLS 연결을 설정하여 프로세스를 관리합니다.

탑재 도우미를 사용하지 않아도 전송 중 데이터 암호화를 활성화할 수 있습니다. 이를 수행하는 단계는 다음과 같습니다.

탑재 도우미 없이 전송 중 데이터 암호화 활성화 방법

1. `stunnel`을 다운로드해 설치하고, 애플리케이션이 수신 대기하는 포트를 기록합니다. 자세한 내용은 `stunnel` 업그레이드 단원을 참조하십시오.

2. `stunnel`을 실행해 TLS를 사용하는 포트 2049의 EFS 파일 시스템에 연결합니다.
3. NFS 클라이언트를 사용해 `localhost:`*`port`*을 탑재합니다. *`port`*는 첫 번째 단계에서 기록한 포트입니다.

연결 별로 전송 중 데이터 암호화를 구성했기 때문에, 구성한 각 탑재에서는 전용 `stunnel` 프로세스가 인스턴스에서 실행됩니다. 기본적으로 탑재 도우미가 사용하는 stunnel 프로세스는 로컬 포트 20049에서 20449 사이에서 리스너로 작동하며, Amazon EFS에는 포트 2049로 연결합니다.

> **Note**
>
> 기본적으로 TLS와 함께 EFS 탑재 도우미를 사용하는 경우 Online Certificate Status Protocol(OCSP) 및 인증서 호스트 이름 확인을 사용합니다. EFS 탑재 도우미는 TLS 기능에 stunnel 프로그램을 사용합니다. 일부 Linux 버전에는 이 TLS 기능을 기본값으로 지원하지 않는 stunnel 버전이 포함되어 있습니다. 이러한 Linux 버전 중 하나를 사용하는 경우 TLS를 사용해서 EFS 파일 시스템을 탑재할 수 없습니다.
>
> amazon-efs-utils 패키지를 설치해 시스템 stunnel 버전을 업그레이드한 후 stunnel 업그레이드 단원을 참조하십시오.
>
> 암호화 관련 문제는 암호화 문제 해결을 참조하세요.

전송 중 데이터 암호화를 사용하는 경우 NFS 클라이언트 설정이 변경됩니다. 능동적으로 탑재한 파일 시스템을 검사할 때, 다음 예와 같이 127.0.0.1이나 `localhost`에 탑재된 파일 시스템을 확인할 수 있습니다.

```
$ mount | column -t
127.0.0.1:/  on  /home/ec2-user/efs        type  nfs4
 (rw,relatime,vers=4.1,rsize=1048576,wsize=1048576,namlen=255,hard,proto=tcp,port=20127,timeo=6
```

TLS와 EFS 탑재 도우미로 탑재를 할 때 로컬 포트로 탑재되도록 NFS 클라이언트를 다시 구성합니다. EFS 탑재 도우미가 이 로컬 포트에서 수신 대기하는 클라이언트 `stunnel` 프로세스를 시작하며, `stunnel` 프로세스는 TLS를 사용하여 EFS 파일 시스템에 대해 암호화된 연결을 엽니다. EFS 탑재 도우미가 암호화된 연결과 연결된 구성에 대한 설정과 유지 관리를 맡습니다.

다음 명령을 사용해 로컬 탑재 지점에 해당되는 Amazon EFS 파일 시스템 ID를 파악할 수 있습니다. *`efs-mount-point`*를 파일 시스템을 탑재시킨 로컬 경로로 바꿔야 합니다.

```
grep -E "Successfully mounted.*efs-mount-point" /var/log/amazon/efs/mount.log | tail -1
```

전송 중 데이터 암호화에 EFS 탑재 도우미를 사용하면 amazon-efs-mount-watchdog라는 프로세스가 생성됩니다. 이 프로세스는 각 탑재의 stunnel 프로세스를 실행시키고, EFS 파일 시스템의 탑재가 해제되었을 때 stunnel을 중단합니다. 어떤 이유로 stunnel 프로세스가 예기치 않게 종료된 경우, 이 감시 프로세스가 stunnel 프로세스를 다시 시작합니다.

Amazon EFS에 AWS KMS 키 사용

Amazon EFS는 키 관리를 위해 AWS Key Management Service (AWS KMS)와 통합됩니다. Amazon EFS는 고객 관리형 키를 사용하여 다음과 같은 방식으로 파일 시스템을 암호화합니다.

- 유휴 메타데이터 암호화 – Amazon EFS는 Amazon EFS용 AWS 관리형 키 인 aws/elasticfilesystem을 사용해 파일 시스템 메타데이터(예: 파일 이름, 디렉터리 이름, 디렉터리 내용)를 암호화하고 암호를 해제합니다.
- 저장 데이터 암호화 - 고객 관리형 키를 선택해 파일 데이터(파일의 내용)를 암호화하고 암호를 해제합니다. 이 고객 관리형 키에 대한 권한을 활성화, 비활성화 또는 취소할 수 있습니다. 이 고객 관리형 키는 다음 두 유형 중 하나일 수 있습니다.
  - AWS 관리형 키 Amazon EFS용 - 기본 고객 관리형 키인 입니다aws/elasticfilesystem. 고객 관리형 키를 생성하고 저장하는 데 요금이 부과되는 것은 아니지만, 사용 요금이 있습니다. 자세한 내용은 AWS Key Management Service 요금을 참조하세요.
  - 고객 관리형 키 – 여러 사용자나 서비스에 대한 키 정책 및 권한을 구성할 수 있는 가장 유연한 KMS 키입니다. 고객 관리형 키 생성에 대한 자세한 내용은 개발자 안내서의 키 생성을 참조하세요. AWS Key Management Service

    고객 관리형 키를 데이터 암호화 및 암호화 해제를 위해 사용하면 키 교체를 활성화할 수 있습니다. 키 교체를 활성화하면가 1년에 한 번 AWS KMS 자동으로 키를 교체합니다. 또한 고객 관리형 키를 사용하면 고객 관리형 키에 대한 액세스를 비활성화, 재활성화, 삭제, 취소하는 시기를 선택할 수 있습니다. 자세한 내용은 Amazon EFS에 AWS KMS 키 사용 단원을 참조하십시오.

> **Important**
>
> Amazon EFS는 대칭적인 고객 관리형 키만 허용합니다. Amazon EFS에서는 비대칭 고객 관리형 키를 사용할 수 없습니다.

유휴 데이터 암호화 및 암호화 해제는 투명하게 처리됩니다. 그러나 Amazon EFS와 관련된 AWS 계정 IDs AWS KMS 작업과 관련된 AWS CloudTrail 로그에 표시됩니다. 자세한 내용은 파일 시스템 유휴 암호화용 Amazon EFS 로그 파일 항목 단원을 참조하십시오.

에 대한 Amazon EFS 키 정책 AWS KMS

키 정책은 고객 관리형 키(CMK)에 대한 액세스를 제어하는 기본적인 방법입니다. 키 정책에 대한 자세한 내용은AWS Key Management Service 개발자 안내서의 AWS KMS의 키 정책을 참조하세요. 다음은 Amazon EFS가 암호화된 저장 파일 시스템을 위해 필수적이거나 기타를 지원하는 모든 AWS KMS 관련 권한을 나열한 목록입니다.

- kms:Encrypt – (선택 사항)일반 텍스트를 사이퍼텍스트로 암호화합니다. 이 권한은 기본 키 정책에 포함되어 있습니다.
- kms:Decrypt – (필수 사항) 사이퍼텍스트를 암호화 해제합니다. 사이퍼텍스트는 이전에 암호화한 일반 텍스트입니다. 이 권한은 기본 키 정책에 포함되어 있습니다.
- kms:ReEncrypt – (선택 사항)클라이언트 측에서 데이터의 일반 텍스트를 노출하지 않으면서 새 고객 관리형 키로 서버 측의 데이터를 암호화합니다. 먼저 데이터를 복호화한 후 다시 암호화합니다. 이 권한은 기본 키 정책에 포함되어 있습니다.
- kms:GenerateDataKeyWithoutPlaintext – (필수 사항) 고객 관리형 키로 암호화된 데이터 암호화 키를 반환합니다. 이 권한은 kms:GenerateDataKey* 아래 기본 키 정책에 포함되어 있습니다.
- kms:CreateGrant – (필수 사항)특정 조건 하에 키를 사용할 수 있는 사람을 지정할 수 있도록 키에 권한을 추가합니다. 이런 권한 부여는 키 정책을 대체하는 권한 메커니즘입니다. 권한 부여에 대한 자세한 내용은 AWS Key Management Service 개발자 안내서의 AWS KMS의 권한 부여를 참조하세요. 이 권한은 기본 키 정책에 포함되어 있습니다.
- kms:DescribeKey – (필수 사항)지정한 고객 관리형 키에 대한 세부 정보를 제공합니다. 이 권한은 기본 키 정책에 포함되어 있습니다.
- kms:ListAliases – (선택 사항)계정의 모든 키 별칭을 나열합니다. 콘솔을 사용해 암호화된 파일 시스템을 생성하는 경우, 이 권한이 KMS 키 선택 목록을 채웁니다. 최상의 사용자 경험을 제공하기 위해 이 권한을 사용하는 것이 좋습니다. 이 권한은 기본 키 정책에 포함되어 있습니다.

AWS 관리형 키 Amazon EFS KMS 정책용

Amazon EFS용에 AWS 관리형 키 대한 KMS 정책 JSONaws/elasticfilesystem은 다음과 같습니다.

```
{
    "Version": "2012-10-17",
    "Id": "auto-elasticfilesystem-1",
    "Statement": [
        {
```

```
            "Sid": "Allow access to EFS for all principals in the account that are
 authorized to use EFS",
            "Effect": "Allow",
            "Principal": {
                "AWS": "*"
            },
            "Action": [
                "kms:Encrypt",
                "kms:Decrypt",
                "kms:ReEncrypt*",
                "kms:GenerateDataKey*",
                "kms:CreateGrant",
                "kms:DescribeKey"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "kms:ViaService": "elasticfilesystem.us-east-2.amazonaws.com",
                    "kms:CallerAccount": "111122223333"
                }
            }
        },
        {
            "Sid": "Allow direct access to key metadata to the account",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:root"
            },
            "Action": [
                "kms:Describe*",
                "kms:Get*",
                "kms:List*",
                "kms:RevokeGrant"
            ],
            "Resource": "*"
        }
    ]
}
```

주요 상태 및 그 효과

KMS 키의 상태는 암호화된 파일 시스템에 대한 액세스에 직접적인 영향을 미칩니다.

활성화됨

일반 작업 - 파일 시스템에 대한 전체 읽기 및 쓰기 액세스

비활성화됨

짧은 기간이 지나면 파일 시스템에 액세스할 수 없게 됩니다. 다시 활성화할 수 있습니다.

삭제 보류 중

파일 시스템에 액세스할 수 없게 됩니다. 삭제는 대기 기간 동안 취소할 수 있습니다.

삭제됨

파일 시스템에 영구적으로 액세스할 수 없습니다. 이 작업은 되돌릴 수 없습니다.

> **Warning**
>
> 파일 시스템에 사용되는 KMS 키를 비활성화 또는 삭제하거나 키에 대한 Amazon EFS 액세스를 취소하면 파일 시스템에 액세스할 수 없게 됩니다. 백업이 없는 경우 데이터가 손실될 수 있습니다. 암호화 키를 변경하기 전에 항상 적절한 백업 절차를 마련해야 합니다.

암호화 문제 해결

아래에서는 Amazon EFS의 암호화 문제 해결에 대한 정보를 알아봅니다.

- 전송 중 데이터 암호화를 사용한 탑재에 실패
- 전송 중 데이터 암호화를 사용한 탑재가 중단됨
- 유휴 암호화된 파일 시스템을 생성할 수 없음
- 사용할 수 없는 암호화된 파일 시스템

전송 중 데이터 암호화를 사용한 탑재에 실패

기본적으로 전송 계층 보안(TLS)으로 Amazon EFS 탑재 도우미를 사용할 때 호스트 이름 확인을 적용합니다. 일부 시스템은 이 기능을 지원하지 않습니다(예: Red Hat Enterprise Linux 또는 CentOS 사용하는 경우). 이 경우 TLS를 사용한 EFS 파일 시스템 탑재에 실패합니다.

취할 조치

클라이언트의 stunnel 버전을 호스트 이름 확인을 지원하는 버전으로 업그레이드하는 것이 좋습니다. 자세한 내용은 `stunnel` 업그레이드 단원을 참조하십시오.

전송 중 데이터 암호화를 사용한 탑재가 중단됨

가능성은 낮지만 클라이언트 측 이벤트로 인해 Amazon EFS 파일 시스템에 대한 암호화된 연결이 해제되거나 중단될 수도 있습니다.

취할 조치

전송 중 데이터 암호화가 적용된 Amazon EFS 파일 시스템 연결이 중단되는 경우 다음 단계를 수행합니다.

1. stunnel 서비스가 클라이언트에서 실행되고 있는지 확인합니다.
2. 감시 애플리케이션 `amazon-efs-mount-watchdog`가 클라이언트에서 실행되고 있는지 확인합니다. 다음 명령으로 이 애플리케이션 실행 여부를 확인할 수 있습니다.

   ```
   ps aux | grep [a]mazon-efs-mount-watchdog
   ```

3. 지원 로그를 확인합니다. 자세한 내용은 지원 로그 얻기 단원을 참조하십시오.
4. stunnel 로그를 활성화한 후 여기의 정보를 확인하는 방법도 있습니다. `/etc/amazon/efs/efs-utils.conf`의 로그 구성을 변경해 stunnel 로그를 활성화할 수 있습니다. 그러나 이렇게 하기 위해서는 탑재 도우미가 있는 파일 시스템의 탑재를 해제한 후 다시 탑재해야 합니다. 그래야 변경 사항이 적용됩니다.

   > **Important**
   >
   > 활성화된 stunnel 로그가 파일 시스템에서 상당한 공간을 사용할 수 있습니다.

중단이 계속되면 AWS Support에 문의하십시오.

유휴 암호화된 파일 시스템을 생성할 수 없음

유휴 암호화된 새 파일 시스템 생성을 시도했습니다. 그러나를 AWS KMS 사용할 수 없다는 오류 메시지가 표시됩니다.

취할 조치

이 오류는 드물게에서 일시적으로 사용할 수 없게 AWS KMS 되는 경우에 발생할 수 있습니다 AWS 리전. 이 경우가 전체 가용성으로 AWS KMS 돌아갈 때까지 기다린 다음 파일 시스템을 다시 생성해 보십시오.

사용할 수 없는 암호화된 파일 시스템

암호화된 파일 시스템이 계속 NFS 서버 오류를 반환합니다. 이러한 오류는 EFS가 다음 이유 중 하나로 AWS KMS 인해에서 마스터 키를 검색할 수 없을 때 발생할 수 있습니다.

- 키가 비활성화되어 있습니다.
- 키가 삭제됐습니다.
- Amazon EFS가 키를 사용할 수 있는 권한이 취소되었습니다.
- AWS KMS 를 일시적으로 사용할 수 없습니다.

취할 조치

먼저 AWS KMS 키가 활성화되어 있는지 확인합니다. 콘솔에서 키를 확인하면 됩니다. 자세한 내용은 AWS Key Management Service 개발자 안내서에서 키 보기를 참조하세요.

키가 활성화되어 있지 않으면 활성화합니다. 자세한 내용은AWS Key Management Service 개발자 안내서의 키 활성화 및 비활성화를 참조하세요.

키가 삭제 대기 중인 경우, 이 상태가 키를 비활성화합니다. 삭제를 취소한 후 키를 다시 활성화할 수 있습니다. 자세한 내용은AWS Key Management Service 개발자 안내서에서 키 삭제 예약 및 취소를 참조하세요.

키가 활성화되어 있지만 여전히 문제가 발생하거나 키를 다시 활성화하는 데 문제가 발생하면 AWS Support에 문의하세요.

인터네트워크 개인 정보 보호

이 주제에서는 Amazon EFS가 서비스에서 다른 위치로의 연결을 어떻게 보호하는지 설명합니다.

서비스와 온프레미스 클라이언트 및 애플리케이션 간의 트래픽

프라이빗 네트워크와 간에 AWS는 두 가지 연결 옵션이 있습니다.

- AWS Site-to-Site VPN 연결입니다. 자세한 내용은 란 무엇입니까 AWS Site-to-Site VPN?를 참조하세요.
- Direct Connect 연결입니다. 자세한 내용은 란 무엇입니까 Direct Connect?를 참조하세요.

네트워크를 통한 Amazon EFS 액세스는 AWS 게시된 APIs. 클라이언트가 Transport Layer 1.2 이상을 지원해야 합니다. TLS 1.3 이상을 사용하는 것이 좋습니다. 클라이언트는 DHE(Ephemeral Diffie-

Hellman) 또는 ECDHE(Elliptic Curve Diffie-Hellman Ephemeral)와 같은 PFS(전달 완전 보안)가 포함된 암호 제품군도 지원해야 합니다. Java 7 이상의 최신 시스템은 대부분 이러한 모드를 지원합니다. 또한, 반드시 액세스 키 ID와 IAM 보안 주체와 관련된 비밀 액세스 키를 사용하여 요청에 서명하거나 AWS Security Token Service (AWS STS)를 사용하여 요청에 서명할 수 있는 임시 보안 인증 정보를 생성할 수 있습니다.

VPC와 Amazon EFS API 간의 트래픽

Virtual Private Cloud(VPC)와 Amazon EFS API 간에 프라이빗 연결을 설정하기 위해 인터페이스 VPC 엔드포인트를 생성할 수 있습니다. 인터넷을 통해 트래픽을 보내지 않고 이 연결을 사용하여 VPC에서 Amazon EFS API를 호출할 수 있습니다. 엔드포인트는 인터넷 게이트웨이, NAT 인스턴스 또는 가상 프라이빗 네트워크(VPN) 연결 없이도 Amazon EFS API에 대한 안전한 연결을 제공합니다. 자세한 내용은 Amazon EFS에서 인터페이스 VPC 엔드포인트 작업 단원을 참조하십시오.

동일한 리전의 AWS 리소스 간 트래픽

Amazon EFS용 Amazon Virtual Private Cloud(Amazon VPC) 엔드포인트는 VPC 내의 논리적 엔터티로서, Amazon EFS에만 연결을 허용합니다. Amazon VPC는 Amazon EFS로 요청을 라우팅하고, 대응을 다시 VPC로 라우팅합니다. 자세한 내용은 Amazon VPC 사용 설명서의 VPC 엔드포인트를 참조하세요.

Amazon EFS의 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 어떤 사용자가 Amazon EFS 리소스를 사용할 수 있도록 인증(로그인)되고 권한이 부여(권한 있음)될 수 있는지 제어합니다. IAM은 추가 비용 없이 사용할 수 AWS 서비스 있는 입니다.

주제

- 대상
- ID를 통한 인증
- 정책을 사용하여 액세스 관리
- Amazon Elastic File System과 IAM의 작동 방식
- Amazon Elastic File System의 자격 증명 기반 정책 예제
- Amazon EFS의 리소스 기반 정책 예제
- AWS Amazon EFS에 대한 관리형 정책

- Amazon EFS에서 태그 사용
- Amazon EFS에 대해 서비스 연결 역할 사용
- Amazon Elastic File System 자격 증명 및 액세스 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 Elastic File System 자격 증명 및 액세스 문제 해결)
- 서비스 관리자 - 사용자 액세스 결정 및 권한 요청 제출(Amazon Elastic File System과 IAM의 작동 방식 참조)
- IAM 관리자 - 액세스를 관리하기 위한 정책 작성(Amazon Elastic File System의 자격 증명 기반 정책 예제 참조)

ID를 통한 인증

인증은 AWS 자격 증명으로에 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 Elastic File System과 IAM의 작동 방식

IAM을 사용하여 Amazon EFS에 대한 액세스를 관리하기 전에 Amazon EFS에서 사용할 수 있는 IAM 기능에 대해 알아봅니다.

Amazon Elastic File System과 함께 사용할 수 있는 IAM 기능

IAM 특성	Amazon EFS 지원
자격 증명 기반 정책	예
리소스 기반 정책	예
정책 작업	예
정책 리소스	예
정책 조건 키(서비스별)	예
ACL	아니요
ABAC(정책 내 태그)	부분적
임시 자격 증명	예
엔터티 권한	예
서비스 역할	예
서비스 연결 역할	예

Amazon EFS 및 기타 AWS 서비스에서 대부분의 IAM 기능을 사용하는 방법을 전체적으로 알아보려면 IAM 사용 설명서의 AWS IAM으로 작업하는 서비스를 참조하세요.

Amazon EFS의 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 EFS의 자격 증명 기반 정책 예

Amazon EFS 자격 증명 기반 정책 예제를 보려면 Amazon Elastic File System의 자격 증명 기반 정책 예제 섹션을 참조하세요.

Amazon EFS 내의 리소스 기반 정책

리소스 기반 정책 지원: 예

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.

리소스 정책을 사용하여 파일 시스템 데이터 액세스를 제어하는 방법에 대한 자세한 내용은 IAM을 사용하여 파일 시스템에 대한 액세스 제어 섹션을 참조하세요. 파일 시스템에 리소스 기반 정책을 연결하는 방법은 파일 시스템 정책 생성 섹션을 참조하세요.

Amazon EFS에 사용되는 리소스 기반 정책 예제

Amazon EFS 리소스 기반 정책의 예제를 보려면 Amazon EFS의 리소스 기반 정책 예제 섹션을 참조하세요.

Amazon EFS의 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.

Amazon EFS 작업 목록을 보려면 서비스 승인 참조의 Amazon Elastic File System에서 정의한 작업을 참조하세요.

Amazon EFS의 정책 작업은 작업 앞에 다음 접두사를 사용합니다.

```
elasticfilesystem
```

단일 명령문에서 여러 작업을 지정하려면 다음과 같이 쉼표로 구분합니다.

```
"Action": [
      "elasticfilesystem:action1",
      "elasticfilesystem:action2"
         ]
```

Amazon EFS 자격 증명 기반 정책 예제를 보려면 Amazon Elastic File System의 자격 증명 기반 정책 예제 섹션을 참조하세요.

Amazon SNS의 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": "*"
```

Amazon EFS 리소스 유형 및 해당 ARN의 목록을 보려면 서비스 권한 부여 참조에서 Amazon Elastic File System에서 정의한 리소스를 참조하세요. 각 리소스의 ARN을 지정할 수 있는 작업을 알아보려면 Amazon Elastic File System에서 정의한 작업을 참조하세요.

Amazon EFS 자격 증명 기반 정책 예제를 보려면 Amazon Elastic File System의 자격 증명 기반 정책 예제 섹션을 참조하세요.

Amazon EFS의 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.

Amazon EFS 조건 키 목록을 보려면 서비스 승인 참조의 Amazon Elastic File System에 사용되는 조건 키를 참조하세요. 조건 키를 사용할 수 있는 작업과 리소스를 알아보려면 Amazon Elastic File System에서 정의한 작업을 참조하세요.

Amazon EFS 자격 증명 기반 정책 예제를 보려면 Amazon Elastic File System의 자격 증명 기반 정책 예제 섹션을 참조하세요.

Amazon EFS의 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 EFS의 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 EFS에서 임시 보안 인증 사용

임시 자격 증명 지원: 예

임시 자격 증명은 AWS 리소스에 대한 단기 액세스를 제공하며 페더레이션을 사용하거나 역할을 전환할 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 작업하는AWS 서비스 섹션을 참조하세요.

Amazon EFS에 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 EFS의 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 EFS 기능이 중단될 수 있습니다. Amazon EFS에서 관련 지침을 제공하는 경우에만 서비스 역할을 편집합니다.

Amazon EFS의 서비스 연결 역할

서비스 연결 역할 지원: 예

서비스 연결 역할은에 연결된 서비스 역할의 한 유형입니다 AWS 서비스. 서비스는 사용자를 대신하여 작업을 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.

Amazon EFS 서비스 연결 역할 생성 또는 관리에 대한 자세한 정보는 Amazon EFS에 대해 서비스 연결 역할 사용 섹션을 참조하세요.

Amazon Elastic File System의 자격 증명 기반 정책 예제

기본적으로 사용자 및 역할은 Amazon EFS 리소스를 생성하거나 수정할 수 있는 권한이 없습니다. 사용자에게 사용자가 필요한 리소스에서 작업을 수행할 권한을 부여하려면 IAM 관리자가 IAM 정책을 생성하면 됩니다.

이러한 예제 JSON 정책 문서를 사용하여 IAM ID 기반 정책을 생성하는 방법을 알아보려면 IAM 사용 설명서의 IAM 정책 생성(콘솔)을 참조하세요.

각 리소스 유형에 대한 ARN 형식을 비롯하여 Amazon EFS에서 정의되는 작업 및 리소스 유형에 대한 자세한 내용은 서비스 권한 부여 참조의 Amazon Elastic File System에 사용되는 작업, 리소스 및 조건 키를 참조하세요.

주제

- 정책 모범 사례
- Amazon EFS 콘솔 사용
- 예: 사용자가 자신이 권한을 볼 수 있도록 허용
- 예: 암호화된 파일 시스템 생성 적용
- 예: 암호화되지 않은 파일 시스템 생성 적용

정책 모범 사례

ID 기반 정책에 따라 계정에서 사용자가 Amazon EFS 리소스를 생성, 액세스 또는 삭제할 수 있는지 여부가 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.

Amazon EFS 콘솔 사용

Amazon Elastic File System 콘솔에 액세스하려면 최소한의 권한 집합이 있어야 합니다. 이러한 권한은에서 Amazon EFS 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 EFS 콘솔을 계속 사용할 수 있도록 하려면 Amazon EFS `AmazonElasticFileSystemReadOnlyAccess` AWS 관리형 정책도 엔터티에 연결합니다. 자세한 내용은 IAM 사용 설명서의 사용자에게 권한 추가를 참조하세요.

AmazonElasticFileSystemReadOnlyAccess 및 기타 Amazon EFS 관리 서비스 정책은 AWS Amazon EFS에 대한 관리형 정책 섹션에서 확인할 수 있습니다.

예: 사용자가 자신이 권한을 볼 수 있도록 허용

이 예제는 IAM 사용자가 자신의 사용자 ID에 연결된 인라인 및 관리형 정책을 볼 수 있도록 허용하는 정책을 생성하는 방법을 보여 줍니다. 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": "*"
        }
    ]
}
```

예: 암호화된 파일 시스템 생성 적용

다음 예제에서는 보안 주체에게 암호화된 파일 시스템만 생성하도록 권한을 부여하는 자격 증명 기반 정책을 보여 줍니다.

```
{
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "elasticfilesystem:CreateFileSystem",
            "Condition": {
                "Bool": {
                    "elasticfilesystem:Encrypted": "true"
                }
            },
            "Resource": "*"
        }
    ]
}
```

암호화되지 않은 파일 시스템을 만들려는 사용자에게 이 정책을 할당하면 요청이 실패합니다. 사용자는 , AWS Management Console AWS CLI또는 AWS API 또는 SDK를 사용하는지 여부에 관계없이 다음과 유사한 메시지를 볼 수 있습니다.

```
User: arn:aws:iam::111122223333:user/username is not authorized to
      perform: elasticfilesystem:CreateFileSystem on the specified resource.
```

예: 암호화되지 않은 파일 시스템 생성 적용

다음 예제에서는 보안 주체에게 암호화되지 않은 파일 시스템만 생성하도록 권한을 부여하는 자격 증명 기반 정책을 보여 줍니다.

```
{
      "Statement": [
        {
            "Effect": "Allow",
            "Action": "elasticfilesystem:CreateFileSystem",
            "Condition": {
                "Bool": {
                    "elasticfilesystem:Encrypted": "false"
                }
```

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

암호화된 파일 시스템을 만들려는 사용자에게 이 정책을 할당하면 요청이 실패합니다. 사용자는 , AWS Management Console AWS CLI또는 AWS API 또는 SDK를 사용하는지 여부에 관계없이 다음과 유사한 메시지를 볼 수 있습니다.

```
User: arn:aws:iam::111122223333:user/username is not authorized to
      perform: elasticfilesystem:CreateFileSystem on the specified resource.
```

AWS Organizations 서비스 제어 정책(SCP)을 생성하여 암호화되거나 암호화되지 않은 EFS 파일 시스템의 생성을 적용할 수도 있습니다. 의 서비스 제어 정책에 대한 자세한 내용은 AWS Organizations 사용 설명서의 서비스 제어 정책을 AWS Organizations참조하세요.

Amazon EFS의 리소스 기반 정책 예제

이 단원에서는 다양한 Amazon EFS 작업에 대한 권한을 부여하거나 거부하는 파일 시스템 정책의 예를 제공합니다. EFS 파일 시스템 정책은 20,000자로 제한됩니다. 리소스 기반 정책의 요소에 대한 자세한 내용은 Amazon EFS 내의 리소스 기반 정책 섹션을 참조하세요.

> **Important**
>
> 파일 시스템 정책에서 개별 IAM 사용자 또는 역할에 권한을 부여하는 경우, 정책이 파일 시스템에 적용되는 동안에는 해당 사용자 또는 역할을 삭제하거나 다시 생성하지 마세요. 그러한 경우, 해당 사용자 또는 역할이 파일 시스템에서 실제적으로 잠겨서 액세스할 수 없게 됩니다. 자세한 내용은 IAM 사용 설명서의 보안 주체 지정을 참조하세요.

파일 시스템 정책을 생성하는 방법에 대한 자세한 내용은 파일 시스템 정책 생성 섹션을 참조하세요.

주제

- 예: 특정 AWS 역할에 대한 읽기 및 쓰기 액세스 권한 부여
- 예: 읽기 전용 액세스 권한 부여
- 예: EFS 액세스 포인트에 대한 액세스 권한 부여

예: 특정 AWS 역할에 대한 읽기 및 쓰기 액세스 권한 부여

이 예제에서, EFS 파일 시스템 정책에는 다음과 같은 특징이 있습니다.

- 효과는 `Allow`입니다.
- 보안 주체는 AWS 계정에서 Testing_Role로 설정됩니다.
- 작업은 `ClientMount`(읽기) 및 `ClientWrite`로 설정됩니다.
- 권한 부여 조건은 `AccessedViaMountTarget`로 설정되어 있습니다.

```
{
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:role/Testing_Role"
            },
            "Action": [
                "elasticfilesystem:ClientWrite",
                "elasticfilesystem:ClientMount"
            ],
            "Resource": "arn:aws:elasticfilesystem:us-east-2:111122223333:file-system/
fs-1234abcd",
            "Condition": {
                "Bool": {
                    "elasticfilesystem:AccessedViaMountTarget": "true"
                }
            }
        }
    ]
}
```

예: 읽기 전용 액세스 권한 부여

다음 파일 시스템 정책은 `EfsReadOnly` IAM 역할에 `ClientMount` 권한 또는 읽기 전용 권한만 부여합니다.

```
{
    "Id": "read-only-example-policy02",
    "Statement": [
        {
```

```
            "Sid": "efs-statement-example02",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:role/EfsReadOnly"
            },
            "Action": [
                "elasticfilesystem:ClientMount"
            ],
            "Resource": "arn:aws:elasticfilesystem:us-east-2:111122223333:file-system/
fs-12345678"
        }
    ]
}
```

특정 관리 워크스테이션을 제외한 모든 보안 주체에 대한 루트 액세스 거부를 포함하여 추가 파일 시스템 정책을 설정하는 방법은 NFS 클라이언트에 대한 IAM 권한 부여를 사용하여 루트 스쿼싱 활성화 섹션을 참조하세요.

예: EFS 액세스 포인트에 대한 액세스 권한 부여

EFS 액세스 정책을 사용하여 EFS 파일 시스템의 공유 파일 기반 데이터 세트에 대한 애플리케이션별 보기를 NFS 클라이언트에 제공할 수 있습니다. 파일 시스템 정책을 사용하여 파일 시스템에 대한 액세스 포인트 권한을 부여합니다.

이 파일 정책 예제에서는 조건 요소를 사용하여 해당 ARN으로 식별되는 특정 액세스 포인트에 파일 시스템에 대한 모든 액세스 권한을 부여합니다.

EFS 액세스 포인트에 대한 자세한 내용은 액세스 포인트 작업 섹션을 참조하세요.

```
{
    "Id": "access-point-example03",
    "Statement": [
        {
            "Sid": "access-point-statement-example03",
            "Effect": "Allow",
            "Principal": {"AWS": "arn:aws:iam::555555555555:role/
EfsAccessPointFullAccess"},
            "Action": "elasticfilesystem:Client*",
            "Resource": "arn:aws:elasticfilesystem:us-east-2:111122223333:file-system/
fs-12345678",
            "Condition": {
                "StringEquals": {
```

```
                    "elasticfilesystem:AccessPointArn":"arn:aws:elasticfilesystem:us-
east-2:555555555555:access-point/fsap-12345678" }
            }
        }
    ]
}
```

AWS Amazon EFS에 대한 관리형 정책

AWS 관리형 정책은에서 생성하고 관리하는 독립 실행형 정책입니다 AWS. AWS 관리형 정책은 사용자, 그룹 및 역할에 권한 할당을 시작할 수 있도록 많은 일반적인 사용 사례에 대한 권한을 제공하도록 설계되었습니다.

AWS 관리형 정책은 모든 AWS 고객이 사용할 수 있으므로 특정 사용 사례에 대해 최소 권한을 부여하지 않을 수 있습니다. 사용 사례에 고유한 고객 관리형 정책을 정의하여 권한을 줄이는 것이 좋습니다.

AWS 관리형 정책에 정의된 권한은 변경할 수 없습니다. 가 관리형 정책에 정의된 권한을 AWS 업데이트하는 AWS 경우 업데이트는 정책이 연결된 모든 보안 주체 자격 증명(사용자, 그룹 및 역할)에 영향을 미칩니다. AWS AWS 서비스 는 새가 시작되거나 기존 서비스에 새 API 작업을 사용할 수 있게 되면 AWS 관리형 정책을 업데이트할 가능성이 높습니다.

자세한 내용은 IAM 사용자 가이드의 AWS 관리형 정책을 참조하세요.

AWS 관리형 정책: AWSServiceRoleForAmazonElasticFileSystem

Amazon EFS는 라는 서비스 연결 역할을 `AWSServiceRoleForAmazonElasticFileSystem` 사용하여 Amazon EFS가 사용자를 대신하여 AWS 리소스를 관리할 수 있도록 합니다. 이 역할은 역할을 수임하기 위해 `elasticfilesystem.amazonaws.com` 서비스를 신뢰합니다. 자세한 내용은 Amazon EFS에 대해 서비스 연결 역할 사용 단원을 참조하십시오.

AWS 관리형 정책: AmazonElasticFileSystemFullAccess

`AmazonElasticFileSystemFullAccess` 정책을 IAM ID에 연결할 수 있습니다.

이 정책은를 통해 Amazon EFS에 대한 전체 액세스 및 관련 AWS 서비스에 대한 액세스를 허용하는 관리 권한을 부여합니다 AWS Management Console.

권한 세부 정보

이 정책에는 다음 권한이 포함되어 있습니다.

- `elasticfilesystem` – 보안 주체가 Amazon EFS 콘솔에서 모든 작업을 수행할 수 있습니다. 또한 보안 주체가를 사용하여 (`elasticfilesystem:Backup`) 백업을 생성하고 복원 (`elasticfilesystem:Restore`)할 수 있습니다 AWS Backup.
- `cloudwatch` – 보안 주체가 Amazon EFS 콘솔의 지표에 대한 Amazon CloudWatch 파일 시스템 지표 및 경보를 설명할 수 있습니다.
- `ec2` – 보안 주체가 Amazon EFS 콘솔에서 네트워크 인터페이스를 생성, 삭제 및 설명하고, 네트워크 인터페이스 속성을 설명 및 수정하고, EFS 파일 시스템과 관련된 가용 영역, 보안 그룹, 서브넷, 가상 프라이빗 클라우드(VPC) 및 VPC 속성을 설명할 수 있습니다.
- `kms` - 보안 주체가 AWS Key Management Service (AWS KMS) 키의 별칭을 나열하고 Amazon EFS 콘솔에서 KMS 키를 설명할 수 있습니다.
- `iam` - Amazon EFS가 사용자를 대신하여 AWS 리소스를 관리할 수 있도록 서비스 연결 역할을 생성할 수 있는 권한을 부여합니다.
- `iam:PassRole` – Amazon EFS에 IAM 역할을 전달할 수 있는 권한을 부여합니다.

이 정책의 권한을 보려면 AWS 관리형 정책 참조의 AmazonElasticFileSystemFullAccess를 참조하세요.

AWS 관리형 정책: AmazonElasticFileSystemReadOnlyAccess

`AmazonElasticFileSystemReadOnlyAccess` 정책을 IAM ID에 연결할 수 있습니다.

이 정책은를 통해 Amazon EFS에 대한 읽기 전용 액세스 권한을 부여합니다 AWS Management Console.

권한 세부 정보

이 정책에는 다음 권한이 포함되어 있습니다.

- `elasticfilesystem` – 보안 주체가 계정 기본 설정, 백업 및 파일 시스템 정책, 수명 주기 구성, 탑재 대상 및 Amazon EFS 콘솔의 보안 그룹, 태그, 액세스 포인트 등 Amazon EFS 파일 시스템의 속성을 설명할 수 있습니다.
- `cloudwatch` – 보안 주체가 Amazon EFS 콘솔에서 CloudWatch 지표를 검색하고 지표에 대한 경보를 설명할 수 있습니다.
- `ec2` – 보안 주체가 Amazon EFS 콘솔에서 가용 영역, 네트워크 인터페이스 및 속성, 보안 그룹, 서브넷, VPC 및 해당 속성을 볼 수 있습니다.
- `kms` - 보안 주체가 Amazon EFS 콘솔에서 AWS KMS 키의 별칭을 나열할 수 있도록 허용합니다.

이 정책의 권한을 보려면 AWS 관리형 정책 참조의 AmazonElasticFileSystemReadOnlyAccess를 참조하세요.

AWS 관리형 정책: AmazonElasticFileSystemClientFullAccess

`AmazonElasticFileSystemClientFullAccess` 정책을 IAM 엔터티에 연결할 수 있습니다.

이 정책은 EFS 파일 시스템에 대한 읽기 및 쓰기 클라이언트 액세스를 부여합니다. 이 정책은 NFS 클라이언트가 EFS 파일 시스템에 탑재하고 읽고 쓸 수 있도록 허용합니다.

이 정책의 권한을 보려면 AWS 관리형 정책 참조의 AmazonElasticFileSystemClientFullAccess를 참조하세요.

AWS 관리형 정책: AmazonElasticFileSystemClientReadWriteAccess

`AmazonElasticFileSystemClientReadWriteAccess` 정책을 IAM 엔터티에 연결할 수 있습니다.

이 정책은 EFS 파일 시스템에 대한 읽기 및 쓰기 클라이언트 액세스를 부여합니다. 이 정책은 NFS 클라이언트가 EFS 파일 시스템에 탑재하고 읽고 쓸 수 있도록 허용합니다.

이 정책의 권한을 보려면 AWS 관리형 정책 참조의 AmazonElasticFileSystemClientReadWriteAccess를 참조하세요.

AWS 관리형 정책에 대한 Amazon EFS 업데이트

이 서비스가 이러한 변경 사항을 추적하기 시작한 이후부터 Amazon EFS의 AWS 관리형 정책 업데이트에 대한 세부 정보를 봅니다. 이 페이지의 변경 사항에 대한 자동 알림을 받아보려면 Amazon EFS 문서 기록 페이지에서 RSS 피드를 구독하세요.

변경	설명	Date
기존 정책 업데이트	정책: AmazonElasticFileSystemFullAccess Amazon EFS에서 다음을 추가했습니다. • `ReplicationRead` 및 `ReplicationWrite` 를 사용하여 복제를 위해 파일 시스템 데이터를 읽고 쓸 수 있는 권한을 부여합니다. • `iam:PassRole` 는 Amazon EFS가 복제 구성을 생성할 수 있도록 권한을 부여합니다.	2024년 11월 7일

변경	설명	Date
기존 정책 업데이트	정책: AmazonElasticFileSystemServiceRolePolicy Amazon EFS는 복제를 위해 파일 시스템 데이터를 읽고 쓸 수 있는 권한을 부여하기 위해 `ReplicationRead` 및 `ReplicationWrite` 를 추가했습니다.	2024년 11월 7일
기존 정책에 대한 업데이트	정책: AmazonElasticFileSystemReadOnlyAccess Amazon EFS는 복제를 위해 파일 시스템 데이터를 읽을 수 있는 권한을 부여하는 `ReplicationRead` 작업을 추가했습니다.	2024년 11월 7일
기존 정책에 대한 업데이트	정책: AmazonElasticFileSystemReadOnlyAccess Amazon EFS는 소스 및 대상 계정에 교차 계정 복제를 위한 파일 시스템에 대한 액세스 권한을 부여하는 새로운 권한을 추가했습니다.	2024년 8월 7일
기존 정책 업데이트	정책: AmazonElasticFileSystemFullAccess Amazon EFS는 보안 주체가 파일 시스템에서 보호를 비활성화 및 활성화할 수 있도록 허용하는 새로운 권한을 추가했습니다. Amazon EFS가 기존 파일 시스템에 복제할 수 있으려면 권한이 필요합니다.	2023년 11월 27일
기존 정책 업데이트	정책: AmazonElasticFileSystemServiceRolePolicy Amazon EFS는 보안 주체가 Amazon EFS 복제를 생성, 설명 및 삭제하고 Amazon EFS 파일 시스템을 생성할 수 있는 새로운 권한을 추가했습니다. Amazon EFS가 사용자를 대신하여 파일 시스템 복제 구성을 관리할 수 있도록 하려면 권한이 필요합니다.	2022년 1월 25일

변경	설명	Date
기존 정책에 대한 업데이트	정책: AmazonElasticFileSystemReadOnlyAccess Amazon EFS는 보안 주체가 Amazon EFS 복제에 대해 설명할 수 있는 새로운 권한을 추가했습니다. 사용자가 파일 시스템 복제 구성을 볼 수 있게 하려면 권한이 필요합니다.	2022년 1월 25일
기존 정책에 대한 업데이트	정책: AmazonElasticFileSystemFullAccess Amazon EFS는 보안 주체가 Amazon EFS 복제를 생성, 설명 및 삭제할 수 있는 새로운 권한을 추가했습니다. 사용자가 파일 시스템 복제 구성을 관리할 수 있게 하려면 권한이 필요합니다.	2022년 1월 25일
정책 추적 시작됨	정책: AmazonElasticFileSystemClientReadWriteAccess Amazon EFS 파일 시스템에 대한 읽기 및 쓰기 권한을 NFS 클라이언트에 부여합니다.	2022년 1월 3일
정책 추적 시작됨	정책: AmazonElasticFileSystemServiceRolePolicy Amazon EFS에 대한 서비스 연결 역할 권한	2021년 10월 8일
기존 정책에 대한 업데이트	정책: AmazonElasticFileSystemFullAccess Amazon EFS는 보안 주체가 Amazon EFS 계정 기본 설정을 수정하고 설명할 수 있는 새로운 권한을 추가했습니다. 사용자가 Amazon EFS 콘솔에서 계정 기본 설정을 보고 설정할 수 있게 하려면 권한이 필요합니다.	2021년 5월 7일
기존 정책에 대한 업데이트	정책: AmazonElasticFileSystemReadOnlyAccess Amazon EFS는 보안 주체가 Amazon EFS 계정 기본 설정을 설명할 수 있는 새로운 권한을 추가했습니다. 사용자가 Amazon EFS 콘솔에서 계정 기본 설정을 볼 수 있게 하려면 권한이 필요합니다.	2021년 5월 7일

변경	설명	Date
Amazon EFS에서 변경 사항 추적 시작	Amazon EFS는 AWS 관리형 정책에 대한 변경 사항 추적을 시작했습니다.	2021년 5월 7일

Amazon EFS에서 태그 사용

태그를 사용하여 Amazon EFS 리소스에 대한 액세스를 제어하고 ABAC(속성 기반 액세스 제어)를 구현할 수 있습니다. 자세한 내용은 다음을 참조하세요.

- EFS 리소스에 태그 지정
- 리소스의 태그를 기반으로 액세스 제어
- IAM 사용 설명서의 ABAC는 무엇입니까 AWS?

> **Note**
>
> Amazon EFS 복제는 ABAC(속성 기반 액세스 제어)에 대한 태그 사용을 지원하지 않습니다.

생성 중에 Amazon EFS 리소스에 태그를 적용하려면 사용자에게 AWS Identity and Access Management (IAM) 권한이 있어야 합니다.

생성 시 리소스 태그 지정에 대한 권한 부여

다음과 같은 생성 시 태그 지정 Amazon EFS API 작업을 사용하면 리소스를 생성할 때 태그를 지정할 수 있습니다.

- `CreateAccessPoint`
- `CreateFileSystem`

사용자가 생성 시 리소스에 태그를 지정할 수 있으려면 리소스를 생성하는 작업을 사용할 권한이 있어야 합니다(예: `elasticfilesystem:CreateAccessPoint` 또는 `elasticfilesystem:CreateFileSystem`). 리소스 생성 작업에 태그가 지정된 경우는 `elasticfilesystem:TagResource` 작업에 대한 추가 권한 부여를 AWS 수행하여 사용자에게 태그를 생성할 권한이 있는지 확인합니다. 따라서 사용자는 `elasticfilesystem:TagResource` 작업을 사용할 명시적 권한도 가지고 있어야 합니다.

`elasticfilesystem:TagResource` 작업에 대한 IAM 정책 정의에서 `Condition` 조건 키와 함께 `elasticfilesystem:CreateAction` 요소를 사용하여 리소스를 만드는 작업에 태그 지정 권한을 부여합니다.

Example정책: 파일 시스템 생성 시 태그 추가 허용

다음 예제의 정책은 사용자가 파일 시스템을 생성하고 생성 도중 태그를 적용하는 것을 허용합니다. 사용자는 기존 리소스에 태그를 지정할 수 없습니다(`elasticfilesystem:TagResource` 작업을 직접 호출할 수 없습니다).

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
         "elasticfilesystem:CreateFileSystem"
      ],
      "Resource": "arn:aws:elasticfilesystem:region:account-id:file-system/*"
    },
    {
      "Effect": "Allow",
      "Action": [
         "elasticfilesystem:TagResource"
      ],
      "Resource": "arn:aws:elasticfilesystem:region:account-id:file-system/*",
      "Condition": {
         "StringEquals": {
             "elasticfilesystem:CreateAction": "CreateFileSystem"
          }
       }
    }
  ]
}
```

태그를 사용하여 Amazon EFS 리소스에 대한 액세스 제어

Amazon EFS 리소스 및 작업에 대한 액세스를 제어하려면 태그를 기반으로 IAM 정책을 사용할 수 있습니다. 두 가지 방법으로 제어할 수 있습니다.

- Amazon EFS 리소스의 태그를 기반으로 리소스에 대한 액세스를 제어할 수 있습니다.
- IAM 요청 조건에 어떤 태그가 전달될 수 있는지를 제어할 수 있습니다.

태그를 사용하여 AWS 리소스에 대한 액세스를 제어하는 방법에 대한 자세한 내용은 IAM 사용 설명서의 태그를 사용하여 액세스 제어를 참조하세요.

리소스의 태그를 기반으로 액세스 제어

사용자나 역할이 Amazon EFS 리소스에서 어떤 작업을 수행할 수 있는지를 제어하기 위해 리소스의 태그를 사용할 수 있습니다. 예를 들어, 리소스에 있는 태그의 키-값 페어를 기반으로 파일 시스템 리소스에서 특정 API 작업을 허용하거나 거부할 수 있습니다.

Example정책: 특정 태그를 사용하는 경우에만 파일 시스템을 생성

다음 예제 정책을 통해 사용자는 특정 태그 키-값 페어(이 예제에서는 `key=Department`, `value=Finance`)로 태그를 지정하는 경우에만 파일 시스템을 생성할 수 있습니다.

```
{
    "Effect": "Allow",
    "Action": [
        "elasticfilesystem:CreateFileSystem",
        "elasticfilesystem:TagResource"
    ],
    "Resource": "arn:aws:elasticfilesystem:region:account-id:file-system/*",
    "Condition": {
        "StringEquals": {
            "aws:RequestTag/Department": "Finance"
        }
    }
}
```

Example정책: 특정 태그가 있는 파일 시스템 삭제

다음 예제 정책은 사용자가 `Department=Finance` 태그가 지정된 파일 시스템만 삭제하도록 허용합니다.

JSON

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

```
            ],
            "Resource": "arn:aws:elasticfilesystem:us-east-1:111122223333:file-
system/*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/Department": "Finance"
                }
            }
        }
    ]
}
```

Amazon EFS에 대해 서비스 연결 역할 사용

Amazon EFS는 AWS Identity and Access Management (IAM) 서비스 연결 역할을 사용합니다. Amazon EFS 서비스 연결 역할은 Amazon EFS에 직접 연결된 고유한 유형의 IAM 역할입니다. 사전 정의된 Amazon EFS 서비스 연결 역할에는 서비스가 AWS 서비스 사용자를 대신하여 다른를 호출하는 데 필요한 권한이 포함됩니다.

필요한 권한을 수동으로 추가할 필요가 없으므로 서비스 연결 역할은 Amazon EFS를 더 쉽게 설정할 수 있습니다. Amazon EFS에서 서비스 연결 역할의 권한을 정의하므로, Amazon EFS만 해당 역할을 수임할 수 있습니다. 정의된 권한에는 신뢰 정책과 권한 정책이 포함되며, 이 권한 정책은 다른 IAM 엔터티에 연결할 수 없습니다.

먼저 Amazon EFS 파일 시스템을 삭제한 다음에만 Amazon EFS 서비스 연결 역할을 삭제할 수 있습니다. 이렇게 하면 리소스에 대한 액세스 권한을 실수로 삭제할 수 없기 때문에 Amazon EFS 리소스가 보호됩니다.

서비스 연결 역할을 사용하면 모든 API 호출을 볼 수 있습니다 AWS CloudTrail. Amazon EFS가 사용자를 대신하여 수행하는 모든 작업을 추적할 수 있기 때문에 요구 사항 모니터링 및 감사에 유용합니다. 자세한 내용은 EFS 서비스 연결 역할에 대한 로그 항목 단원을 참조하십시오.

자세한 내용은 IAM 사용 설명서의 서비스 연결 역할 권한을 참조하세요.

Amazon EFS에 대한 서비스 연결 역할 권한

Amazon EFS는 AWSServiceRoleForAmazonElasticFileSystem이라는 서비스 연결 역할을 사용하여 Amazon EFS가 EFS 파일 시스템을 대신하여 AWS 리소스를 호출하고 관리할 수 있도록 합니다.

AWSServiceRoleForAmazonElasticFileSystem 서비스 연결 역할은 `elasticfilesystem.amazonaws.com`을 신뢰하여 역할을 맡습니다.

역할 권한 정책은 Amazon EFS가 정책 정의 JSON에 포함된 작업을 완료하도록 허용합니다.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "backup-storage:MountCapsule",
                "ec2:CreateNetworkInterface",
                "ec2:DeleteNetworkInterface",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeSubnets",
                "ec2:DescribeNetworkInterfaceAttribute",
                "ec2:ModifyNetworkInterfaceAttribute",
                "tag:GetResources"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:DescribeKey"
            ],
            "Resource": "arn:aws:kms:*:*:key/*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "backup:CreateBackupVault",
                "backup:PutBackupVaultAccessPolicy"
            ],
            "Resource": [
                "arn:aws:backup:*:*:backup-vault:aws/efs/automatic-backup-vault"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "backup:CreateBackupPlan",
                "backup:CreateBackupSelection"
```

```
            ],
            "Resource": [
                "arn:aws:backup:*:*:backup-plan:*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:CreateServiceLinkedRole"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "iam:AWSServiceName": [
                        "backup.amazonaws.com"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:PassRole"
            ],
            "Resource": [
                "arn:aws:iam::*:role/aws-service-role/backup.amazonaws.com/
AWSServiceRoleForBackup"
            ],
            "Condition": {
                "StringLike": {
                    "iam:PassedToService": "backup.amazonaws.com"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "elasticfilesystem:DescribeFileSystems",
                "elasticfilesystem:CreateReplicationConfiguration",
                "elasticfilesystem:DescribeReplicationConfigurations",
                "elasticfilesystem:DeleteReplicationConfiguration",
                "elasticfilesystem:ReplicationRead",
                "elasticfilesystem:ReplicationWrite"
            ],
```

```
            "Resource": "*"
        }
    ]
}
```

> **Note**
>
> 유휴 시 암호화된 새 EFS 파일 시스템을 생성할 AWS KMS 때에 대한 IAM 권한을 수동으로 구성해야 합니다. 자세한 내용은 저장 데이터 암호화를 참조하세요.

IAM 엔터티(사용자, 그룹, 역할 등)가 서비스 연결 역할을 생성하고 편집하거나 삭제할 수 있도록 권한을 구성할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 서비스 연결 역할 권한을 참조하세요.

Amazon EFS에 대한 서비스 연결 역할 생성

대부분의 경우 서비스 연결 역할을 수동으로 생성할 필요가 없습니다. AWS Management Console AWS CLI, 또는 AWS API에서 EFS 파일 시스템에 대한 탑재 대상 또는 복제 구성을 생성하면 Amazon EFS가 서비스 연결 역할을 생성합니다.

또한, 이 서비스 연결 역할을 수동으로 삭제한 후 다시 생성해야 하는 경우 동일한 절차를 사용하여 계정에 역할을 재구성할 수 있습니다. EFS 파일 시스템에 대한 탑재 대상 또는 복제 구성을 생성하는 경우, Amazon EFS가 서비스 연결 역할을 생성합니다.

그러나 Amazon EFS가 서비스 연결 역할을 생성하지 않거나 서비스 연결 역할을 지원하기 전에 Amazon EFS 사용을 시작한 경우 서비스 연결 역할을 수동으로 생성할 수 있습니다. 지침은 IAM 사용 설명서의 서비스 연결 역할 생성을 참조하세요.

Amazon EFS에 대한 서비스 연결 역할 편집

Amazon EFS는 사용자가 `AWSServiceRoleForAmazonElasticFileSystem` 서비스 연결 역할을 편집하도록 허용하지 않습니다. 서비스 연결 역할을 생성한 후에는 다양한 개체가 역할을 참조할 수 있기 때문에 역할 이름을 변경할 수 없습니다. 하지만 IAM을 사용하여 역할의 설명을 편집할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 서비스 연결 역할 업데이트를 참조하세요.

Amazon EFS에 대한 서비스 연결 역할 삭제

서비스 연결 역할이 필요한 기능 또는 서비스가 더 이상 필요 없는 경우에는 해당 역할을 삭제할 것을 권합니다. 이렇게 하면 적극적으로 모니터링하거나 유지 관리하지 않은 미사용 엔터티가 없습니다. 단,

서비스 연결 역할에 대한 리소스를 먼저 정리해야 수동으로 삭제할 수 있습니다. 자세한 내용은 리소스 정리 및 AWS 계정 보호 단원을 참조하십시오.

> **Note**
>
> 리소스를 삭제하려고 할 때 Amazon EFS 서비스가 역할을 사용 중이면 삭제에 실패할 수 있습니다. 이 문제가 발생하면 몇 분 기다렸다가 작업을 다시 시도하세요.

IAM 콘솔 AWS CLI, 또는 AWS API를 사용하여 AWSServiceRoleForAmazonElasticFileSystem 서비스 연결 역할을 삭제합니다. 자세한 내용은 IAM 사용 설명서의 서비스 연결 역할 삭제를 참조하십시오.

Amazon EFS 서비스 연결 역할이 지원되는 리전

Amazon EFS는 서비스를 사용할 수 AWS 리전 있는 모든에서 서비스 연결 역할 사용을 지원합니다. 자세한 내용은 AWS 일반 참조 사용 설명서의 AWS 서비스 엔드포인트를 참조하세요.

Amazon Elastic File System 자격 증명 및 액세스 문제 해결

다음 정보를 사용하여 Amazon EFS 및 IAM으로 작업할 때 발생할 수 있는 일반적인 문제를 진단하고 수정할 수 있습니다.

주제

- Amazon EFS에서 작업을 수행할 권한이 없음
- iam:PassRole을 수행하도록 인증되지 않음
- 내 외부의 사람이 내 Amazon EFS 리소스 AWS 계정 에 액세스하도록 허용하고 싶습니다.

Amazon EFS에서 작업을 수행할 권한이 없음

작업을 수행할 권한이 없다는 오류가 표시되면 작업을 수행할 수 있도록 정책을 업데이트해야 합니다.

다음의 예제 오류는 `mateojackson` IAM 사용자가 콘솔을 사용하여 가상 *`my-example-widget`* 리소스에 대한 세부 정보를 보려고 하지만 가상 `elasticfilesystem:`*`GetWidget`* 권한이 없을 때 발생합니다.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 elasticfilesystem:GetWidget on resource: my-example-widget
```

이 경우, `elasticfilesystem:GetWidget` 작업을 사용하여 `my-example-widget` 리소스에 액세스할 수 있도록 `mateojackson` 사용자 정책을 업데이트해야 합니다.

도움이 필요한 경우 AWS 관리자에게 문의하세요. 관리자는 로그인 자격 증명을 제공한 사람입니다.

iam:PassRole을 수행하도록 인증되지 않음

`iam:PassRole` 작업을 수행할 수 있는 권한이 없다는 오류가 수신되면 Amazon EFS에 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 예제 오류는 `marymajor`라는 IAM 사용자가 콘솔을 사용하여 Amazon EFS에서 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 EFS 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 EFS에서 이러한 기능을 지원하는지 여부를 알아보려면 Amazon Elastic File System과 IAM의 작동 방식 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.

IAM을 사용하여 파일 시스템에 대한 액세스 제어

IAM 자격 증명 정책과 리소스 정책을 모두 사용하여 클라우드 환경에 맞게 확장 가능하고 최적화된 방식으로 Amazon EFS 리소스에 대한 클라이언트 액세스를 제어할 수 있습니다. IAM을 사용하면 클라이언트가 읽기 전용, 쓰기 및 루트 액세스를 포함하여 파일 시스템에서 특정 작업을 수행하도록 허용할 수 있습니다. IAM ID 정책 또는 파일 시스템 리소스 정책의 작업에 대한 "허용" 권한은 해당 작업에 대한 액세스를 허용합니다. 자격 증명 및 리소스 정책 모두에서 권한을 부여할 필요는 없습니다.

NFS 클라이언트는 EFS 파일 시스템에 연결할 때 IAM 역할을 사용하여 자신을 식별할 수 있습니다. 클라이언트가 파일 시스템에 연결하면 Amazon EFS에서 파일 시스템의 IAM 리소스 정책(즉, 파일 시스템 정책)과 자격 증명 기반 IAM 정책을 평가하여 부여할 적절한 파일 시스템 액세스 권한을 결정합니다.

NFS 클라이언트에 대한 IAM 권한 부여를 사용하면 클라이언트 연결 및 IAM 권한 부여 결정이 AWS CloudTrail에 기록됩니다. CloudTrail에서 Amazon EFS API 직접 호출을 기록하는 방법에 대한 자세한 내용은 를 사용하여 Amazon EFS API 호출 로깅 AWS CloudTrail 섹션을 참조하세요.

> **Important**
>
> IAM 권한 부여를 사용하여 클라이언트 액세스를 제어하려면 EFS 탑재 도우미를 사용하여 EFS 파일 시스템을 탑재해야 합니다. 자세한 내용은 IAM 권한 부여를 통한 탑재 단원을 참조하십시오.

기본 EFS 파일 시스템 정책

기본 EFS 파일 시스템 정책은 IAM을 사용하여 인증하지 않으며 탑재 대상을 사용하여 파일 시스템에 연결할 수 있는 익명 클라이언트에 대한 모든 액세스 권한을 부여합니다. 기본 정책은 파일 시스템 생성 시를 포함하여 사용자가 구성한 파일 시스템 정책이 존재하지 않을 때마다 적용됩니다. 기본 파일 시스템 정책이 적용될 때마다 `DescribeFileSystemPolicy` API 작업은 `PolicyNotFound` 응답을 반환합니다.

클라이언트에 대한 EFS 작업

파일 시스템 정책을 사용하여 클라이언트의 파일 시스템 액세스에 대해 다음 작업을 지정할 수 있습니다.

작업	설명
`elasticfilesystem:ClientMount`	파일 시스템에 대한 읽기 전용 액세스를 제공합니다.
`elasticfilesystem:ClientWrite`	파일 시스템에 대한 쓰기 권한을 제공합니다.
`elasticfilesystem:ClientRootAccess`	파일 시스템에 액세스할 때 루트 사용자를 사용할 수 있는 기능을 제공합니다.

클라이언트에 대한 EFS 조건 키

조건을 표시하려면 미리 정의된 조건 키를 사용합니다. Amazon EFS에는 NFS 클라이언트를 위해 다음과 같은 사전 정의된 조건 키가 있습니다. IAM 제어를 사용하여 EFS 파일 시스템에 대한 액세스를 보호하는 경우 다른 조건 키는 적용되지 않습니다.

EFS 조건 키	설명	연산자
`aws:SecureTransport`	EFS 파일 시스템에 연결할 때 클라이언트가 TLS를 사용하도록 요구하려면 이 키를 사용합니다.	부울
`aws:SourceIp`	이 키를 사용하여 요청자의 IP 주소를 정책에서 지정한 IP 주소와 비교합니다. `aws:SourceIp` 조건 키는 퍼블릭 IP 주소 범위에만 사용할 수 있습니다.	문자열
`elasticfilesystem:AccessPointArn`	클라이언트가 연결 중인 EFS 액세스 포인트의 ARN입니다.	문자열

EFS 조건 키	설명	연산자
`elasticfilesystem:AccessedViaMountTarget`	이 키를 사용하면 파일 시스템 탑재 대상을 사용하지 않는 클라이언트가 EFS 파일 시스템에 액세스하는 것을 방지할 수 있습니다.	부울

파일 시스템 정책 예제

Amazon EFS 파일 시스템 정책의 예를 보려면 Amazon EFS의 리소스 기반 정책 예제 섹션을 참조하세요.

Amazon EFS에 대한 규정 준수 확인

AWS 서비스가 특정 규정 준수 프로그램의 범위에 포함되는지 알아보려면 규정 준수 프로그램 제공 범위 내 AWS 서비스를 참조하고 관심 있는 규정 준수 프로그램을 선택하십시오. 일반적인 정보는 AWS 규정 준수 프로그램을 참조하세요.

AWS Artifact(을)를 사용하여 타사 감사 보고서를 다운로드할 수 있습니다. 자세한 내용은 AWS Artifact에서 보고서 다운로드를 참조하세요.

AWS 서비스 사용 시 규정 준수 책임은 데이터의 민감도, 회사의 규정 준수 목표 및 관련 법률 및 규정에 따라 결정됩니다. AWS 서비스 사용 시 규정 준수 책임에 대한 자세한 내용은 AWS 보안 설명서를 참조하세요.

Amazon EFS의 복원성

AWS 글로벌 인프라는 AWS 및 가용 영역(AZ)을 중심으로 구축됩니다. AWS 리전에서는 물리적으로 분리되고 격리된 여러 개의 AZ를 제공하며, 이 AZ는 지연 시간이 짧고 처리량이 많으며 고도로 이중화된 네트워킹으로 연결됩니다. AZ를 사용하면 중단 없이 영역 간에 자동으로 장애 조치가 이루어지는 애플리케이션과 데이터베이스를 설계하고 운영할 수 있습니다. AZ는 기존의 단일 또는 다중 데이터 센터 인프라보다 가용성, 내결함성, 확장성이 뛰어납니다.

Amazon EFS 파일 시스템은 AWS 리전에 있는 한 개 이상의 가용 영역 장애에 대해 복원력이 뛰어납니다. 탑재 대상 자체는 가용성이 매우 뛰어나게 설계되어 있습니다. 고가용성 및 다른 AZ으로의 장애

조치를 설계하는 경우 각 AZ에 있는 탑재 대상의 IP 주소와 DNS는 정적이지만 여러 리소스에서 지원하는 중복 구성 요소라는 점에 유의하세요. 자세한 내용은 Amazon EFS가 Amazon EC2와 함께 작동하는 방식 섹션을 참조하세요.

AWS 리전 및 가용 영역에 대한 자세한 내용은 AWS 글로벌 인프라 섹션을 참조하세요.

NFS 클라이언트용 EFS 파일 시스템에 대한 네트워크 액세스 제어

네트워크 계층 보안 및 EFS 파일 시스템 정책을 사용하여 NFS 클라이언트의 Amazon EFS 파일 시스템에 대한 액세스를 제어할 수 있습니다. VPC 보안 그룹 규칙 및 네트워크 ACL 등 Amazon EC2에서 사용할 수 있는 네트워크 계층 보안 메커니즘을 사용할 수 있습니다. 또한 AWS IAM을 사용하여 EFS 파일 시스템 정책 및 자격 증명 기반 정책을 통해 NFS 액세스를 제어할 수 있습니다.

주제

- VPC 보안 그룹 사용
- Amazon EFS에서 인터페이스 VPC 엔드포인트 작업

VPC 보안 그룹 사용

Amazon EFS를 사용하는 경우, EC2 인스턴스의 VPC 보안 그룹 및 파일 시스템과 연결된 EFS 탑재 대상의 보안 그룹을 지정합니다. 보안 그룹은 방화벽 역할을 하고, 추가한 규칙은 트래픽 흐름을 정의합니다. 시작하기 연습에서 EFS 인스턴스를 시작할 때 보안 그룹 한 개를 생성했습니다. 그런 다음 다른 보안 그룹을 EFS 탑재 대상에 연결했습니다(기본 VPC에 대한 기본 보안 그룹). 이 방법은 시작하기 연습에는 적합합니다. 그러나 프로덕션 시스템의 경우, Amazon EFS에서 사용하려면 최소한의 권한을 가진 보안 그룹을 설정해야 합니다.

EFS 파일 시스템에 대한 인바운드 및 아웃바운드 액세스를 승인할 수 있습니다. 이렇게 하려면 EFS 인스턴스가 NFS(네트워크 파일 시스템) 포트를 사용하고 탑재 대상을 통해 EFS 파일 시스템을 연결할 수 있는 규칙을 추가합니다.

- 파일 시스템을 탑재한 각 EC2 인스턴스에는 NFS 포트 2049에서 탑재 대상에 대한 아웃바운드 액세스를 허용하는 보안 그룹이 있어야 합니다.
- EFS 탑재 대상에는 파일 시스템을 탑재하려는 각 EC2 인스턴스에서 NFS 포트 2049에 대한 인바운드 액세스를 허용하는 규칙이 있는 보안 그룹이 있어야 합니다.

다음 테이블에는 필요한 특정 보안 그룹 규칙이 나와 있습니다.

보안 그룹	규칙 유형	프로토콜	포트	소스/대상
EC2 인스턴스	아웃바운드	TCP	2049	탑재 대상 보안 그룹
탑재 대상	인바운드	TCP	2049	EC2 인스턴스 보안 그룹

Amazon EFS 작업용 소스 포트

Amazon EFS는 다양한 NFS 클라이언트를 지원하기 위해 모든 소스 포트로부터의 연결을 허용합니다. 권한이 있는 사용자만 Amazon EFS에 액세스할 수 있게 만들려면 다음 클라이언트 방화벽 규칙을 사용하는 것이 좋습니다. SSH를 사용하여 파일 시스템에 연결하고 다음 명령을 실행합니다.

```
iptables -I OUTPUT 1 -m owner --uid-owner 1-4294967294 -m tcp -p tcp --dport 2049 -j
 DROP
```

이 명령은 출력 체인(`-I OUTPUT 1`)의 시작 부분에 새로운 규칙을 삽입합니다. 권한이 없는 비 커널 프로세스(`-m owner --uid-owner 1-4294967294`)가 NFS 포트 2049(`-m tcp -p tcp –dport 2049`) 연결을 여는 것을 방지하는 규칙입니다.

네트워크 액세스에 대한 보안 고려 사항

NFS 버전 4.1(NFSv4.1) 클라이언트는 파일 시스템의 탑재 대상 중 하나의 NFS 포트(TCP 포트 2049)에 네트워크 연결을 구성할 수 있는 경우에만 파일 시스템을 탑재할 수 있습니다. 유사하게 NFSv4.1 클라이언트는 이러한 네트워크 연결을 구성할 수 있는 경우에만 파일 시스템에 액세스할 때 사용자 및 그룹 ID를 확인할 수 있습니다.

이러한 네트워크 연결을 수행할 수 있는지 여부는 다음 조합에 따라 관리됩니다.

- 탑재 대상의 VPC에서 제공하는 네트워크 격리 – 파일 시스템 탑재 대상에는 해당 탑재 대상과 연결된 퍼블릭 IP 주소가 있을 수 없습니다. 파일 시스템을 탑재할 수 있는 유일한 대상은 다음과 같습니다.
  - 로컬 Amazon VPC 및 Amazon EC2 인스턴스
  - 연결된 VPC의 EC2 인스턴스
  - AWS Direct Connect 및 AWS Virtual Private Network (VPN)를 사용하여 Amazon VPC에 연결된 온프레미스 서버

- 클라이언트 및 탑재 대상의 VPC 서브넷에 대한 네트워크 액세스 제어 목록(ACL) (탑재 대상 서브넷 외부에서의 액세스용) – 파일 시스템을 탑재하려면 클라이언트가 탑재 대상의 NFS 포트 2049에 TCP 연결을 수행하고 반송 트래픽을 수신할 수 있어야 합니다.
- 클라이언트 및 탑재 대상의 VPC 보안 그룹 규칙(모든 액세스에 해당) – 파일 시스템을 탑재하기 위한 EC2 인스턴스는 다음 보안 그룹 규칙이 유효해야 합니다.
  - 파일 시스템에는 인스턴스의 NFS 포트 2049에 대한 인바운드 연결을 활성화하는 규칙과 함께 네트워크 인터페이스에 보안 그룹이 있는 탑재 대상이 있어야 합니다. IP 주소(CIDR 범위)나 보안 그룹을 사용해 인바운드 연결을 활성화할 수 있습니다. 탑재 대상 네트워크 인터페이스에 대한 인바운드 NFS 포트 보안 그룹의 소스는 파일 시스템 액세스 제어의 핵심 요소입니다. NFS 포트 2049 이외의 인바운드 규칙과 모든 아웃바운드 규칙은 파일 시스템 탑재 대상 네트워크 인터페이스에 사용할 수 없습니다.
  - 탑재 인스턴스에는 파일 시스템의 탑재 대상 중 하나의 NFS 포트 2049에 대해 아웃바운드 연결을 활성화시키는 보안 그룹 규칙이 있는 네트워크 인터페이스가 있어야 합니다. IP 주소(CIDR 범위)나 보안 그룹을 사용해 아웃바운드 연결을 활성화할 수 있습니다.

자세한 내용은 탑재 대상 생성 단원을 참조하십시오.

보안 그룹 만들기

EC2 인스턴스 및 EFS 탑재 대상에 대한 보안 그룹 생성

다음은 Amazon EFS에 대한 보안 그룹을 생성할 때 수행할 일반적인 단계입니다. 보안 그룹 생성에 대한 지침은 Amazon VPC 사용 설명서의 보안 그룹 생성을 참조하세요.

1. EC2 인스턴스에 대해 다음 규칙이 포함된 보안 그룹을 생성합니다.
   - IP 주소 또는 네트워크에서 포트 22의 Secure Shell(SSH)를 사용하여 인바운드 액세스를 허용하는 인바운드 규칙입니다. 필요할 경우 보안을 위해 소스 주소를 제한할 수 있습니다.
   - 탑재 대상 보안 그룹에 대한 NFS 포트 2049의 아웃바운드 액세스를 허용하는 아웃바운드 규칙입니다. 탑재 대상 보안 그룹은 대상으로 식별됩니다.
2. EFS 탑재 대상에 대해 다음 규칙을 사용하여 보안 그룹을 생성합니다.
   - EC2 보안 그룹에서 NFS 포트 2049에 대한 액세스를 허용하는 인바운드 규칙입니다. EC2 보안 그룹은 소스로 식별됩니다.

> **Note**
>
> 기본 아웃바운드 규칙이 모든 아웃바운드 트래픽을 허용하므로 별도의 아웃바운드 규칙을 추가할 필요가 없습니다.

Amazon EFS에서 인터페이스 VPC 엔드포인트 작업

Virtual Private Cloud(VPC)와 Amazon EFS API 간에 프라이빗 연결을 설정하기 위해 인터페이스 VPC 엔드포인트를 생성할 수 있습니다. 엔드포인트는 인터넷 게이트웨이, NAT 인스턴스 또는 가상 프라이빗 네트워크(VPN) 연결 없이도 Amazon EFS API에 대한 안전한 연결을 제공합니다. 자세한 내용은 Amazon VPC 사용 설명서의 인터페이스 VPC 엔드포인트를 사용하여 AWS 서비스 액세스를 참조하세요.

인터페이스 VPC 엔드포인트는 프라이빗 IP 주소를 사용하여 AWS 서비스 간에 프라이빗 통신을 활성화하는 AWS PrivateLink기능으로 구동됩니다. AWS PrivateLink를 사용하려면 Amazon VPC 콘솔, API 또는 CLI를 사용하여 VPC에서 Amazon EFS에 대한 인터페이스 VPC 엔드포인트를 생성합니다. 이렇게 하면 Amazon EFS API 요청을 처리하는 프라이빗 IP 주소가 있는 탄력적 네트워크 인터페이스가 서브넷에서 생성됩니다. 또는 VPC 피어링을 사용하여 온프레미스 환경이나 다른 VPCs에서 Site-to-Site VPN Direct Connect VPC 엔드포인트에 액세스할 수도 있습니다. 자세한 내용은 Amazon VPC 사용 설명서의 AWS PrivateLink를 사용하여 서비스에 VPC 연결을 참조하세요.

Amazon EFS에 대한 인터페이스 엔드포인트 생성

Amazon EFS에 대한 인터페이스 VPC 엔드포인트를 생성하려면 다음 중 하나를 사용합니다.

- **com.amazonaws.*region*.elasticfilesystem** – Amazon EFS API 작업을 위한 엔드포인트를 생성합니다.
- **com.amazonaws.*region*.elasticfilesystem-fips** - Federal Information Processing Standard(FIPS) 140-2를 준수하는 Amazon EFS API의 엔드포인트를 생성합니다.

Amazon EFS 엔드포인트의 전체 목록은 Amazon Web Services 일반 참조의 Amazon Elastic File System 엔드포인트 및 할당량을 참조하세요.

인터페이스 엔드포인트를 생성하는 방법에 대한 자세한 내용은 Amazon VPC 사용 설명서의 인터페이스 VPC 엔드포인트를 사용하여 AWS 서비스 액세스를 참조하세요.

Amazon EFS에 대한 VPC 엔드포인트 정책 생성

Amazon EFS API에 대한 액세스를 제어하기 위해 VPC 엔드포인트에 AWS Identity and Access Management (IAM) 정책을 연결할 수 있습니다. 이 정책은 다음을 지정합니다.

- 작업을 수행할 수 있는 보안 주체.
- 수행할 수 있는 작업.
- 작업을 수행할 수 있는 리소스

자세한 내용은 Amazon VPC 사용자 가이드의 엔드포인트 정책을 사용하여 VPC 엔드포인트로 액세스 제어를 참조하세요.

다음 예에서는 엔드포인트를 통해 EFS 파일 시스템을 생성할 수 있는 모든 사람 권한을 거부하는 VPC 엔드포인트 정책을 보여 줍니다. 또한 이 정책 예에서는 모든 사용자에게 다른 모든 작업을 수행할 수 있는 권한을 부여합니다.

```
{
   "Statement": [
        {
            "Action": "*",
            "Effect": "Allow",
            "Resource": "*",
            "Principal": "*"
        },
        {
            "Action": "elasticfilesystem:CreateFileSystem",
            "Effect": "Deny",
            "Resource": "*",
            "Principal": "*"
        }
    ]
}
```

NFS(Network File System) 수준 사용자, 그룹 및 권한 사용

파일 시스템을 만들면 기본적으로 루트 사용자(UID 0)에게만 읽기, 쓰기 및 실행 권한이 있습니다. 다른 사용자가 파일 시스템을 수정할 수 있게 하려면 루트 사용자가 다른 사용자에게 액세스 권한을 명시적으로 부여해야 합니다. 액세스 포인트를 사용하여 루트가 아닌 사용자가 쓰기 가능한 디렉터리를 자동으로 생성할 수 있습니다. 자세한 내용은 액세스 포인트 작업 단원을 참조하십시오.

EFS 파일 시스템 객체에는 연결된 Unix 스타일 모드가 있습니다. 이 모드 값은 해당 객체에 대한 작업을 수행할 수 있는 권한을 정의합니다. Unix 스타일 시스템에 익숙한 사용자는 이러한 권한과 관련된 Amazon EFS 작동 방식을 쉽게 이해할 수 있습니다.

또한 Unix 스타일 시스템에서 사용자와 그룹은 Amazon EFS가 파일 소유권을 나타내는 데 사용하는 숫자 식별자에 매핑됩니다 Amazon EFS의 경우 파일 시스템 객체(파일, 디렉터리 등)는 단일 소유자 및 단일 그룹이 소유합니다. Amazon EFS는 사용자가 파일 시스템 객체에 액세스하려고 할 때 매핑된 숫자 ID를 사용하여 권한을 확인합니다.

> **Note**
>
> NFS 프로토콜은 사용자당 최대 16개의 그룹 ID(GID)를 지원하며 추가 GID는 NFS 클라이언트 요청에서 잘립니다. 자세한 내용은 NFS 파일 시스템에서 허용된 파일에 대한 액세스가 거부되었습니다. 단원을 참조하십시오.

다음에서는 권한의 예제와 Amazon EFS에 대한 NFS 권한 고려 사항 관련 토론을 확인할 수 있습니다.

주제

- 파일 및 디렉터리 권한
- 예제: EFS 파일 시스템 사용 사례 및 권한
- 파일 시스템 내의 파일 및 디렉터리에 대한 사용자 및 그룹 ID 권한
- 루트 스쿼싱 사용 안 함
- 권한 캐싱
- 파일 시스템 객체 소유권 변경
- EFS 액세스 포인트

파일 및 디렉터리 권한

EFS 파일 시스템의 파일 및 디렉터리는 EFS 액세스 포인트에서 재정의하지 않는 한, 탑재 NFSv4.1 클라이언트에서 어설션하는 사용자 및 그룹 ID를 기반으로 표준 Unix 스타일의 읽기, 쓰기, 실행 권한을 지원합니다. 자세한 내용은 NFS(Network File System) 수준 사용자, 그룹 및 권한 사용 단원을 참조하십시오.

> **Note**
>
> 기본적으로 이러한 액세스 제어 계층은 사용자 및 그룹 ID 어설션에서 NFSv4.1 클라이언트를 신뢰하는지에 따라 달라집니다. AWS Identity and Access Management (IAM) 리소스 기반 정책 및 자격 증명 정책을 사용하여 NFS 클라이언트에 권한을 부여하고 읽기 전용, 쓰기 및 루트 액세스 권한을 제공할 수 있습니다. EFS 액세스 포인트를 사용하여 NFS 클라이언트에서 제공하는 운영 체제 사용자 및 그룹 자격 증명 정보를 재정의할 수 있습니다. 자세한 내용은 IAM을 사용하여 파일 시스템에 대한 액세스 제어 및 액세스 포인트 생성 섹션을 참조하세요.

파일 및 디렉터리 읽기, 쓰기, 실행 권한에 대한 예를 들겠습니다. Alice는 파일 시스템의 개인 디렉터리인 `/alice`에서 원하는 모든 파일을 읽고 쓸 수 있는 권한을 갖고 있습니다. 그러나 동일한 파일 시스템에 위치한 Mark의 개인 디렉터리인 `/mark`의 파일을 읽거나 쓸 권한은 갖고 있지 않습니다. Alice와 Mark는 둘 다 공유 디렉터리 `/share`에서 파일을 읽을 수 있지만 쓸 수는 없습니다.

예제: EFS 파일 시스템 사용 사례 및 권한

VPC에서 EFS 파일 시스템과 이 파일 시스템의 탑재 대상을 만들면 Amazon EC2 인스턴스에서 로컬로 원격 파일 시스템을 탑재할 수 있습니다. `mount` 명령은 파일 시스템에서 디렉터리를 얼마든지 탑재할 수 있습니다. 그러나 파일 시스템을 처음으로 만든 경우 `/`에는 루트 디렉터리가 하나뿐입니다. 루트 사용자 및 루트 그룹이 탑재된 디렉터리를 소유합니다.

다음 `mount` 명령은 `/efs-mount-point` 로컬 디렉터리에서 파일 시스템 DNS 이름으로 식별되는 Amazon EFS 파일 시스템의 루트 디렉터리를 탑재합니다.

```
sudo mount -t nfs -o
 nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport file-
system-id.efs.aws-region.amazonaws.com:/ efs-mount-point
```

초기 권한 모드에서는 다음과 같이 허용됩니다.

- 소유자 루트에 대한 `read-write-execute` 권한
- 그룹 루트에 대한 `read-execute` 권한
- 다른 사용자에게 `read-execute` 권한이 허용됨

루트 사용자만 이 디렉터리를 수정할 수 있습니다. 또한 루트 사용자는 다른 사용자에게 이 디렉터리에 쓸 수 있는 권한을 부여할 수도 있습니다. 예:

- 쓰기 가능한 사용자별 하위 디렉터리를 만듭니다. 단계별 지침은 자습서: 쓰기 가능한 사용자별 하위 디렉터리 생성섹션을 참조하세요.
- 사용자가 EFS 파일 시스템 루트에 쓸 수 있도록 합니다. 루트 권한이 있는 사용자는 다른 사용자에게 파일 시스템에 대한 액세스 권한을 부여할 수 있습니다.
  - EFS 파일 시스템 소유권을 루트가 아닌 사용자 및 그룹에게로 변경하려면 다음 명령을 사용합니다.

    ```
    $ sudo chown user:group /EFSroot
    ```

  - 파일 시스템의 권한을 더욱 허용적으로 변경하려면 다음 명령을 사용합니다.

    ```
    $ sudo chmod 777 /EFSroot
    ```

    이 명령은 파일 시스템이 탑재된 모든 EC2 인스턴스에 대한 읽기-쓰기-실행 권한을 사용자 모두에게 부여합니다.

파일 시스템 내의 파일 및 디렉터리에 대한 사용자 및 그룹 ID 권한

EFS 파일 시스템의 파일 및 디렉터리는 사용자 ID 및 그룹 ID를 기반으로 표준 Unix 스타일의 읽기, 쓰기 및 실행 권한을 지원합니다. NFS 클라이언트가 액세스 포인트를 사용하지 않고 EFS 파일 시스템을 탑재하면 클라이언트에서 제공한 사용자 ID와 그룹 ID를 신뢰할 수 있게 됩니다. EFS 액세스 포인트를 사용하여 NFS 클라이언트에서 사용하는 사용자 ID 및 그룹 ID를 재정의할 수 있습니다. 사용자가 파일 및 디렉터리에 액세스하려고 하면 Amazon EFS는 사용자 ID와 그룹 ID를 검사하여 각 사용자에게 객체에 액세스할 권한이 있는지 확인합니다. 또한 Amazon EFS는 이러한 ID를 사용하여 사용자가 생성한 새 파일 및 디렉터리의 소유자 및 그룹 소유자를 나타냅니다. Amazon EFS는 사용자 또는 그룹 이름을 검사하지 않고 숫자 식별자만 사용합니다.

> **Note**
>
> EC2 인스턴스에서 사용자를 만드는 경우 만든 사용자에게 임의의 숫자 사용자 ID(UID) 및 그룹 ID(GID)를 할당할 수 있습니다. 숫자 사용자 ID는 Linux 시스템의 `/etc/passwd` 파일에서 설정합니다. 숫자 그룹 ID는 `/etc/group` 파일에 있습니다. 이러한 파일은 이름과 ID 간의 매핑을 정의합니다. EC2 인스턴스 외부에서 Amazon EFS는 루트 ID 0을 비롯하여 이러한 ID를 인증하지 않습니다.

사용자가 서로 다른 두 EC2 인스턴스에서 EFS 파일 시스템에 액세스하는 경우 이러한 인스턴스에서 사용자의 UID가 동일한지 혹은 다른지 여부에 따라 다음과 같이 다르게 동작할 수 있습니다.

- 두 EC2 인스턴스에서 사용자 ID가 동일한 경우, Amazon EFS는 사용하는 EC2 인스턴스와 상관없이 해당 ID가 동일한 사용자를 나타내는 것으로 간주합니다. 따라서 어느 쪽 EC2 인스턴스에서 파일 시스템에 액세스하든 사용자 환경이 동일합니다.
- 두 EC2 인스턴스에서 사용자 ID가 동일하지 않은 경우, Amazon EFS는 사용자를 서로 다른 사용자로 간주합니다. 서로 다른 두 EC2 인스턴스에서 EFS 파일 시스템에 액세스할 때는 사용자 환경이 동일하지 않습니다.
- 다른 EC2 인스턴스에 있는 서로 다른 두 사용자가 ID를 공유하는 경우, Amazon EFS는 이들을 동일한 사용자로 간주합니다.

EC2 인스턴스에서 일관된 사용자 ID 매핑 관리를 고려할 수 있습니다. 사용자는 `id` 명령을 사용하여 자신의 숫자 ID를 확인할 수 있습니다.

```
$ id

uid=502(joe) gid=502(joe) groups=502(joe)
```

ID 매퍼 끄기

운영 체제의 NFS 유틸리티에는 사용자 이름과 ID 간의 매핑을 관리하는 ID 매퍼라는 데몬이 포함되어 있습니다. Amazon Linux에서는 이 대몬(daemon)을 `rpc.idmapd`라고 하며, Ubuntu에서는 `idmapd`라고 합니다. ID 매퍼는 사용자 및 그룹 ID를 이름으로 전환하고 그 반대로도 전환합니다. 그러나 Amazon EFS에서는 숫자 ID만 처리합니다. EC2 인스턴스에서는 이 프로세스를 끄는 것이 좋습니다. Amazon Linux에서는 일반적으로 ID 매퍼가 비활성화되어 있으므로 이러한 경우에는 ID 매퍼를 활성화하지 마세요. ID 매퍼를 끄려면 다음과 같은 명령을 사용합니다.

```
$  service rpcidmapd status
$  sudo service rpcidmapd stop
```

루트 스쿼싱 사용 안 함

기본적으로 EFS 파일 시스템에서는 루트 스쿼시가 비활성화됩니다. Amazon EFS는 `no_root_squash`를 사용하여 Linux NFS 서버처럼 동작합니다. 사용자 또는 그룹 ID가 `0`이면 에서는 해당 사용자를 `root` 사용자로 취급하고 권한 검사를 우회하여 모든 파일 시스템 객체에 대한 액세스 및 수정을 허용합니다. AWS Identity and Access Management (AWS IAM) 자격 증명 또는 리소스 정

책이 `ClientRootAccess` 작업에 대한 액세스를 허용하지 않는 경우 클라이언트 연결에서 루트 스쿼싱을 활성화할 수 있습니다. 루트 스쿼싱이 활성화되어 있으면 루트 사용자가 NFS 서버에 대한 권한이 제한되는 사용자로 전환됩니다.

자세한 내용은 IAM을 사용하여 파일 시스템에 대한 액세스 제어 단원을 참조하십시오.

NFS 클라이언트에 대한 IAM 권한 부여를 사용하여 루트 스쿼싱 활성화

단일 관리 워크스테이션을 제외한 모든 AWS 보안 주체의 EFS 파일 시스템에 대한 루트 액세스를 방지하도록 Amazon EFS를 구성할 수 있습니다. 이를 위해 NFS(Network File System) 클라이언트에 대한 AWS Identity and Access Management (IAM) 권한 부여를 구성합니다.

이렇게 하려면 다음과 같이 두 가지 IAM 권한 정책을 구성해야 합니다.

- 파일 시스템에 대한 읽기 및 쓰기 액세스를 명시적으로 허용하고 루트 액세스를 암시적으로 거부하는 EFS 파일 시스템 정책을 생성합니다.
- EC2 인스턴스 프로파일을 사용하여 파일 시스템에 대한 루트 액세스가 필요한 Amazon EC2 관리 워크스테이션에 IAM 자격 증명을 할당합니다. Amazon EC2 인스턴스 프로파일에 대한 자세한 내용은 AWS Identity and Access Management 사용 설명서의 인스턴스 프로파일 사용을 참조하세요.
- `AmazonElasticFileSystemClientFullAccess` AWS 관리 워크스테이션의 IAM 역할에 관리형 정책을 할당합니다. Amazon EFS의 AWS 관리형 정책에 대한 자세한 내용은 섹션을 참조하세요Amazon EFS의 자격 증명 및 액세스 관리.

NFS 클라이언트에 대한 IAM 권한 부여를 사용하여 루트 스쿼싱을 활성화하려면 다음 절차를 따르세요.

파일 시스템에 대한 루트 액세스를 차단하려면

1. Amazon Elastic File System 콘솔(https://console.aws.amazon.com/efs/)을 엽니다.
2. 파일 시스템을 선택합니다.
3. 루트 스쿼싱을 활성화할 파일 시스템을 선택합니다.
4. 파일 시스템 세부 정보 페이지에서 파일 시스템 정책을 선택한 다음 편집을 선택합니다. 파일 시스템 정책 페이지가 나타납니다.
5. 정책 옵션에서 기본적으로 루트 액세스 차단*을 선택합니다. 정책 JSON 객체가 정책 편집기에 나타납니다.
6. 저장을 선택하여 파일 시스템 정책을 저장합니다.

익명이 아닌 클라이언트는 자격 증명 기반 정책을 통해 파일 시스템에 대한 루트 액세스를 얻을 수 있습니다. `AmazonElasticFileSystemClientFullAccess` 관리형 정책을 워크스테이션의 역할에 연결하면 IAM은 자격 증명 정책에 따라 워크스테이션에 루트 액세스 권한을 부여합니다.

관리 워크스테이션에서 루트 액세스를 활성화하려면

1. IAM 콘솔(https://console.aws.amazon.com/iam/)을 엽니다.
2. `EFS-client-root-access`라는 Amazon EC2의 역할을 생성합니다. IAM은 생성한 EC2 역할과 같은 이름의 인스턴스 프로파일을 생성합니다.
3. 생성한 EC2 역할에 AWS 관리`AmazonElasticFileSystemClientFullAccess`형 정책을 할당합니다. 이 정책의 내용은 다음과 같습니다.

   JSON

   ```
   {
       "Version":"2012-10-17",
       "Statement": [
           {
               "Effect": "Allow",
               "Action": [
                   "elasticfilesystem:ClientMount",
                   "elasticfilesystem:ClientRootAccess",
                   "elasticfilesystem:ClientWrite",
                   "elasticfilesystem:DescribeMountTargets"
               ],
               "Resource": "*"
           }
       ]
   }
   ```

4. 다음 설명에 따라 관리 워크스테이션으로 사용 중인 EC2 인스턴스에 인스턴스 프로파일을 연결합니다. 자세한 내용은 Linux 인스턴스용 Amazon EC2 사용 설명서의 인스턴스에 IAM 역할 연결을 참조하세요.

   a. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.

   b. 탐색 창에서 인스턴스를 선택합니다.

   c. 인스턴스를 선택합니다. 작업에서 인스턴스 설정, IAM 역할 연결/바꾸기을 차례로 선택합니다.

d. 첫 번째 단계에서 생성한 IAM 역할인 `EFS-client-root-access`를 선택하고 적용을 선택합니다.

5. 관리 워크스테이션에 EFS 탑재 도우미를 설치합니다. EFS 탑재 도우미 및 amazon-efs-utils 패키지에 대한 자세한 내용은 Amazon EFS 클라이언트 수동 설치 섹션을 참조하세요.

6. 다음 명령을 `iam` 탑재 옵션과 함께 사용하여 관리 워크스테이션에 EFS 파일 시스템을 탑재합니다.

```
$ sudo mount -t efs -o tls,iam file-system-id:/ efs-mount-point
```

IAM 권한 부여를 사용하여 파일 시스템을 자동으로 탑재하도록 Amazon EC2 인스턴스를 구성할 수 있습니다. IAM 권한 부여를 사용하여 EFS 파일 시스템을 탑재하는 방법에 대한 자세한 내용은 IAM 권한 부여를 통한 탑재 섹션을 참조하세요.

권한 캐싱

Amazon EFS에서는 짧은 기간 동안 파일 권한을 캐싱합니다. 따라서 최근에 액세스 권한이 취소된 사용자가 짧은 기간 동안 해당 객체에 계속해서 액세스하게 될 수 있습니다.

파일 시스템 객체 소유권 변경

Amazon EFS에서는 POSIX `chown_restricted` 속성을 강제합니다. 즉, 루트 사용자만 파일 시스템 객체의 소유자를 변경할 수 있습니다. 루트 사용자 또는 소유자는 파일 시스템 객체의 소유자 그룹을 변경할 수 있습니다. 그러나 사용자가 루트 사용자가 아닌 경우 그룹은 소유자가 멤버인 그룹으로만 변경할 수 있습니다.

EFS 액세스 포인트

액세스 포인트는 운영 체제 사용자, 그룹 및 파일 시스템 경로를 액세스 포인트를 사용하여 수행된 모든 파일 시스템 요청에 적용합니다. 액세스 포인트의 운영 체제 사용자 및 그룹은 NFS 클라이언트에서 제공하는 모든 자격 증명 정보를 재정의합니다. 파일 시스템 경로는 액세스 포인트의 루트 디렉터리로 클라이언트에 노출됩니다. 이렇게 하면 공유 파일 기반 데이터 세트에 액세스할 때 각 애플리케이션이 항상 올바른 운영 체제 자격 증명과 올바른 디렉터리를 사용할 수 있습니다. 액세스 포인트를 사용하는 애플리케이션은 자체 디렉터리 및 해당 하위 디렉터리의 데이터에만 액세스할 수 있습니다. 액세스 포인트에 대한 자세한 내용은 액세스 포인트 작업 섹션을 참조하십시오.

액세스 포인트 작업

EFS 액세스 포인트는 EFS 파일 시스템에 대한 애플리케이션별 진입점으로, 공유 데이터 세트에 대한 애플리케이션 액세스를 더 쉽게 관리할 수 있도록 합니다. 액세스 포인트는 액세스 포인트를 통해 이루어지는 모든 파일 시스템 요청에 대해 사용자의 POSIX 그룹을 포함한 사용자 자격 증명을 적용할 수 있습니다. 또한 클라이언트가 지정된 디렉터리 또는 하위 디렉터리의 데이터에만 액세스할 수 있도록 파일 시스템에 대해 다른 루트 디렉터리를 적용할 수 있습니다.

AWS Identity and Access Management (IAM) 정책을 사용하여 특정 애플리케이션이 특정 액세스 포인트를 사용하도록 적용할 수 있습니다. IAM 정책을 액세스 포인트와 결합하면 애플리케이션의 특정 데이터 세트에 안전하게 액세스할 수 있습니다.

AWS Management Console, AWS Command Line Interface (AWS CLI) 및 Amazon EFS API를 사용하여 기존 EFS 파일 시스템에 대한 액세스 포인트를 생성할 수 있습니다. 액세스 포인트를 만드는 단계별 절차는 액세스 포인트 생성을 참조하세요.

액세스 포인트는 탑재 대상과 함께 작동

액세스 포인트를 사용하기 전에 VPC에 탑재 대상을 하나 이상 생성해야 합니다. 탑재 대상은 EFS 파일 시스템에 대한 네트워크 연결을 제공하는 반면 액세스 포인트는 액세스 제어 및 애플리케이션별 진입점을 제공합니다.

액세스 포인트는 탑재 대상의 가용 영역 배치를 상속합니다.

- 보안 그룹은 액세스 포인트 수준이 아닌 탑재 대상 수준에서 적용됩니다.
- 액세스 포인트는 탑재 대상이 있는 모든 가용 영역에서 사용할 수 있습니다.
- IAM 조건 키 `elasticfilesystem:AccessedViaMountTarget`은 파일 시스템 액세스가 직접 탑재 및 액세스 포인트 탑재 모두에 적용되는 탑재 대상을 통해서만 이루어지도록 합니다.

액세스 포인트를 사용하여 파일 시스템을 탑재할 때 EFS 탑재 도우미를 사용합니다. 탑재 명령에는 다음 예제와 같이 파일 시스템 ID, 액세스 포인트 ID 및 `tls` 탑재 옵션을 포함합니다.

```
$ mount -t efs -o tls,iam,accesspoint=fsap-abcdef0123456789a fs-
abc0123def456789a: /localmountpoint
```

액세스 포인트를 사용하여 파일 시스템을 탑재하는 방법에 대한 자세한 내용은 EFS 액세스 포인트를 사용한 탑재 섹션을 참조하세요.

주제

- 액세스 포인트를 사용하여 사용자 자격 증명 적용
- 액세스 포인트를 사용하여 루트 디렉터리 적용
- IAM 정책에서 액세스 포인트 사용

액세스 포인트를 사용하여 사용자 자격 증명 적용

액세스 포인트를 사용하여 액세스 포인트를 통해 수행된 모든 파일 시스템 요청에 대해 사용자 및 그룹 정보를 적용할 수 있습니다. 이 기능을 활성화하려면 액세스 포인트를 생성할 때 적용할 운영 체제 자격 증명을 지정해야 합니다.

이 과정에서 다음을 제공합니다.

- 사용자 ID - 사용자의 숫자 POSIX 사용자 ID입니다.
- 그룹 ID - 사용자의 숫자 POSIX 그룹 ID입니다.
- 보조 그룹 ID - 보조 그룹 ID 목록(선택 사항)입니다.

사용자 적용이 활성화되면 Amazon EFS가 NFS 클라이언트의 사용자 및 그룹 ID를 모든 파일 시스템 작업에 대한 액세스 포인트에 구성된 자격 증명으로 바꿉니다. 또한 사용자 적용은 다음과 같은 영향을 미칩니다.

- 새 파일과 디렉터리의 소유자 및 그룹은 액세스 포인트의 사용자 ID 및 그룹 ID로 설정됩니다.
- EFS는 파일 시스템 권한을 평가할 때 액세스 포인트의 사용자 ID, 그룹 ID 및 보조 그룹 ID를 고려하고 NFS 클라이언트의 ID는 무시합니다.

> **Important**
>
> 사용자 자격 증명 적용에는 `ClientRootAccess` IAM 권한이 적용됩니다.
> 예를 들어 액세스 포인트 사용자 ID, 그룹 ID 또는 둘 다를 루트로 구성할 수 있습니다(즉, UID, GID 또는 둘 다를 0으로 설정). 이 경우 NFS 클라이언트에 `ClientRootAccess` IAM 권한을 부여해야 합니다.

액세스 포인트를 사용하여 루트 디렉터리 적용

액세스 포인트를 사용하여 파일 시스템의 루트 디렉터리를 재정의할 수 있습니다. 루트 디렉터리를 적용할 때 액세스 포인트를 사용하는 NFS 클라이언트는 파일 시스템의 루트 디렉터리 대신 액세스 포인트에 구성된 루트 디렉터리를 사용합니다.

액세스 포인트를 생성할 때 액세스 포인트 `Path` 속성을 설정하여 이 기능을 활성화합니다. `Path` 속성은 이 액세스 포인트를 통해 수행된 모든 파일 시스템 요청에 대한 파일 시스템의 루트 디렉터리 전체 경로입니다. 전체 경로의 길이는 100자를 초과할 수 없습니다. 최대 4개의 하위 디렉터리를 포함할 수 있습니다.

액세스 포인트에서 루트 디렉터리를 지정하면 이 디렉터리가 액세스 포인트가 탑재된 NFS 클라이언트에 대한 파일 시스템의 루트 디렉터리가 됩니다. 예를 들어 액세스 포인트의 루트 디렉터리가 `/data`인 경우 액세스 포인트를 사용하여 `fs-12345678:/`을 탑재하면 액세스 포인트를 사용하지 않고 `fs-12345678:/data`을 탑재하는 것과 같은 효과가 있습니다.

액세스 포인트에서 루트 디렉터리를 지정할 때 액세스 포인트의 사용자가 파일 시스템을 성공적으로 탑재할 수 있도록 디렉터리 권한이 구성되어 있는지 확인합니다. 특히 실행 비트가 액세스 포인트 사용자 또는 그룹이나 모든 사용자에 대해 설정되어 있는지 확인합니다. 예를 들어 디렉터리 권한 값이 755이면 디렉터리 사용자 소유자가 파일을 나열하고 파일을 생성하고 탑재할 수 있으며 다른 모든 사용자는 파일을 나열하고 탑재할 수 있습니다.

액세스 포인트에 대한 루트 디렉터리 생성

파일 시스템에 액세스 포인트에 대한 루트 디렉터리 경로가 없는 경우, Amazon EFS는 지정된 소유권 및 권한을 가진 액세스 포인트 루트 디렉터리를 자동으로 생성합니다. 생성 시 디렉터리 소유권 및 권한을 지정하지 않으면 Amazon EFS는 루트 디렉터리를 생성하지 않습니다. 따라서 Linux 호스트에서 파일 시스템을 탑재하지 않고도 특정 사용자 또는 애플리케이션에 대한 파일 시스템 액세스를 프로비저닝할 수 있습니다. 루트 디렉터리를 생성하려면, 액세스 포인트를 생성할 때 다음 속성을 사용하여 루트 디렉터리 소유권 및 권한을 구성해야 합니다.

- `OwnerUid` – 루트 디렉터리 소유자로 사용할 숫자 POSIX 사용자 ID입니다.
- `OwnerGiD` – 루트 디렉터리 소유자 그룹으로 사용할 숫자 POSIX 그룹 ID입니다.
- Permissions – 디렉터리의 Unix 모드입니다. 일반적인 구성은 755입니다. 탑재할 수 있도록 실행 비트가 액세스 포인트 사용자에 대해 설정되어 있는지 확인합니다. 이 구성은 디렉터리 소유자에게 디렉터리에서 새 파일을 입력하고 나열하고 쓸 수 있는 권한을 부여하며, 다른 모든 사용자에게 파일을 입력하고 나열할 수 있는 권한을 부여합니다. Unix 파일 및 디렉터리 모드 작업에 대한 자세한 내용은 NFS(Network File System) 수준 사용자, 그룹 및 권한 사용 섹션을 참조하세요.

Amazon EFS는 디렉터리에 대해 OwnUid, OwnGID 및 권한이 지정된 경우에만 액세스 포인트 루트 디렉터리를 생성합니다. 이 정보를 제공하지 않으면 Amazon EFS는 루트 디렉터리를 생성하지 않습니다. 루트 디렉터리가 없으면 액세스 포인트를 사용하는 탑재 시도가 실패합니다.

액세스 포인트가 있는 파일 시스템을 탑재할 때, 액세스 포인트가 생성되면서 루트 디렉터리의 OwnerUid와 Permissions가 지정된 경우 액세스 포인트의 루트 디렉터리가 아직 존재하지 않는다면 해당 디렉터리가 생성됩니다. 탑재 전에 이미 액세스 포인트에 루트 디렉터리가 존재하는 경우 액세스 포인트에서 기존 권한을 덮어쓰지 않습니다. 루트 디렉터리를 삭제하면 다음에 액세스 포인트를 사용하여 파일 시스템을 탑재할 때 EFS에서 루트 디렉터리를 다시 만듭니다.

> **Note**
>
> 액세스 포인트 루트 디렉터리의 소유권 및 권한을 지정하지 않으면 Amazon EFS는 루트 디렉터리를 생성하지 않습니다. 액세스 포인트를 탑재하려는 모든 시도가 실패합니다.

액세스 포인트 루트 디렉터리의 보안 모델

루트 디렉터리 재정의가 적용되면 Amazon EFS가 `no_subtree_check` 옵션이 활성화된 Linux NFS 서버처럼 동작합니다.

NFS 프로토콜에서 서버는 파일에 액세스할 때 클라이언트가 고유한 참조로 사용하는 파일 핸들을 생성합니다. EFS는 예측할 수 없고 EFS 파일 시스템과 관련된 파일 핸들을 안전하게 생성합니다. 루트 디렉터리 재정의가 적용되면 Amazon EFS는 지정된 루트 디렉터리 외부의 파일에 대한 파일 핸들을 공개하지 않습니다. 그러나 경우에 따라 사용자가 대역 외 메커니즘을 사용하여 액세스 포인트 외부의 파일에 대한 파일 핸들을 가져올 수 있습니다. 예를 들어 두 번째 액세스 포인트에 대한 액세스 권한이 있는 경우 이 작업을 수행할 수 있습니다. 이렇게 하면 파일에 대한 읽기 및 쓰기 작업을 수행할 수 있습니다.

파일 소유권 및 액세스 권한은 사용자의 액세스 포인트 루트 디렉터리 내/외부의 파일에 대한 액세스에 항상 적용됩니다.

IAM 정책에서 액세스 포인트 사용

IAM 정책을 사용하여 IAM 역할로 식별되는 특정 NFS 클라이언트가 특정 액세스 포인트에만 액세스 가능하도록 할 수 있습니다. 이렇게 하려면 `elasticfilesystem:AccessPointArn` IAM 조건 키를 사용합니다. `AccessPointArn`은 파일 시스템이 탑재된 액세스 포인트의 Amazon 리소스 이름(ARN)입니다.

다음 예에 나온 파일 시스템 정책은 IAM 역할 app1이 액세스 포인트 fsap-01234567을 사용하여 파일 시스템에 액세스하고 app2가 액세스 포인트 fsap-89abcdef를 사용하여 파일 시스템을 사용할 수 있도록 허용합니다.

JSON

```
{
    "Version":"2012-10-17",
    "Id": "MyFileSystemPolicy",
    "Statement": [
        {
            "Sid": "App1Access",
            "Effect": "Allow",
            "Principal": { "AWS": "arn:aws:iam::111122223333:role/app1" },
            "Action": [
                "elasticfilesystem:ClientMount",
                "elasticfilesystem:ClientWrite"
            ],
            "Resource": "arn:aws:elasticfilesystem:us-east-1:111122223333:file-
system/*",
            "Condition": {
                "StringEquals": {
                    "elasticfilesystem:AccessPointArn" :
 "arn:aws:elasticfilesystem:us-east-1:222233334444:access-point/fsap-01234567"
                }
            }
        },
        {
            "Sid": "App2Access",
            "Effect": "Allow",
            "Principal": { "AWS": "arn:aws:iam::111122223333:role/app2" },
            "Action": [
                "elasticfilesystem:ClientMount",
                "elasticfilesystem:ClientWrite"
            ],
            "Resource": "arn:aws:elasticfilesystem:us-east-1:111122223333:file-
system/*",
            "Condition": {
                "StringEquals": {
                    "elasticfilesystem:AccessPointArn" :
 "arn:aws:elasticfilesystem:us-east-1:222233334444:access-point/fsap-89abcdef"
                }
```

```
            }
        }
    ]
}
```

EFS 파일 시스템에 대한 퍼블릭 액세스 차단

Amazon EFS 퍼블릭 액세스 차단 기능은 EFS 파일 시스템에 대한 퍼블릭 액세스를 관리하는 데 도움이 되는 설정을 제공합니다. 기본적으로 새 EFS 파일 시스템에서는 퍼블릭 액세스가 허용되지 않습니다. 그러나 퍼블릭 액세스를 허용하도록 파일 시스템 정책을 수정할 수 있습니다.

Important

퍼블릭 액세스 차단을 활성화하면 파일 시스템에 직접 연결된 리소스 정책을 통해 퍼블릭 액세스가 허용되지 않으므로 리소스를 보호하는 데 도움이 됩니다. 퍼블릭 액세스 차단을 활성화하는 것 외에도 다음 정책을 주의 깊게 검토하여 퍼블릭 액세스를 허용하지 않는지 확인하세요.

- 연결된 AWS 보안 주체에 연결된 자격 증명 기반 정책(예: IAM 역할)
- 연결된 리소스에 연결된 AWS 리소스 기반 정책(예:AWS Key Management Service (KMS) 키)

주제

- 를 사용하여 퍼블릭 액세스 차단 AWS Transfer Family
- "퍼블릭"의 의미

를 사용하여 퍼블릭 액세스 차단 AWS Transfer Family

Amazon EFS를와 함께 사용하면 AWS Transfer Family파일 시스템이 퍼블릭 액세스를 허용하는 경우 파일 시스템과 다른 계정이 소유한 Transfer Family 서버에서 수신한 파일 시스템 액세스 요청이 차단됩니다. Amazon EFS는 파일 시스템의 IAM 정책을 평가하여 정책이 퍼블릭인 경우 요청을 차단합니다. 파일 시스템에 대한 AWS Transfer Family 액세스를 허용하려면 퍼블릭으로 간주되지 않도록 파일 시스템 정책을 업데이트합니다.

> **Note**
>
> Amazon EFS에서 Transfer Family를 사용하는 것은 2021년 AWS 계정 1월 6일 이전에 생성된 퍼블릭 액세스를 허용하는 정책이 있는 EFS 파일 시스템이 있는에 대해 기본적으로 비활성화됩니다. Transfer Family를 사용하여 파일 시스템에 액세스할 수 있도록 하려면 AWS Support에 문의하세요.

"퍼블릭"의 의미

파일 시스템이 퍼블릭 액세스를 허용하는지 여부를 평가할 때 Amazon EFS는 파일 시스템 정책을 퍼블릭으로 가정합니다. 그런 다음 정책을 평가하여 비공개로 판단할 수 있는지 결정합니다. 퍼블릭이 아닌 것으로 평가되려면 파일 시스템 정책은 다음 중 하나 이상의 고정 값(와일드카드가 없는 값)에만 액세스 권한을 부여해야 합니다.

- AWS 보안 주체, 사용자, 역할 또는 서비스 보안 주체(예: `aws:PrincipalOrgID`)
- `aws:SourceArn`
- `aws:SourceVpc`
- `aws:SourceVpce`
- `aws:SourceOwner`
- `aws:SourceAccount`
- `elasticfilesystem:AccessedViaMountTarget`
- `aws:userid, outside the pattern "AROLEID:*"`

이 규칙에서 다음 예제 정책은 퍼블릭으로 간주됩니다.

JSON

```
{
    "Version":"2012-10-17",
    "Id": "efs-policy-wizard-15ad9567-2546-4bbb-8168-5541b6fc0e55",
    "Statement": [
        {
            "Sid": "efs-statement-14a7191c-9401-40e7-a388-6af6cfb7dd9c",
            "Effect": "Allow",
            "Principal": {
```

```
                "AWS": "*"
            },
            "Action": [
                "elasticfilesystem:ClientMount",
                "elasticfilesystem:ClientWrite",
                "elasticfilesystem:ClientRootAccess"
            ],
            "Resource": "arn:aws:elasticfilesystem:us-east-1:111122223333:file-
system/*"
            }
    ]
}
```

true로 설정된 EFS 조건 키 elasticfilesystem:AccessedViaMountTarget을 사용하여 이 파일 시스템 정책을 비공개로 설정할 수 있습니다. elasticfilesystem:AccessedViaMountTarget을 사용하여 파일 시스템 탑재 대상을 사용하여 EFS 파일 시스템에 액세스하는 클라이언트에 지정된 EFS 작업을 허용할 수 있습니다. 다음 비공개 정책은 true로 설정된 elasticfilesystem:AccessedViaMountTarget 조건 키를 사용합니다.

JSON

```
{
    "Version":"2012-10-17",
    "Id": "efs-policy-wizard-15ad9567-2546-4bbb-8168-5541b6fc0e55",
    "Statement": [
        {
            "Sid": "efs-statement-14a7191c-9401-40e7-a388-6af6cfb7dd9c",
            "Effect": "Allow",
            "Principal": {
                "AWS": "*"
            },
            "Action": [
                "elasticfilesystem:ClientMount",
                "elasticfilesystem:ClientWrite",
                "elasticfilesystem:ClientRootAccess"
            ],
            "Resource": "arn:aws:elasticfilesystem:us-east-1:111122223333:file-
system/*",
            "Condition": {
                "Bool": {
                    "elasticfilesystem:AccessedViaMountTarget": "true"
```

```
                }
            }
        }
    ]
}
```

Amazon EFS 조건 키에 대한 자세한 내용은 클라이언트에 대한 EFS 조건 키 섹션을 참조하세요. 파일 시스템 정책 만들기에 대한 자세한 내용은 파일 시스템 정책 생성 섹션을 참조하세요.

Amazon EFS용 네트워크 격리

관리형 서비스인 Amazon Elastic File System은 AWS 글로벌 네트워크 보안으로 보호됩니다. AWS 보안 서비스와 AWS의 인프라 보호 방법에 대한 자세한 내용은 AWS Cloud Security을 참조하세요. 인프라 보안에 대한 모범 사례를 사용하여 AWS 환경을 설계하려면 보안 원칙 AWS Well-Architected Framework의 인프라 보호를 참조하세요.

AWS에서 게시한 API 직접 호출을 사용하여 네트워크를 통해 Amazon EFS에 액세스합니다. 고객은 다음을 지원해야 합니다.

- Transport Layer Security(TLS) TLS 1.2는 필수이며 TLS 1.3을 권장합니다.
- DHE(Ephemeral Diffie-Hellman) 또는 ECDHE(Elliptic Curve Ephemeral Diffie-Hellman)와 같은 완전 전송 보안(PFS)이 포함된 암호 제품군 Java 7 이상의 최신 시스템은 대부분 이러한 모드를 지원합니다.

이러한 API는 모든 네트워크 위치에서 호출할 수 있지만 Amazon EFS는 소스 IP 주소 기반의 제한 사항을 포함할 수 있는 리소스 기반 액세스 정책을 지원합니다. Amazon EFS 정책을 사용하여 특정 Amazon Virtual Private Cloud(VPC) 엔드포인트 또는 특정 VPC에서 액세스를 제어할 수도 있습니다. 그러면 AWS 네트워크의 특정 VPC에서만 특정 Amazon EFS 리소스에 대한 네트워크 액세스가 효과적으로 격리됩니다.

Amazon EFS 할당량

다음에서는 Amazon EFS 작업 시 할당량에 대해 알아봅니다.

주제

- 늘릴 수 있는 Amazon EFS 할당량
- 변경할 수 없는 Amazon EFS 리소스 할당량
- NFS 클라이언트에 대한 할당량
- Amazon EFS 파일 시스템 할당량
- 지원되지 않는 NFSv4.0 및 4.1 기능
- 추가 고려 사항
- 할당량과 관련된 파일 작업 오류 문제 해결

늘릴 수 있는 Amazon EFS 할당량

Service Quotas는 한 위치에서 할당량 또는 제한을 관리하는 데 도움이 되는 AWS 서비스입니다. Service Quotas 콘솔에서 Amazon EFS 제한 값을 보고의 EFS 파일 시스템 수에 대한 할당량 증가 AWS 리전 및 자주 액세스하는 데이터에 대한 읽기 IOPS를 요청할 수 있습니다.

AWS Support에 문의하여 다음 Amazon EFS 할당량 증가를 요청할 수도 있습니다. 자세한 내용은 할당량 증가 요청 섹션을 참조하세요. Amazon EFS 서비스 팀이 각 요청을 개별적으로 검토합니다.

- 각 고객 계정별 파일 시스템 수
- 각 파일 시스템에 대한 액세스 포인트 수
- 탄력적 처리량을 사용하는 파일 시스템당 최대 읽기 IOPS. 자주 액세스하는 파일 시스템의 읽기 IOPS가 증가하면 자주 액세스하지 않는 파일 시스템의 읽기 IOPS와 쓰기 IOPS도 모두 증가합니다.
- AWS 리전내 연결된 모든 클라이언트에 대한 리전별 파일 시스템당 탄력적 처리량
- AWS 리전내 연결된 모든 클라이언트에 대한 리전별 파일 시스템당 프로비저닝된 처리량

다음 테이블에는 변경을 요청할 수 있는 일반적인 기본 리소스 할당량이 나열되어 있습니다.

Resource	기본 할당량
의 각 고객 계정에 대한 파일 시스템 수 AWS 리전	1,000
각 파일 시스템에 대한 액세스 포인트 수	10,000
탄력적 처리량을 사용하는 파일 시스템당 최대 IOPS 자주 액세스하는 데이터에 대한 읽기 IOPS의 기본 할당량의 최대 10배 증가를 요청할 수 있습니다. 자주 액세스하는 데이터의 읽기 IOPS가 증가하면 자주 액세스하지 않는 데이터의 읽기 IOPS와 쓰기 IOPS도 증가합니다.	자주 액세스하지 않는 데이터 읽기: 90,000
	자주 액세스하는 데이터 읽기: 250,000
	쓰기: 50,000

다음 표에는 각의 연결된 모든 클라이언트에 대한 파일 시스템당 탄력적 처리량 할당량이 나열되어 있습니다 AWS 리전.

리전 파일 시스템 - 각의 연결된 모든 클라이언트에 대한 파일 시스템당 총 기본 탄력적 처리량 AWS 리전

AWS 리전	최대 읽기 처리량	최대 쓰기 처리량(측정된 처리량)
미국 동부(오하이오) 리전	60GiBps(초당 기가바이트)	5GiBps
미국 동부(버지니아 북부) 리전		
미국 서부(오리곤) 리전		
아시아 태평양(뭄바이) 리전		
아시아 태평양(서울) 리전		

AWS 리전	최대 읽기 처리량	최대 쓰기 처리량(측정된 처리량)
아시아 태평양(싱가포르) 리전 아시아 태평양(시드니) 리전 아시아 태평양(도쿄) 리전 유럽(프랑크푸르트) 리전 유럽(아일랜드) 리전 Europe (London) Region		
기타 모든 AWS 리전	20GiBps	1GiBps

다음 표에는 각의 연결된 모든 클라이언트에 대한 파일 시스템당 프로비저닝된 처리량 할당량이 나열되어 있습니다 AWS 리전.

리전 파일 시스템 - 각의 연결된 모든 클라이언트에 대한 파일 시스템당 총 기본 프로비저닝 처리량 AWS 리전

AWS 리전	최대 읽기 처리량	최대 쓰기 처리량(측정된 처리량)
미국 동부(오하이오) 리전 미국 동부(버지니아 북부) 리전 미국 서부(오리곤) 리전 유럽(아일랜드) 리전	10GiBps	3.33GiBps
기타 모든 AWS 리전	3GiBps	1GiBps

할당량 증가 요청

이러한 할당량 증가를 요청하려면 다음 단계를 AWS Support수행합니다. Amazon EFS 팀이 각 할당량 증가 요청을 검토합니다.

를 통해 할당량 증가를 요청하려면 AWS Support

1. AWS Support Center 페이지를 열고 필요한 경우 로그인합니다. 그런 다음 사례 생성을 선택합니다.
2. 사례 생성에서 서비스 제한 증가를 선택합니다.
3. 제한 유형에서 증가할 제한 유형을 선택합니다. 양식에서 필요한 필드를 입력한 다음, 선호하는 연락 방법을 선택합니다.

변경할 수 없는 Amazon EFS 리소스 할당량

다음을 포함하여 몇 가지 Amazon EFS 리소스의 할당량은 변경할 수 없습니다.

- 일반 리소스 할당량(예: 각 파일 시스템의 연결 수).
- AWS 리전에 연결된 모든 클라이언트에 대한 One Zone 파일 시스템별 탄력적 및 프로비저닝된 처리량 할당량
- AWS 리전에 연결된 모든 클라이언트에 대한 리전 또는 One Zone 파일 시스템별 버스트 처리량 할당량

다음 테이블에는 변경할 수 없는 일반 리소스 할당량이 나열되어 있습니다.

변경할 수 없는 일반 리소스 할당량

Resource	할당량
각 파일 시스템에 대한 연결 수	25,000
가용 영역당 파일 시스템별 탑재 대상 수	1
각 Virtual Private Cloud(VPC) 탑재 대상 수	1,400
탑재 대상당 보안 그룹의 수	5
파일 시스템별 태그 수	50

Resource	할당량
파일 시스템별 VPC 수	1

> **Note**
>
> 클라이언트는 파일 시스템과 다른 계정이나 VPC에 있는 탑재 대상에도 연결할 수 있습니다. 자세한 내용은 다른 AWS 계정 또는 VPC에서 EFS 파일 시스템 탑재 단원을 참조하십시오.

다음 표에는 각의 연결된 모든 클라이언트에 대한 파일 시스템당 총 기본 탄력적 및 프로비저닝된 처리량 제한이 나열되어 있습니다 AWS 리전.

One Zone 파일 시스템 - 각의 연결된 모든 클라이언트에 대한 파일 시스템당 총 기본 탄력적 처리량 및 프로비저닝 처리량 AWS 리전

AWS 리전	최대 읽기 처리량	최대 쓰기 처리량(측정된 처리량)
모두 AWS 리전	3GiBps	1GiBps

다음 표에는 각의 연결된 모든 클라이언트에 대한 파일 시스템당 총 버스팅 처리량 제한이 나열되어 있습니다 AWS 리전.

리전 및 One Zone 파일 시스템 - 각의 연결된 모든 클라이언트에 대한 파일 시스템당 총 버스팅 처리량 AWS 리전

AWS 리전	최대 읽기 처리량	최대 쓰기 처리량
미국 동부(오하이오) 리전	5GiBps	3GiBps
미국 동부(버지니아 북부) 리전		
미국 서부(오리곤) 리전		
아시아 태평양(시드니) 리전		
유럽(아일랜드) 리전		

AWS 리전	최대 읽기 처리량	최대 쓰기 처리량
기타 모든 AWS 리전	3GiBps	1GiBps

NFS 클라이언트에 대한 할당량

NFS 클라이언트에 대한 다음 할당량은 Linux NFSv4.1 클라이언트를 사용한다는 가정 하에 적용됩니다.

- 탄력적 처리량을 사용하고 버전 2.0 이상의 Amazon EFS 클라이언트(amazon-efs-utils 버전) 또는 Amazon EFS CSI 드라이버(aws-efs-csi-driver)를 사용하여 탑재된 파일 시스템의 경우 최대 읽기 및 쓰기 결합 처리량은 1,500MiBps(초당 메비바이트)입니다. 다른 모든 파일 시스템의 최대 처리량은 500MiBps입니다. 성능에 대한 자세한 내용은 성능 요약 섹션을 참조하세요. NFS 클라이언트 처리량은 최소 NFS 요청 크기가 4KB(읽기 요청에 1/3 측정 비율 적용 후)인 송수신된 총 바이트 수로 계산됩니다.
- 클라이언트당 최대 65,536명의 활성 사용자가 동시에 파일을 열 수 있습니다.
- 최대 65,536개의 파일이 인스턴스에서 동시에 열립니다. 디렉터리 콘텐츠 나열은 파일 열기로 계산되지 않습니다.
- 클라이언트의 각 고유 탑재는 연결당 최대 총 65,536개의 잠금을 획득할 수 있습니다.
- Amazon EFS에 연결할 때 온프레미스 또는 다른 AWS 리전 에 있는 NFS 클라이언트는 동일한 AWS 리전에서 EFS에 연결할 때보다 처리량이 낮아질 수 있습니다. 이 효과는 네트워크 대기 시간이 늘어났기 때문입니다. 최대 클라이언트별 처리량을 달성하려면 1ms 이하의 네트워크 대기 시간이 필요합니다. 온프레미스 NFS 서버에서 EFS로 대용량 데이터 세트를 마이그레이션하는 경우 DataSync 데이터 마이그레이션 서비스를 사용하세요.
- NFS 프로토콜은 사용자당 최대 16개의 그룹 ID(GID)를 지원하며 추가 GID는 NFS 클라이언트 요청에서 잘립니다. 자세한 내용은 NFS 파일 시스템에서 허용된 파일에 대한 액세스가 거부되었습니다. 단원을 참조하십시오.
- Microsoft Windows에서는 Amazon EFS를 사용할 수 없습니다.

Amazon EFS 파일 시스템 할당량

다음 할당량은 Amazon EFS 파일 시스템에만 해당됩니다.

Resource	할당량
파일 이름 길이(바이트)	255
최대 심볼 링크(symlink) 길이(바이트)	4,080
파일에 대한 최대 하드 링크 수	177
단일 파일 크기	52,673,613,135,872바이트(47.9TiB)
디렉터리 깊이 수준 수	1,000
모든 인스턴스 및 사용자에 대한 단일 파일의 잠금 수	512
각 파일 시스템 정책의 글자 수 제한	20,000건
*범용 모드의 초당 파일 작업 수	250,000

*범용 모드의 초당 파일 작업 수에 대한 자세한 내용은 성능 요약 섹션을 참조하십시오.

지원되지 않는 NFSv4.0 및 4.1 기능

Amazon EFS는 NFSv2 또는 NFSv3를 지원하지 않지만, 다음 기능을 제외하고 NFSv4.1과 NFSv4.0을 모두 지원합니다.

- pNFS
- 확장된 속성
- 모든 유형의 클라이언트 위임 또는 콜백
  - OPEN 작업은 항상 위임 유형으로 `OPEN_DELEGATE_NONE`을 반환합니다.
  - OPEN 작업은 `CLAIM_DELEGATE_CUR` 및 `CLAIM_DELEGATE_PREV` 클레임 유형에 대해 `NFSERR_NOTSUPP`을 반환합니다.
- 필수 잠금

  Amazon EFS의 모든 잠금은 권고 사항입니다. 즉, 쓰기 및 읽기 작업은 작업 실행 전 잠금이 충돌하는지 검사하지 않습니다.
- 공유 거부

NFS는 공유 거부 개념을 지원합니다. 공유 거부는 주로 Windows 클라이언트에서 사용자가 열려 있는 특정 파일에 대한 다른 사용자의 액세스를 거부하는 데 사용됩니다. Amazon EFS는 이를 지원하지 않으며, `OPEN4_SHARE_DENY_NONE` 이외의 공유 거부 값을 지정하는 모든 OPEN 명령에 대해 NFS 오류 `NFS4ERR_NOTSUPP`를 반환합니다. Linux NFS 클라이언트에서는 `OPEN4_SHARE_DENY_NONE` 이외의 다른 값은 사용하지 않습니다.

- 액세스 제어 목록(ACL)
- Amazon EFS는 파일 읽기 시 `time_access` 속성을 업데이트하지 않습니다. Amazon EFS는 다음과 같은 이벤트에서 `time_access`를 업데이트합니다.
  - 파일을 만든 경우(inode가 생성됨)
  - NFS 클라이언트가 명시적 `setattr` 직접 호출을 실행한 경우
  - 예를 들어 파일 크기 변경 또는 파일 메타데이터 변경으로 인해 inode에 쓰는 경우
  - inode 속성이 업데이트된 경우
- 네임스페이스
- 지속적인 응답 캐시
- Kerberos 기반 보안
- NFSv4.1 데이터 보존
- 디렉터리의 SetUID
- CREATE 작업 사용 시 지원되지 않는 파일 형식: 블록 디바이스(NF4BLK), 문자 디바이스(NF4CHR), 속성 디렉터리(NF4ATTRDIR) 및 명명된 속성(NF4NAMEDATTR)
- 지원되지 않는 속성: FATTR4_ARCHIVE, FATTR4_FILES_AVAIL, FATTR4_FILES_FREE, FATTR4_FILES_TOTAL, FATTR4_FS_LOCATIONS, FATTR4_MIMETYPE, FATTR4_QUOTA_AVAIL_HARD, FATTR4_QUOTA_AVAIL_SOFT, FATTR4_QUOTA_USED, FATTR4_TIME_BACKUP 및 FATTR4_ACL

  이러한 속성을 설정하려고 하면 `NFS4ERR_ATTRNOTSUPP` 오류가 발생하고 이 오류는 클라이언트에서 다시 전송됩니다.

추가 고려 사항

또한 다음 사항에 유의하세요.

- Amazon EFS 파일 시스템을 생성할 수 있는 AWS 리전 목록은 AWS 일반 참조 사용 설명서를 참조하세요.

- Amazon EFS는 `nconnect` 탑재 옵션을 지원하지 않습니다.
- Amazon EFS 파일 시스템은 Direct Connect 및 VPN을 사용하여 온프레미스 데이터 센터 서버에서 탑재할 수 있습니다. 자세한 내용은 자습서: 온프레미스 클라이언트로 탑재 단원을 참조하십시오.

할당량과 관련된 파일 작업 오류 문제 해결

EFS 파일 시스템에 액세스할 때, 파일 시스템의 파일에 특정 제한 사항이 적용됩니다. 이러한 제한을 초과하면 파일 작업 오류가 발생합니다. Amazon EFS의 파일 기반 제한에 대한 자세한 내용은 Amazon EFS 할당량 단원을 참조하세요.

아래에는 일반적인 몇 가지 파일 작업 오류와 각 오류와 관련된 제한이 나와 있습니다.

주제

- 열린 파일 및 파일 잠금 확인
- 명령에 실패하고 "디스크 할당량이 초과됨" 오류가 표시됨
- 명령에 실패하고 "I/O 오류"가 표시됨
- 명령에 실패하고 "파일 이름이 너무 김" 오류가 표시됨
- 명령이 실패하고 "File not found(파일을 찾을 수 없음)" 오류가 표시됨
- 명령에 실패하고 "링크가 너무 많음" 오류가 표시됨
- 명령에 실패하고 "파일이 너무 큼" 오류가 표시됨

열린 파일 및 파일 잠금 확인

파일 작업 오류를 해결하려면 EFS 탑재 지점에서 열린 파일과 파일 잠금을 검사하여 클라이언트 시스템이 한도에 도달했는지 확인할 수 있습니다.

- 열린 파일 확인 - 운영 체제의 도구를 사용하여 EFS 탑재 경로에 열린 파일을 나열합니다. 이렇게 하면 열린 파일 한도에 근접하고 있는지 식별하는 데 도움이 됩니다. 예를 들어, Linux에서 다음을 수행할 수 있습니다.

```
lsof <efs-mount-path>
```

- 파일 잠금 확인 - 시스템의 잠금 모니터링 도구를 사용하여 EFS 탑재 경로에 잠금이 있는 파일을 나열합니다. 이렇게 하면 잠금 한도에 도달했는지 식별하는 데 도움이 됩니다. 예를 들어, Linux에서 다음을 수행할 수 있습니다.

```
lslocks | grep <efs-mount-path>
```

이러한 명령은 EFS 한도에 대한 현재 사용량을 표시하므로 파일 작업 오류가 시스템 또는 서비스 한도 도달과 관련이 있는지 확인할 수 있습니다.

명령에 실패하고 "디스크 할당량이 초과됨" 오류가 표시됨

Amazon EFS에서는 현재 사용자 디스크 할당량을 지원하지 않습니다. 이 오류는 다음 제한 중 하나를 초과한 경우 발생할 수 있습니다.

- 최대 65,536명의 활성 사용자가 동시에 파일을 열 수 있습니다. 여러 번 로그인한 사용자 계정은 하나의 활성 사용자로 계산됩니다.
- 인스턴스당 최대 65,536개의 파일을 한 번에 열 수 있습니다. 디렉터리 콘텐츠 나열은 파일 열기로 계산되지 않습니다.
- 클라이언트의 각 고유 탑재는 연결당 최대 총 65,536개의 잠금을 획득할 수 있습니다.

취할 조치

이 문제가 발생하면 위 제한 사항 중 초과된 부분을 찾은 다음 그에 맞게 변경하여 문제를 해결할 수 있습니다. 자세한 내용은 NFS 클라이언트에 대한 할당량 단원을 참조하십시오. 현재 사용량을 확인하려면 열린 파일 및 파일 잠금 확인 섹션을 참조하세요.

명령에 실패하고 "I/O 오류"가 표시됨

이 오류는 다음 문제 중 하나가 있을 경우 발생합니다.

- 각 인스턴스당 65,536개 이상의 활성 사용자 계정에서 동시에 파일이 열려 있습니다.

  취할 조치

  이 문제가 발생하면 인스턴스의 열린 파일에 대한 제한을 충족시켜 해결할 수 있습니다. 제한을 충족하려면 인스턴스에서 동시에 Amazon EFS 파일 시스템의 파일을 열고 있는 활성 사용자의 수를 줄입니다. 현재 사용량을 확인하려면 열린 파일 및 파일 잠금 확인 섹션을 참조하세요.
- 파일 시스템을 암호화하는 AWS KMS 키가 삭제되었습니다.

  취할 조치

이 문제가 발생할 경우 CMK가 삭제된 후에는 더 이상 키 하에서 암호화된 데이터를 해독할 수 없습니다. 즉 데이터를 복구할 수 없게 됩니다.

명령에 실패하고 "파일 이름이 너무 깊" 오류가 표시됨

이 오류는 파일 이름 또는 심볼 링크(symlink)의 크기가 너무 긴 경우 발생합니다. 파일 이름에 대한 제한은 다음과 같습니다.

- 이름의 길이는 최대 255바이트입니다.
- symlink의 최대 길이는 4080바이트입니다.

취할 조치

이 문제가 발생하면 지원되는 제한을 충족하도록 파일 이름 또는 symlink 길이를 줄여 해결할 수 있습니다.

명령이 실패하고 "File not found(파일을 찾을 수 없음)" 오류가 표시됨

이 오류는 일부 이전 32비트 버전의 Oracle E-Business Suite가 32비트 파일 I/O 인터페이스를 사용하고 EFS가 64비트 inode 번호를 사용하기 때문입니다. 실패할 수 있는 시스템 호출에는 `stat()` 및 `readdir()`이 있습니다.

취할 조치

이 오류가 발생하면 nfs.enable_ino64=0 kernel 부팅 옵션을 사용하여 해결할 수 있습니다. 이 옵션은 64비트 EFS inode 번호를 32비트로 압축합니다. 커널 부팅 옵션은 Linux 배포판마다 다르게 처리됩니다. Amazon Linux에서 `/etc/default/grub`의 `GRUB_CMDLINE_LINUX_DEFAULT` 변수에 `nfs.enable_ino64=0 kernel`을 추가하여 이 옵션을 설정합니다. 커널 부팅 옵션을 설정하는 방법에 대한 자세한 내용은 배포를 참조하세요.

명령에 실패하고 "링크가 너무 많음" 오류가 표시됨

이 오류는 파일에 대한 하드 링크가 너무 많은 경우 발생합니다. 파일 하나의 하드 링크는 최대 177개입니다.

취할 조치

이 문제가 발생하면 지원되는 제한을 충족하도록 파일에 대한 하드 링크 수를 줄여 해결할 수 있습니다.

명령에 실패하고 "파일이 너무 큼" 오류가 표시됨

이 오류는 파일이 너무 큰 경우 발생합니다. 단일 파일의 크기는 최대 52,673,613,135,872바이트(47.9TiB)일 수 있습니다.

취할 조치

이 문제가 발생하면 지원되는 제한을 충족하도록 파일 크기를 줄여 해결할 수 있습니다.

Amazon EFS API

아마존 EFS API는 HTTP (RFC 2616)를 기반으로 하는 네트워크 프로토콜입니다. 각 API 직접 호출에 대해 파일 시스템을 관리하려는 AWS 리전의 리전별 Amazon EFS API 엔드포인트에 HTTP 요청을 보냅니다. API에서는 HTTP 요청/응답 본문에 대해 JSON(RFC 4627) 문서를 사용합니다.

아마존 EFS API는 RPC 모델입니다. 이 모델은 고정된 작업 세트가 있고, 이전 상호 작용이 없어도 클라이언트가 각 작업의 구문을 알고 있습니다. 다음 섹션에서 추상 RPC 표기법을 사용한 각 API 작업에 대한 설명을 볼 수 있습니다. 각 작업에는 네트워크 상에 나타나지 않는 작업 이름이 있습니다. 각 작업에 대해 주제는 HTTP 요청 요소에 대한 매핑을 지정합니다.

지정된 요청이 매핑되는 특정 Amazon EFS 작업은 요청 메서드(GET, PUT, POST 또는 DELETE)와 요청-URI가 일치되는 다양한 패턴의 조합에 따라 결정됩니다. PUT 또는 POST 작업인 경우 Amazon EFS에서는 요청-URI 경로 세그먼트, 쿼리 파라미터, 요청 본문의 JSON 객체에서 직접 호출 인수를 추출합니다.

> **Note**
>
> 작업 이름(예: `CreateFileSystem`)은 유선에 표시되지 않지만 AWS Identity and Access Management(IAM) 정책에서 의미가 있습니다. 자세한 내용은 Amazon EFS의 자격 증명 및 액세스 관리를 참조하세요.
>
> 작업 이름은 명령줄 도구 및 AWS SDK API 요소에서 명령의 이름을 지정하는 데에도 사용됩니다. 예를 들어, `CreateFileSystem` 작업에 매핑되는 `create-file-system`이라는 이름의 AWS CLI 명령이 있습니다.
>
> 이 작업 이름은 Amazon EFS API 직접 호출에 대한 AWS CloudTrail 로그에도 표시됩니다.

API 엔드포인트

엔드포인트는 AWS 웹 서비스의 진입점 역할을 하는 URL입니다. Amazon EFS는 다음과 같은 엔드포인트 유형을 지원합니다.

- IPv4 엔드포인트
- 이중 스택(IPv4 및 IPv6 둘 다) 엔드포인트
- FIPS 엔드포인트

요청 시에, 사용할 엔드포인트를 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. 사용 가능한 엔드포인트 표는 Amazon EFS 엔드포인트 및 할당량을 참조하세요.

IPv4 엔드포인트

IPv4 엔드포인트는 IPv4 트래픽만 지원합니다. IPv4 엔드포인트는 모든 리전에 사용할 수 있습니다. IPv4 엔드포인트의 형식은 리전에 따라 다릅니다.

- AWS 상용 리전의 IPv4 엔드포인트의 경우 형식은 `elasticfilesystem.region.amazonaws.com`입니다.

  예를 들어 `elasticfilesystem.us-west-2.amazonaws.com`을 엔드포인트로 지정하면 요청이 미국 서부(오리건) 리전(us-west-2) 엔드포인트로 전달됩니다.

- 중국 리전의 IPv4 엔드포인트의 경우 형식은 `elasticfilesystem.region.amazonaws.com.cn`입니다.

  예를 들어 중국(베이징) 리전의 Amazon EFS API 엔드포인트는 `elasticfilesystem.cn-north-1.amazonaws.com.cn`입니다.

이중 스택(IPv4 및 IPv6) 엔드포인트

이중 스택 엔드포인트는 IPv4 트래픽과 IPv6 트래픽을 모두 지원합니다. 이중 스택 엔드포인트에 요청하는 경우, 엔드포인트 URL이 네트워크 및 클라이언트에서 사용하는 프로토콜에 따라 IPv6 또는 IPv4 주소로 확인됩니다. 듀얼 스택 엔드포인트의 형식은 리전에 따라 다릅니다.

- AWS 상용 리전의 듀얼 스택 엔드포인트의 경우 형식은 `efs.aws-region.api.aws`입니다.

  예를 들어 미국 서부(오리건) 리전(us-west-2)의 이중 스택 엔드포인트는 `efs.eu-west-2.api.aws`입니다.

- 중국 리전의 듀얼 스택 엔드포인트의 경우 형식은 `efs.region.api.amazonwebservices.com.cn`입니다.

  예를 들어 중국(베이징) 리전의 듀얼 스택 엔드포인트는 `efs.cn-north-1.api.amazonwebservices.com.cn`입니다.

엔드포인트 지정

다음 예제에서는 Amazon EFS용 AWS CLI를 사용하여 미국 서부(캘리포니아 북부) 리전(us-west-1)의 엔드포인트를 지정하는 방법을 보여줍니다.

IPv4

```
aws efs get-rest-apis --region us-west-1 --endpoint-url https://elasticfilesystem.us-west-1.amazonaws.com
```

이중 스택

```
aws efs get-rest-apis --region us-west-1 --endpoint-url https://efs.us-west-1.api.aws
```

API 버전

호출에 사용 중인 API 버전은 요청 URI의 첫 번째 경로 세그먼트로 식별할 수 있으며 형식은 ISO 8601 날짜입니다. 예제는 CreateFileSystem을 참조하세요.

이 문서에서는 API 버전 2015-02-01에 대해 설명합니다.

관련 주제

다음 섹션에서는 API 작업에 대한 설명, 요청 인증을 위한 서명을 생성하는 방법, IAM 정책을 사용하여 이러한 API 작업에 대한 권한을 부여하는 방법을 제공합니다.

- Amazon EFS의 자격 증명 및 액세스 관리
- 작업
- 데이터 타입

Amazon EFS의 쿼리 API 요청 속도 활용하기

Amazon EFS API 요청은 서비스 성능을 높이기 위해 AWS 계정마다 리전별로 제한됩니다. 애플리케이션에서 시작하든, AWS CLI 또는 Amazon EFS 콘솔에서 시작되었든 관계없이 모든 Amazon EFS API 직접 호출은 최대 허용 API 요청 속도를 초과해서는 안 됩니다. 최대 API 요청 속도는 AWS 리전별로 다를 수 있습니다. API 요청은 기본 AWS 계정 요청에 따라 생성됩니다.

API 요청이 해당 카테고리의 API 요청 비율을 초과하는 경우 요청은 `ThrottlingException` 오류 코드를 반환합니다. 이 오류를 방지하려면 애플리케이션이 API 요청을 빠른 속도로 재시도하지 않도록 해야 합니다. 폴링 시 주의를 기울이고 지수 백오프 재시도를 사용하면 이 작업을 수행할 수 있습니다.

폴링

애플리케이션이 API 작업을 반복해서 직접 호출하여 업데이트 상태를 확인해야 할 수 있습니다. 폴링을 시작하기 전에 요청이 완료될 수 있도록 시간을 주세요. 폴링을 시작할 때는 연속된 요청 사이에 적절한 휴면 간격을 두세요. 최상의 결과를 얻으려면 휴면 간격을 늘리세요.

재시도 또는 일괄 처리

애플리케이션이 API 요청을 실패하면 다시 시도하거나 여러 리소스(예: 모든 Amazon EFS 파일 시스템)를 처리해야 할 수 있습니다. API 요청 속도를 낮추려면 연속된 요청 사이에 적절한 휴면 간격을 사용하세요. 최상의 결과를 얻으려면 휴면 간격을 늘리거나 가변적으로 사용하세요.

휴면 간격 계산

API 요청을 폴링하거나 재시도해야 하는 경우 지수 백오프 알고리즘을 사용하여 API 직접 호출 간 휴면 간격을 계산하는 것이 좋습니다. 지수 백오프의 기본 개념은 오류 응답이 연이어 나올 때마다 재시도 간 대기 시간을 점진적으로 늘리는 것입니다. 이 알고리즘에 대한 자세한 내용 및 구현 예제는 Amazon Web Services 일반 참조의 재시도 동작을 참조하세요.

작업

다음 작업이 지원됩니다.

- CreateAccessPoint
- CreateFileSystem
- CreateMountTarget
- CreateReplicationConfiguration
- CreateTags
- DeleteAccessPoint
- DeleteFileSystem
- DeleteFileSystemPolicy
- DeleteMountTarget

- DeleteReplicationConfiguration
- DeleteTags
- DescribeAccessPoints
- DescribeAccountPreferences
- DescribeBackupPolicy
- DescribeFileSystemPolicy
- DescribeFileSystems
- DescribeLifecycleConfiguration
- DescribeMountTargets
- DescribeMountTargetSecurityGroups
- DescribeReplicationConfigurations
- DescribeTags
- ListTagsForResource
- ModifyMountTargetSecurityGroups
- PutAccountPreferences
- PutBackupPolicy
- PutFileSystemPolicy
- PutLifecycleConfiguration
- TagResource
- UntagResource
- UpdateFileSystem
- UpdateFileSystemProtection

CreateAccessPoint

EFS 액세스 포인트를 생성합니다. 액세스 포인트는 운영 체제 사용자 및 그룹 및 파일 시스템 경로를 액세스 포인트를 통해 이루어지는 모든 파일 시스템 요청에 적용하는 EFS 파일 시스템에 대한 애플리케이션별 보기입니다. 운영 체제 사용자 및 그룹은 NFS 클라이언트에서 제공하는 모든 자격 증명 정보를 재정의합니다. 파일 시스템 경로는 액세스 포인트의 루트 디렉터리로 노출됩니다. 액세스 포인트를 사용하는 애플리케이션은 자체 디렉터리 및 해당 하위 디렉터리의 데이터에만 액세스할 수 있습니다. 파일 시스템은 증가를 요청하지 않는 한 최대 10,000개의 액세스 포인트를 가질 수 있습니다. 자세한 내용을 알아보려면 EFS 액세스 포인트를 사용하여 파일 시스템 탑재를 참조하세요.

> **Note**
>
> 동일한 파일 시스템에 액세스 포인트를 생성하라는 여러 요청을 연속으로 빠르게 전송하고 파일 시스템이 액세스 포인트 한도에 가까워지면 이러한 요청에 대한 응답이 제한될 수 있습니다. 이는 파일 시스템이 명시된 액세스 포인트 제한을 초과하지 않도록 하기 위한 것입니다.

이 작업에는 `elasticfilesystem:CreateAccessPoint` 액션에 대한 권한이 필요합니다.

액세스 포인트를 만들 때 태그를 지정할 수 있습니다. 생성 작업에서 태그가 지정되면 IAM은 `elasticfilesystem:TagResource` 작업에서 추가 권한 부여를 수행해 사용자에게 태그를 생성할 권한이 있는지 확인합니다. 따라서 사용자는 `elasticfilesystem:TagResource` 작업을 사용할 명시적 권한도 가지고 있어야 합니다. 자세한 내용은 생성 시 리소스 태그 지정에 대한 권한 부여를 참조하세요.

Request Syntax

```
POST /2015-02-01/access-points HTTP/1.1
Content-type: application/json

{
   "ClientToken": "string",
   "FileSystemId": "string",
   "PosixUser": {
      "Gid": number,
      "SecondaryGids": [ number ],
      "Uid": number
   },
   "RootDirectory": {
      "CreationInfo": {
```

```
            "OwnerGid": number,
            "OwnerUid": number,
            "Permissions": "string"
         },
         "Path": "string"
      },
      "Tags": [
         {
            "Key": "string",
            "Value": "string"
         }
      ]
}
```

URI 요청 파라미터

요청은 URI 파라미터를 사용하지 않습니다.

요청 본문

요청은 JSON 형식으로 다음 데이터를 받습니다.

ClientToken

Amazon EFS가 멱등성 생성을 보장하기 위해 사용하는 최대 64개의 ASCII 문자로 구성된 문자열입니다.

유형: 문자열

길이 제한: 최소 길이는 1. 최대 길이는 64.

패턴: `.+`

필수 여부: 예

FileSystemId

액세스 포인트가 액세스를 제공하는 EFS 파일 시스템의 ID입니다.

유형: 문자열

길이 제약: 최대 길이는 128입니다.

패턴: `^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$`

필수 여부: 예

PosixUser

액세스 포인트를 사용하여 이루어진 모든 파일 시스템 요청에 적용되는 운영 체제 사용자 및 그룹입니다.

유형: PosixUser객체

필수 여부: 아니요

RootDirectory

액세스 포인트가 액세스 포인트를 사용하여 NFS 클라이언트에 사용자의 파일 시스템의 루트 디렉터리로 노출하는 EFS 파일 시스템의 디렉터리를 지정합니다. 액세스 포인트를 사용하는 클라이언트는 루트 디렉터리 및 해당 하위 디렉터리에 대한 액세스만 수행할 수 있습니다. 지정된 `RootDirectory > Path`가 없으면 클라이언트가 액세스 포인트에 연결할 때 Amazon EFS가 이를 생성하고 `CreationInfo` 설정을 적용합니다. `RootDirectory`를 지정할 때는 `Path` 및 `CreationInfo`를 제공해야 합니다.

Amazon EFS는 사용자가 CreationInfo(OwNuID, ownGID 및 디렉터리에 대한 권한)를 제공한 경우에만 루트 디렉터리를 생성합니다. 이 정보를 제공하지 않으면 Amazon EFS는 루트 디렉터리를 생성하지 않습니다. 루트 디렉터리가 없으면 액세스 포인트를 사용하는 탑재 시도가 실패합니다.

유형: RootDirectory객체

필수 여부: 아니요

Tags

액세스 포인트와 연결된 태그를 생성합니다. 모든 태그는 키-값 페어이며, 각 키는 고유해야 합니다. 자세한 내용은 AWS 일반 참조 안내서의 AWS 리소스 태그 지정을 참조하세요.

타입: Tag객체 배열

필수: 아니요

응답 구문

```
HTTP/1.1 200
Content-type: application/json

{
```

```
   "AccessPointArn": "string",
   "AccessPointId": "string",
   "ClientToken": "string",
   "FileSystemId": "string",
   "LifeCycleState": "string",
   "Name": "string",
   "OwnerId": "string",
   "PosixUser": {
      "Gid": number,
      "SecondaryGids": [ number ],
      "Uid": number
   },
   "RootDirectory": {
      "CreationInfo": {
         "OwnerGid": number,
         "OwnerUid": number,
         "Permissions": "string"
      },
      "Path": "string"
   },
   "Tags": [
      {
         "Key": "string",
         "Value": "string"
      }
   ]
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반송합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

AccessPointArn

액세스 포인트와 연결된 고유한 Amazon 리소스 이름(ARN)입니다.

유형: 문자열

길이 제약: 최대 길이는 128입니다.

패턴: `^arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:access-point/fsap-[0-9a-f]{8,40}$`

AccessPointId

Amazon EFS에서 할당한 액세스 포인트의 ID입니다.

유형: 문자열

길이 제약: 최대 길이는 128입니다.

패턴: `^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:access-point/fsap-[0-9a-f]{8,40}|fsap-[0-9a-f]{8,40})$`

ClientToken

멱등성 생성을 보장하기 위해 요청에 지정된 불투명한 문자열입니다.

유형: 문자열

길이 제한: 최소 길이는 1. 최대 길이는 64.

패턴: `.+`

FileSystemId

액세스 포인트가 적용되는 EFS 파일 시스템의 ID입니다.

유형: 문자열

길이 제약: 최대 길이는 128입니다.

패턴: `^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$`

LifeCycleState

액세스 포인트의 수명 주기 단계를 식별합니다.

타입: 문자열

유효 값: `creating | available | updating | deleting | deleted | error`

Name

이 액세스 포인트의 이름입니다. `Name` 태그의 값입니다.

유형: 문자열

OwnerId

액세스 포인트 리소스를 소유 AWS 계정 한를 식별합니다.

유형: 문자열

길이 제약 조건: 최대 길이는 14입니다.

패턴: `^(\d{12})|(\d{4}-\d{4}-\d{4})$`

PosixUser

액세스 포인트를 사용하여 NFS 클라이언트가 수행하는 모든 파일 작업에 사용되는 액세스 포인트에서 사용자 ID, 그룹 ID 및 보조 그룹 ID를 포함한 전체 POSIX ID입니다.

유형: PosixUser객체

RootDirectory

액세스 포인트가 액세스 포인트를 사용하는 NFS 클라이언트에 루트 디렉터리로 노출하는 EFS 파일 시스템의 디렉터리입니다.

유형: RootDirectory객체

Tags

액세스 포인트와 관련된 태그는 태그 객체의 배열로 표시됩니다.

타입: Tag 객체 배열

오류

AccessPointAlreadyExists

생성하려는 액세스 포인트가 요청에서 제공한 생성 토큰과 함께 이미 존재하는 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인

오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 409

AccessPointLimitExceeded

AWS 계정 가 파일 시스템당 허용되는 최대 액세스 포인트 수를 이미 생성한 경우 반환됩니다. 자세한 내용은 https://docs.aws.amazon.com/efs/latest/ug/limits.html#limits-efs-resources-per-account-per-region 섹션을 참조하세요.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 403

BadRequest

요청이 잘못되었거나 잘못된 파라미터 값 또는 필수 파라미터 누락 등의 오류가 있는 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 400

FileSystemNotFound

지정된 `FileSystemId` 값이 요청자의에 없는 경우 반환됩니다 AWS 계정.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 404

IncorrectFileSystemLifeCycleState

파일 시스템의 수명 주기 상태가 “사용 가능”이 아닌 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 409

InternalServerError

서버 측에서 오류가 발생한 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 500

ThrottlingException

`CreateAccessPoint` API 작업이 너무 빨리 직접 호출되고 파일 시스템의 액세스 포인트 수가 120개 제한에 근접할 때 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 429

참고 항목

언어별 AWS SDKs

- AWS 명령줄 인터페이스 V2
- AWS .NET V4용 SDK
- AWS SDK for C++
- AWS Go용 SDK v2
- AWS Java V2용 SDK
- AWS SDK for JavaScript V3

- AWS SDK for Kotlin
- AWS PHP V3용 SDK
- AWS Python용 SDK
- AWS SDK for Ruby V3

CreateFileSystem

빈 파일 시스템을 새로 생성합니다. 이 작업에는 Amazon EFS가 멱등성 생성을 보장하기 위해 사용하는 요청에 생성 토큰이 필요합니다(동일한 생성 토큰으로 작업을 직접 호출해도 효과가 없음). AWS 계정 지정된 생성 토큰으로 호출자의가 소유한 파일 시스템이 현재 존재하지 않는 경우이 작업은 다음을 수행합니다.

- 빈 파일 시스템을 새로 생성합니다. 파일 시스템에는 Amazon EFS 할당 ID와 초기 수명 주기 상태 `creating`이 있습니다.
- 생성된 파일 시스템의 설명과 함께 반환됩니다.

그렇지 않으면 이 작업을 수행하면 기존 파일 시스템의 ID와 함께 `FileSystemAlreadyExists` 오류가 반환됩니다.

> **Note**
>
> 기본 사용 사례의 경우 무작위로 생성된 UUID를 생성 토큰으로 사용할 수 있습니다.

멱등성 작업을 사용하면 추가 파일 시스템을 만들 위험 없이 `CreateFileSystem` 직접 호출을 재시도할 수 있습니다. 이는 초기 직접 호출이 실패하여 파일 시스템이 실제로 생성되었는지 여부가 불확실할 때 발생할 수 있습니다. 전송 수준 제한 시간이 초과되었거나 연결이 재설정되는 경우를 예로 들 수 있습니다. 동일한 생성 토큰을 사용하는 한, 초기 직접 호출에서 파일 시스템 생성에 성공한 경우 클라이언트는 `FileSystemAlreadyExists` 오류를 통해 파일 시스템의 존재를 알 수 있습니다.

자세한 내용은 Amazon EFS 사용 설명서에서 파일 시스템 만들기를 참조하세요.

> **Note**
>
> 파일 시스템의 수명 주기 상태가 `creating`으로 유지되는 동안 `CreateFileSystem` 직접 호출이 반환됩니다. DescribeFileSystems 작업을 호출하여 파일 시스템 생성 상태를 확인할 수 있습니다. 이 작업을 직접 호출하면 무엇보다도 파일 시스템 상태가 반환됩니다.

이 작업은 파일 시스템에 대해 선택한 선택적 `PerformanceMode` 파라미터를 받아들입니다. 모든 파일 시스템에 `generalPurpose` `PerformanceMode`를 사용하는 것이 좋습니다. 이 `maxIO` 모드는 `generalPurpose` 모드보다 긴 지연 시간을 견딜 수 있는 고도로 병렬화된 워크로드용으로 설계된

이전 세대 성능 유형입니다. 탄력적 처리량을 사용하는 파일 시스템 또는 One Zone 파일 시스템에는 MaxIO 모드가 지원되지 않습니다.

파일 시스템을 만든 후에는 PerformanceMode를 변경할 수 없습니다. 자세한 내용은 Amazon EFS 성능 모드 섹션을 참조하세요.

ThroughputMode 파라미터를 사용하여 파일 시스템에 대한 처리량 모드를 설정할 수 있습니다.

파일 시스템이 완전히 생성되면 Amazon EFS는 수명 주기 상태를 available로 설정합니다. 이 시점에서 VPC의 파일 시스템에 대한 탑재 대상을 하나 이상 생성할 수 있습니다. 자세한 내용은 CreateMountTarget 단원을 참조하십시오. 탑재 대상을 사용해 VPC의 EC2 인스턴스에서 Amazon EFS 파일 시스템을 탑재합니다. 자세한 내용은 Amazon EFS: 작동 방식을 참조하세요.

이 작업에는 elasticfilesystem:CreateFileSystem 액션에 대한 권한이 필요합니다.

파일 시스템은 생성 시 태그를 지정할 수 있습니다. 생성 작업에서 태그가 지정되면 IAM은 elasticfilesystem:TagResource 작업에서 추가 권한 부여를 수행해 사용자에게 태그를 생성할 권한이 있는지 확인합니다. 따라서 사용자는 elasticfilesystem:TagResource 작업을 사용할 명시적 권한도 가지고 있어야 합니다. 자세한 내용은 생성 시 리소스 태그 지정에 대한 권한 부여를 참조하세요.

Request Syntax

```
POST /2015-02-01/file-systems HTTP/1.1
Content-type: application/json

{
   "AvailabilityZoneName": "string",
   "Backup": boolean,
   "CreationToken": "string",
   "Encrypted": boolean,
   "KmsKeyId": "string",
   "PerformanceMode": "string",
   "ProvisionedThroughputInMibps": number,
   "Tags": [
      {
         "Key": "string",
         "Value": "string"
      }
   ],
   "ThroughputMode": "string"
}
```

URI 요청 파라미터

요청은 URI 파라미터를 사용하지 않습니다.

요청 본문

요청은 JSON 형식으로 다음 데이터를 받습니다.

AvailabilityZoneName

One Zone 파일 시스템의 경우 파일 시스템을 생성할 AWS 가용 영역을 지정합니다. `us-east-1a` 형식을 사용하여 가용 영역을 지정합니다. One Zone 파일 시스템에 대한 자세한 내용은 Amazon EFS 사용 설명서의 EFS 파일 시스템 유형 섹션을 참조하십시오.

> **Note**
>
> Amazon EFS를 사용할 수 AWS 리전 있는의 일부 가용 영역에서는 One Zone 파일 시스템을 사용할 수 없습니다.

유형: 문자열

길이 제한: 최소 길이는 1. 최대 길이는 64.

패턴: `.+`

필수 여부: 아니요

Backup

생성 중인 파일 시스템에서 자동 백업을 활성화할지 여부를 지정합니다. 값을 `true`로 설정하여 자동 백업을 활성화합니다. One Zone 파일 시스템을 생성하는 경우 자동 백업이 기본적으로 활성화됩니다. 자세한 내용은 Amazon EFS 사용 설명서의 자동 백업 섹션을 참조하세요.

기본값은 `false`입니다. 그러나 `AvailabilityZoneName`을 지정하는 경우 기본값은 `true`입니다.

> **Note**
>
> AWS Backup Amazon EFS를 사용할 수 AWS 리전 있는 모든에서를 사용할 수 있는 것은 아닙니다.

유형: 부울

필수 항목 여부: 아니요

CreationToken

최대 64자의 ASCII 문자로 구성된 문자열입니다. Amazon EFS는 이를 사용하여 멱등성 생성을 보장합니다.

유형: 문자열

길이 제한: 최소 길이는 1. 최대 길이는 64.

패턴: `.+`

필수 여부: 예

Encrypted

true인 경우 암호화된 파일 시스템을 생성하는 부울 값입니다. 암호화된 파일 시스템을 생성할 때 기존 AWS Key Management Service 키(KMS 키)를 지정할 수 있습니다. KMS 키를 지정하지 않으면 암호화된 파일 시스템을 보호하는 데 Amazon EFS의 기본 KMS 키(`/aws/elasticfilesystem`)가 사용됩니다.

유형: 부울

필수 항목 여부: 아니요

KmsKeyId

암호화된 파일 시스템을 보호하는 데 사용할 KMS 키의 ID입니다. 기본값이 아닌 KMS 키를 사용할 경우에만 이 파라미터가 필수입니다. 이 파라미터가 지정되지 않으면 Amazon EFS의 기본 KMS 키가 사용됩니다. 다음 형식을 사용하여 KMS 키를 지정할 수 있습니다.

- 키 ID - 키의 고유 식별자(예: `1234abcd-12ab-34cd-56ef-1234567890ab`)입니다.
- ARN - 키의 Amazon 리소스 이름(ARN)입니다(예: `arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab`).
- 키 별칭 - 키에 대해 이전에 생성된 표시 이름입니다(예: `alias/projectKey1`).
- 키 별칭 ARN - 키 별칭의 ARN입니다(예: `arn:aws:kms:us-west-2:444455556666:alias/projectKey1`).

`KmsKeyId`를 사용하는 경우 CreateFileSystem:Encrypted 파라미터를 true로 설정해야 합니다.

> **Important**
>
> EFS는 대칭 KMS 키만 허용합니다. Amazon EFS 파일 시스템에서는 비대칭 KMS 키를 사용할 수 없습니다.

유형: 문자열

길이 제약: 최대 길이 2048.

패턴: `^([0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}|mrk-[0-9a-f]{32}|alias/[a-zA-Z0-9/_-]+|(arn:aws[-a-z]*:kms:[a-z0-9-]+:\d{12}:((key/[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12})|(key/mrk-[0-9a-f]{32})|(alias/[a-zA-Z0-9/_-]+))))$`

필수 여부: 아니요

PerformanceMode

파일 시스템의 성능 모드입니다. 모든 파일 시스템에 `generalPurpose` 성능 모드를 사용하는 것이 좋습니다. `maxIO` 성능 모드를 사용하는 파일 시스템은 대부분의 파일 작업에 대해 지연 시간이 약간 더 길어지는 대가로 더 높은 수준의 집계 처리량 및 초당 작업 수로 확장할 수 있습니다. 파일 시스템을 만든 후에는 성능 모드를 변경할 수 없습니다. One Zone 파일 시스템에서는 `maxIO` 모드가 지원되지 않습니다.

> **Important**
>
> 최대 I/O에서는 작업당 지연 시간이 길어지기 때문에 모든 파일 시스템에 범용 성능 모드를 사용하는 것이 좋습니다.

기본값은 `generalPurpose`입니다.

타입: 문자열

유효 값: `generalPurpose | maxIO`

필수 여부: 아니요

ProvisionedThroughputInMibps

생성 중인 파일 시스템에 대해 프로비저닝하고자 하는 처리량입니다. 초당 메비바이트(MiB/s)로 측정됩니다. ThroughputMode이 provisioned로 설정된 경우 필수입니다. 유효한 값은 1~3414MiBps이며 상한은 리전에 따라 다릅니다. 이 제한을 늘리려면에 문의하세요 지원. 자세한 내용을 알아보려면 Amazon Elastic File System 사용 설명서의 증가할 수 있는 Amazon EFS 할당량을 참조하세요.

유형: Double

유효한 범위: 최소값은 1.0입니다.

필수 여부: 아니요

Tags

파일 시스템과 연결된 하나 이상의 태그를 생성하는 데 사용합니다. 각 태그는 사용자가 정의하는 키-값 페어입니다. 생성 시 "Key":"Name","Value":"{value}" 키-값 페어를 포함시켜 파일 시스템 이름을 지정합니다. 각 키는 고유해야 합니다. 자세한 내용은 AWS 일반 참조 안내서의 AWS 리소스 태그 지정을 참조하세요.

타입: Tag객체 배열

필수: 아니요

ThroughputMode

파일 시스템의 처리량 모드를 지정합니다. 모드는 bursting, provisioned 또는 elastic일 수 있습니다. ThroughputMode를 provisioned로 설정하면 ProvisionedThroughputInMibps의 값도 설정해야 합니다. 파일 시스템을 생성한 후 파일 시스템의 프로비저닝 처리량을 줄이거나 특정 시간 제한이 있는 처리량 모드 간에 변경할 수 있습니다. 자세한 내용을 알아보려면 Amazon Elastic File System 사용 설명서의 프로비저닝 모드를 사용하여 처리량 지정을 참조하세요.

기본값은 bursting입니다.

타입: 문자열

유효 값: bursting | provisioned | elastic

필수 항목 여부: 아니요

응답 구문

```
HTTP/1.1 201
Content-type: application/json

{
   "AvailabilityZoneId": "string",
   "AvailabilityZoneName": "string",
   "CreationTime": number,
   "CreationToken": "string",
   "Encrypted": boolean,
   "FileSystemArn": "string",
   "FileSystemId": "string",
   "FileSystemProtection": {
      "ReplicationOverwriteProtection": "string"
   },
   "KmsKeyId": "string",
   "LifeCycleState": "string",
   "Name": "string",
   "NumberOfMountTargets": number,
   "OwnerId": "string",
   "PerformanceMode": "string",
   "ProvisionedThroughputInMibps": number,
   "SizeInBytes": {
      "Timestamp": number,
      "Value": number,
      "ValueInArchive": number,
      "ValueInIA": number,
      "ValueInStandard": number
   },
   "Tags": [
      {
         "Key": "string",
         "Value": "string"
      }
   ],
   "ThroughputMode": "string"
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 201 응답을 다시 전송합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

AvailabilityZoneId

파일 시스템이 위치한 가용 영역의 고유하고 일관된 식별자로, One Zone 파일 시스템에만 유효합니다. 예를 들어 `use1-az1`는 us-east-1의 가용 영역 ID AWS 리전이며 모든에서 위치가 동일합니다 AWS 계정.

유형: 문자열

AvailabilityZoneName

파일 시스템이 있는 AWS 가용 영역을 설명하며 One Zone 파일 시스템에만 유효합니다. 자세한 내용을 알아보려면 Amazon EFS 사용 설명서의 EFS 스토리지 클래스 사용을 참조하세요.

유형: 문자열

길이 제한: 최소 길이는 1. 최대 길이는 64.

패턴: `.+`

CreationTime

파일 시스템이 생성된 시간(초)입니다(1970-01-01T00:00:00Z 이후).

유형: 타임스탬프

CreationToken

요청에 지정된 불투명한 문자열입니다.

유형: 문자열

길이 제한: 최소 길이는 1. 최대 길이는 64.

패턴: `.+`

Encrypted

true인 경우 파일 시스템이 암호화되었음을 나타내는 부울 값입니다.

유형: Boolean

FileSystemArn

EFS 파일 시스템의 Amazon 리소스 이름(ARN)으로서 `arn:aws:elasticfilesystem:`*`region`*`:`*`account-id`*`:file-system/`*`file-`*

system-id 형식입니다. 샘플 데이터를 사용한 예: arn:aws:elasticfilesystem:us-west-2:1111333322228888:file-system/fs-01234567

유형: 문자열

FileSystemId

Amazon EFS에서 할당한 파일 시스템의 ID입니다.

유형: 문자열

길이 제약: 최대 길이는 128입니다.

패턴: ^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$

FileSystemProtection

파일 시스템의 보호를 설명합니다.

유형: FileSystemProtectionDescription객체

KmsKeyId

암호화된 파일 시스템을 보호하는 데 AWS KMS key 사용되는의 ID입니다.

유형: 문자열

길이 제약: 최대 길이 2048.

패턴: ^([0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}|mrk-[0-9a-f]{32}|alias/[a-zA-Z0-9/_-]+|(arn:aws[-a-z]*:kms:[a-z0-9-]+:\d{12}:((key/[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12})|(key/mrk-[0-9a-f]{32})|(alias/[a-zA-Z0-9/_-]+))))$

LifeCycleState

파일 시스템의 수명 주기 단계입니다.

타입: 문자열

유효 값: creating | available | updating | deleting | deleted | error

Name

태그를 포함하여 파일 시스템에 Name 태그를 추가할 수 있습니다. 자세한 내용은 CreateFileSystem 단원을 참조하십시오. 파일 시스템에 Name 태그가 있는 경우 Amazon EFS는 이 필드에 값을 반환합니다.

유형: 문자열

길이 제약: 최대 길이 256.

패턴: `^([\p{L}\p{Z}\p{N}_.:/=+\-@]*)$`

NumberOfMountTargets

파일 시스템에 있는 탑재 대상의 현재 수. 자세한 내용은 CreateMountTarget 단원을 참조하십시오.

유형: 정수

유효 범위: 최소값 0.

OwnerId

파일 시스템을 AWS 계정 생성한 입니다.

유형: 문자열

길이 제약 조건: 최대 길이는 14입니다.

패턴: `^(\d{12})|(\d{4}-\d{4}-\d{4})$`

PerformanceMode

파일 시스템의 성능 모드입니다.

타입: 문자열

유효 값: `generalPurpose | maxIO`

ProvisionedThroughputInMibps

MiBps로 측정되는 프로비저닝된 파일 시스템 처리량을 정의합니다. `provisioned`로 설정된 `ThroughputMode`을 사용하는 파일 시스템에 유효합니다.

유형: Double

유효한 범위: 최소값은 1.0입니다.

SizeInBytes

파일 시스템에서 해당 Value 필드에 저장된 데이터의 가장 최근 알려진 측정 크기(바이트) 및 해당 Timestamp 필드에서 해당 크기가 결정된 시간입니다. Timestamp 값은 1970-01-01T00:00:00Z 이후의 정수 시간(초)입니다. 이 SizeInBytes 값은 파일 시스템의 일관된 스냅샷 크기를 나타내지는 않지만 파일 시스템에 쓰기가 없는 경우 최종적으로 일관성을 유지합니다. 즉, 몇 시간 이상 파일 시스템을 수정하지 않은 경우에만 SizeInBytes가 실제 크기를 나타냅니다. 그렇지 않으면 값이 특정 시점의 파일 시스템 크기와 정확히 일치하지 않습니다.

유형: FileSystemSize객체

Tags

파일 시스템과 연결된 태그로, Tag 객체 배열로 표시됩니다.

타입: Tag객체 배열

ThroughputMode

파일 시스템의 처리량 모드를 표시합니다. 자세한 내용을 알아보려면 Amazon Elastic File System 사용 설명서의 처리량 모드를 참조하세요.

타입: 문자열

유효 값: bursting | provisioned | elastic

오류

BadRequest

요청이 잘못되었거나 잘못된 파라미터 값 또는 필수 파라미터 누락 등의 오류가 있는 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인

오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 400

FileSystemAlreadyExists

생성하려는 파일 시스템이 사용자가 제공한 생성 토큰과 함께 이미 존재하는 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 409

FileSystemLimitExceeded

AWS 계정 가 계정당 허용되는 최대 파일 시스템 수를 이미 생성한 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 403

InsufficientThroughputCapacity

추가 처리량을 프로비저닝할 용량이 충분하지 않은 경우 반환됩니다. 프로비저닝된 처리량 모드에서 파일 시스템을 생성하려고 할 때, 기존 파일 시스템의 프로비저닝된 처리량을 늘리려고 할 때 또

는 기존 파일 시스템을 버스팅 처리량에서 프로비저닝된 처리량 모드로 변경하려고 할 때 이 값이 반환될 수 있습니다. 나중에 다시 시도해 주세요.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 503

InternalServerError

서버 측에서 오류가 발생한 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 500

ThroughputLimitExceeded

처리량 한도인 1024MiB/s에 도달하여 처리량 모드 또는 프로비저닝된 처리량을 변경할 수 없는 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 400

UnsupportedAvailabilityZone

요청된 Amazon EFS 기능을 지정된 가용 영역에서 사용할 수 없는 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 400

예

암호화된 EFS 파일 시스템 생성

다음 예제는 자동 백업이 활성화된 상태로 `us-west-2` 리전에 파일 시스템을 생성하라는 POST 요청을 보냅니다. 요청은 `myFileSystem1`을 멱등성을 위한 생성 토큰으로 지정합니다.

샘플 요청

```
POST /2015-02-01/file-systems HTTP/1.1
Host: elasticfilesystem.us-west-2.amazonaws.com
x-amz-date: 20140620T215117Z
Authorization: <...>
Content-Type: application/json
```

```
Content-Length: 42

{
  "CreationToken" : "myFileSystem1",
  "PerformanceMode" : "generalPurpose",
  "Backup": true,
  "Encrypted": true,
  "Tags":[
      {
         "Key": "Name",
         "Value": "Test Group1"
      }
   ]
}
```

샘플 응답

```
HTTP/1.1 201 Created
x-amzn-RequestId: 01234567-89ab-cdef-0123-456789abcdef
Content-Type: application/json
Content-Length: 319

{
   "ownerId":"251839141158",
   "CreationToken":"myFileSystem1",
   "Encrypted": true,
   "PerformanceMode" : "generalPurpose",
   "fileSystemId":"fs-01234567",
   "CreationTime":"1403301078",
   "LifeCycleState":"creating",
   "numberOfMountTargets":0,
   "SizeInBytes":{
       "Timestamp": 1403301078,
       "Value": 29313618372,
       "ValueInArchive": 201156,
       "ValueInIA": 675432,
       "ValueInStandard": 29312741784
   },
   "Tags":[
      {
        "Key": "Name",
        "Value": "Test Group1"
      }
```

```
   ],
   "ThroughputMode": "elastic"
}
```

One Zone 가용성을 갖춘 암호화된 EFS 파일 시스템 생성

다음 예제는 자동 백업이 활성화된 상태로 us-west-2 리전에 파일 시스템을 생성하라는 POST 요청을 보냅니다. 파일 시스템은 us-west-2b 가용 영역에 One Zone 스토리지를 갖게 됩니다.

샘플 요청

```
POST /2015-02-01/file-systems HTTP/1.1
Host: elasticfilesystem.us-west-2.amazonaws.com
x-amz-date: 20140620T215117Z
Authorization: <...>
Content-Type: application/json
Content-Length: 42

{
  "CreationToken" : "myFileSystem2",
  "PerformanceMode" : "generalPurpose",
  "Backup": true,
  "AvailabilityZoneName": "us-west-2b",
  "Encrypted": true,
  "ThroughputMode": "elastic",
  "Tags":[
      {
         "Key": "Name",
         "Value": "Test Group1"
      }
   ]
}
```

샘플 응답

```
HTTP/1.1 201 Created
x-amzn-RequestId: 01234567-89ab-cdef-0123-456789abcdef
Content-Type: application/json
Content-Length: 319

{
   "ownerId":"251839141158",
   "CreationToken":"myFileSystem1",
```

```
    "Encrypted": true,
    "AvailabilityZoneId": "usew2-az2",
    "AvailabilityZoneName": "us-west-2b",
    "PerformanceMode" : "generalPurpose",
    "fileSystemId":"fs-01234567",
    "CreationTime":"1403301078",
    "LifeCycleState":"creating",
    "numberOfMountTargets":0,
    "SizeInBytes":{
        "Timestamp": 1403301078,
        "Value": 29313618372,
        "ValueInArchive": 201156,
        "ValueInIA": 675432,
        "ValueInStandard": 29312741784
    },
    "Tags":[
       {
         "Key": "Name",
         "Value": "Test Group1"
       }
    ],
    "ThroughputMode": "elastic"
}
```

참고

언어별 AWS SDKs

- AWS 명령줄 인터페이스 V2
- AWS .NET V4용 SDK
- AWS SDK for C++
- AWS Go v2용 SDK
- AWS Java V2용 SDK
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS PHP V3용 SDK
- AWS Python용 SDK
- AWS SDK for Ruby V3

CreateMountTarget

파일 시스템의 탑재 대상을 생성합니다. 그런 다음 탑재 대상을 사용하여 EC2 인스턴스에 파일 시스템을 탑재할 수 있습니다. 탑재 대상에 대한 자세한 내용은 탑재 대상 관리를 참조하세요.

파일 시스템에 대해 탑재 대상을 생성하려면 파일 시스템의 수명 주기 상태가 `available`이어야 합니다. 자세한 내용은 DescribeFileSystems 단원을 참조하십시오.

탑재 대상 생성 후 Amazon EFS는 `MountTargetId` 및 IP 주소(`IpAddress` 또는 `Ipv6Address`)를 포함한 응답을 반환합니다. EC2 인스턴스에 파일 시스템을 탑재할 때 이 IP 주소를 사용합니다. 파일 시스템을 탑재할 때 탑재 대상의 DNS 이름을 사용할 수도 있습니다. 탑재 대상을 사용하여 파일 시스템을 탑재한 EC2 인스턴스에서는 탑재 대상의 IP 주소에 대해 탑재 대상의 DNS 이름을 확인합니다. 자세한 내용은 Amazon EFS 작동 방식을 참조하세요.

파일 시스템의 탑재 대상은 하나의 VPC에서만 만들 수 있으며, 가용 영역당 탑재 대상은 하나씩만 있을 수 있습니다. 자세한 내용은 탑재 대상 생성을 참조하세요.

요청이 요구 사항을 충족하는 경우 Amazon EFS는 다음을 수행합니다.

- 지정된 서브넷에 새 탑재 대상을 생성합니다.
- 서브넷에 다음과 같은 새 네트워크 인터페이스를 생성합니다.
  - 설명 `Mount target` *`fsmt-id`* `for file system` *`fs-id`* 여기서 *`fsmt-id`* 는 탑재 대상 ID이며, *`fs-id`* 는 `FileSystemId`입니다.
  - 네트워크 인터페이스의 `requesterManaged` 속성은 `true`로 설정되고, `requesterId` 값은 `EFS`로 설정됩니다.

  각 탑재 대상에는 해당하는 요청자 관리형 EC2 네트워크 인터페이스가 하나씩 있습니다. 네트워크 인터페이스가 생성되면 Amazon EFS는 탑재 대상 설명의 `NetworkInterfaceId` 필드를 네트워크 인터페이스 ID로 설정하고 IP 주소를 해당 주소로 설정합니다. 네트워크 인터페이스 생성에 실패하면 전체 `CreateMountTarget` 작업이 실패합니다.

> **Note**
>
> 네트워크 인터페이스를 생성한 후에만 `CreateMountTarget` 직접 호출이 반환되지만 탑재 대상 상태가 여전히 `creating`인 동안에는 DescribeMountTargets 작업을 직접 호출하여 탑재 대상 생성 상태를 확인할 수 있습니다. 이 경우 무엇보다도 탑재 대상 상태가 반환됩니다.

이 작업에는 파일 시스템에서 다음 작업에 대한 권한이 필요합니다.

- `elasticfilesystem:CreateMountTarget`

이 작업을 수행하려면 다음과 같은 Amazon EC2 작업에 대한 권한도 필요합니다.

- `ec2:DescribeSubnets`
- `ec2:DescribeNetworkInterfaces`
- `ec2:CreateNetworkInterface`

Request Syntax

```
POST /2015-02-01/mount-targets HTTP/1.1
Content-type: application/json

{
   "FileSystemId": "string",
   "IpAddress": "string",
   "IpAddressType": "string",
   "Ipv6Address": "string",
   "SecurityGroups": [ "string" ],
   "SubnetId": "string"
}
```

URI 요청 파라미터

요청은 URI 파라미터를 사용하지 않습니다.

요청 본문

요청은 JSON 형식으로 다음 데이터를 받습니다.

FileSystemId

탑재 대상을 생성할 파일 시스템의 ID입니다.

유형: 문자열

길이 제약: 최대 길이는 128입니다.

패턴: `^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$`

필수 여부: 예

IpAddress

탑재 대상의 `IpAddressType`이 IPv4(`IPV4_ONLY` 또는 `DUAL_STACK`)인 경우 사용할 IPv4 주소를 지정합니다. `IpAddress`를 지정하지 않으면 Amazon EFS가 `SubnetId`로 지정된 서브넷에서 사용되지 않는 IP 주소를 선택합니다.

유형: 문자열

길이 제약: 최소 길이는 7입니다. 최대 길이는 15입니다.

패턴: `^[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}$`

필수 여부: 아니요

IpAddressType

탑재 대상의 IP 주소입니다. 가능한 값은 `IPV4_ONLY`(IPv4 주소만), `IPV6_ONLY`(IPv6 주소만), `DUAL_STACK`(듀얼 스택, IPv4 및 IPv6 주소 모두)입니다. `IpAddressType`를 지정하지 않으면 `IPV4_ONLY`가 사용됩니다.

> **Note**
>
> `IPAddressType`은 서브넷의 IP 유형과 일치해야 합니다. 또한 `IPAddressType` 파라미터는 VPC의 서브넷에 대한 기본 IP 주소로 설정된 값을 재정의합니다. 예를 들어 `IPAddressType`이 `IPV4_ONLY`이고 `AssignIpv6AddressOnCreation`이 `true`인 경우 탑재 대상에 IPv4가 사용됩니다. 자세한 내용은 서브넷의 IP 주소 지정 속성 수정을 참조하세요.

타입: 문자열

유효 값: `IPV4_ONLY | IPV6_ONLY | DUAL_STACK`

필수 여부: 아니요

Ipv6Address

탑재 대상의 IPAddressType이 IPv6(IPV6_ONLY 또는 DUAL_STACK)인 경우 사용할 IPv6 주소를 지정합니다. Ipv6Address를 지정하지 않으면 Amazon EFS가 SubnetId로 지정된 서브넷에서 사용되지 않는 IP 주소를 선택합니다.

유형: 문자열

길이 제약 조건: 최소 길이는 3입니다. 최대 길이는 39입니다.

필수 여부: 아니요

SecurityGroups

VPC 보안 그룹 ID(sg-xxxxxxxx 형식)입니다. 이들은 지정된 서브넷과 동일한 VPC에 속해야 합니다. 보안 그룹의 최대 개수는 계정 할당량에 따라 다릅니다. 자세한 내용은 Amazon VPC 사용 설명서의 Amazon VPC 할당량을 참조하세요(보안 그룹 테이블 참조). 보안 그룹을 지정하지 않으면 Amazon EFS는 해당 서브넷 VPC의 기본 보안 그룹을 사용합니다

유형: 문자열 어레이

배열 멤버: 최대 항목 수는 100개입니다.

길이 제약: 최소 길이는 11입니다. 최대 길이는 43입니다.

패턴: ^sg-[0-9a-f]{8,40}

필수 여부: 아니요

SubnetId

탑재 대상을 추가할 서브넷의 ID입니다. One Zone 파일 시스템의 경우 파일 시스템의 가용 영역에 연결된 서브넷을 사용합니다. 서브넷 유형은 IpAddressType과 동일한 유형이어야 합니다.

유형: 문자열

길이 제약: 최소 길이는 15입니다. 최대 길이는 47입니다.

패턴: ^subnet-[0-9a-f]{8,40}$

필수 여부: 예

응답 구문

```
HTTP/1.1 200
```

```
Content-type: application/json

{
   "AvailabilityZoneId": "string",
   "AvailabilityZoneName": "string",
   "FileSystemId": "string",
   "IpAddress": "string",
   "Ipv6Address": "string",
   "LifeCycleState": "string",
   "MountTargetId": "string",
   "NetworkInterfaceId": "string",
   "OwnerId": "string",
   "SubnetId": "string",
   "VpcId": "string"
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반송합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

AvailabilityZoneId

탑재 대상이 위치한 가용 영역의 고유하고 일관된 식별자입니다. 예를 들어 `use1-az1`는 us-east-1 리전의 AZ ID이며 모든에서 위치가 동일합니다 AWS 계정.

유형: 문자열

AvailabilityZoneName

탑재 대상이 위치한 가용 영역의 이름입니다. 가용 영역은 각 가용 영역의 이름에 독립적으로 매핑됩니다 AWS 계정. 예를 들어의 가용 영역은 다른의 가용 영역과 동일한 위치가 아닐 `us-east-1a` AWS 계정 수 `us-east-1a` 있습니다 AWS 계정.

유형: 문자열

길이 제한: 최소 길이는 1. 최대 길이는 64.

패턴: `.+`

FileSystemId

탑재 대상으로 의도된 파일 시스템의 ID입니다.

유형: 문자열

길이 제약: 최대 길이는 128입니다.

패턴: `^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$`

IpAddress

탑재 대상의 IPv4 주소입니다.

유형: 문자열

길이 제약: 최소 길이는 7입니다. 최대 길이는 15입니다.

패턴: `^[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}$`

Ipv6Address

탑재 대상의 IPv6 주소입니다.

유형: 문자열

길이 제약 조건: 최소 길이는 3입니다. 최대 길이는 39입니다.

LifeCycleState

탑재 대상의 생명 주기 상태입니다.

타입: 문자열

유효 값: `creating | available | updating | deleting | deleted | error`

MountTargetId

시스템에서 할당한 탑재 대상 ID.

유형: 문자열

길이 제약: 최소 길이는 13입니다. 최대 길이는 45입니다.

패턴: `^fsmt-[0-9a-f]{8,40}$`

NetworkInterfaceId

탑재 대상을 생성할 때 Amazon EFS가 생성한 네트워크 인터페이스의 ID입니다.

유형: 문자열

OwnerId

AWS 계정 리소스를 소유한 ID입니다.

유형: 문자열

길이 제약 조건: 최대 길이는 14입니다.

패턴: `^(\d{12})|(\d{4}-\d{4}-\d{4})$`

SubnetId

탑재 대상의 서브넷의 ID입니다.

유형: 문자열

길이 제약: 최소 길이는 15입니다. 최대 길이는 47입니다.

패턴: `^subnet-[0-9a-f]{8,40}$`

VpcId

탑재 대상이 구성된 Virtual Private Cloud(VPC) ID입니다.

유형: 문자열

오류

AvailabilityZonesMismatch

탑재 대상에 지정된 가용 영역이 파일 시스템의 가용 영역과 다를 경우 반환됩니다. 자세한 내용은 리전 및 One Zone 스토리지 중복성을 참조하세요.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야

할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 400

BadRequest

요청이 잘못되었거나 잘못된 파라미터 값 또는 필수 파라미터 누락 등의 오류가 있는 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 400

FileSystemNotFound

지정된 `FileSystemId` 값이 요청자의에 없는 경우 반환됩니다 AWS 계정.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 404

IncorrectFileSystemLifeCycleState

파일 시스템의 수명 주기 상태가 “사용 가능”이 아닌 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 409

InternalServerError

서버 측에서 오류가 발생한 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 500

IpAddressInUse

요청이 서브넷에서 이미 사용 중인 IP 주소(`IpAddress` 또는 `Ipv6Address`)를 지정한 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 409

MountTargetConflict

탑재 대상이 파일 시스템의 기존 탑재 대상을 기준으로 지정된 제한 사항 중 하나를 위반할 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 409

NetworkInterfaceLimitExceeded

직접 호출 계정이 해당 AWS 리전의 탄력적 네트워크 인터페이스 한도에 도달했습니다. 일부 네트워크 인터페이스를 삭제하거나 계정 할당량 상향을 요청하세요. 자세한 내용은 Amazon VPC 사용 설명서의 Amazon VPC 할당량을 참조하세요(네트워크 인터페이스 표의 리전별 네트워크 인터페이스 항목 참조).

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 409

NoFreeAddressesInSubnet

요청에 `IpAddress` 또는 `Ipv6Address`가 제공되지 않았으며 지정된 서브넷에 사용 가능한 IP 주소가 없습니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 409

SecurityGroupLimitExceeded

요청에 지정된 `SecurityGroups`의 수가 계정 할당량에 기반한 한도를 초과할 경우 반환됩니다. 일부 보안 그룹을 삭제하거나 계정 할당량 상향을 요청하세요. 자세한 내용은 Amazon VPC 사용 설명서의 Amazon VPC 할당량을 참조하세요(보안 그룹 테이블 참조).

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야

할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 400

SecurityGroupNotFound

지정된 보안 그룹 중 하나가 서브넷의 Virtual Private Cloud(VPC)에 존재하지 않는 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 400

SubnetNotFound

요청에 ID `SubnetId`가 제공된 서브넷이 없는 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 400

UnsupportedAvailabilityZone

요청된 Amazon EFS 기능을 지정된 가용 영역에서 사용할 수 없는 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 400

예

서브넷의 사용 가능한 IPv4 주소에서 탑재 대상 생성

다음 요청은 탑재 대상에 대한 파일 시스템과 서브넷만 지정합니다. 대상은 VPC와 연결된 기본 보안 그룹을 사용하여 지정된 서브넷의 사용 가능한 IPv4 주소에서 생성됩니다.

샘플 요청

```
POST /2015-02-01/mount-targets HTTP/1.1
Host: elasticfilesystem.us-west-2.amazonaws.com
x-amz-date: 20140620T221118Z
Authorization: <...>
Content-Type: application/json
Content-Length: 160

{"SubnetId": "subnet-748c5d03", "FileSystemId": "fs-01234567"}
```

샘플 응답

```
HTTP/1.1 200 OK
x-amzn-RequestId: 01234567-89ab-cdef-0123-456789abcdef
Content-Type: application/json
```

```
Content-Length: 252

{
    "OwnerId": "231243201240",
    "MountTargetId": "fsmt-55a4413c",
    "FileSystemId": "fs-01234567",
    "SubnetId": "subnet-01234567",
    "LifeCycleState": "available",
    "IpAddress": "172.31.22.183"
    "NetworkInterfaceId":"eni-1bcb7772"
    "AvailabilityZoneId": "eus1-az2",
    "AvailabilityZoneName": "eu-south-1b",
    "VpcId": "vpc-08d45b31fa009a15e"
}
```

특정 IPv4 주소에서 탑재 대상 생성

다음 요청은 탑재 대상에 사용할 파일 시스템, 서브넷, 보안 그룹 및 IPv4 주소를 지정합니다.

샘플 요청

```
POST /2015-02-01/mount-targets HTTP/1.1
Host: elasticfilesystem.us-west-2.amazonaws.com
x-amz-date: 20140620T221118Z
Authorization: <...>
Content-Type: application/json
Content-Length: 160

{
   "FileSystemId":"fs-01234567",
   "SubnetId":"subnet-01234567",
   "IpAddress":"10.0.2.42",
   "SecurityGroups":[
      "sg-01234567"
   ]
}
```

샘플 응답

```
HTTP/1.1 200 OK
x-amzn-RequestId: 01234567-89ab-cdef-0123-456789abcdef
Content-Type: application/json
Content-Length: 252
```

```
{
   "OwnerId":"251839141158",
   "MountTargetId":"fsmt-9a13661e",
   "FileSystemId":"fs-01234567",
   "SubnetId":"subnet-fd04ff94",
   "LifeCycleState":"available",
   "IpAddress":"10.0.2.42",
   "NetworkInterfaceId":"eni-1bcb7772"
   "AvailabilityZoneId": "eus1-az2",
   "AvailabilityZoneName": "eu-south-1b",
   "VpcId": "vpc-08d45b31fa009a15e"
}
```

특정 IPv6 주소에서 탑재 대상 생성

다음 요청은 탑재 대상에 사용할 파일 시스템, 서브넷, 보안 그룹 및 IPv6 주소를 지정합니다.

샘플 요청

```
POST /2015-02-01/mount-targets HTTP/1.1
Host: elasticfilesystem.us-west-2.amazonaws.com
x-amz-date: 20140620T221118Z
Authorization: <...>
Content-Type: application/json
Content-Length: 160

{
   "FileSystemId":"fs-01234567",
   "SubnetId":"subnet-01234567",
   "Ipv6Address":"2001:0db8:85a3:0000:0000:8a2e:0370:7334",
   "IpAddressType": "IPV6_ONLY",
   "SecurityGroups":[
      "sg-01234567"
   ]
}
```

샘플 응답

```
HTTP/1.1 200 OK
x-amzn-RequestId: 01234567-89ab-cdef-0123-456789abcdef
Content-Type: application/json
Content-Length: 252
```

```
{
   "OwnerId":"251839141158",
   "MountTargetId":"fsmt-9a13661e",
   "FileSystemId":"fs-01234567",
   "SubnetId":"subnet-fd04ff94",
   "LifeCycleState":"available",
   "Ipv6Address":"2001:0db8:85a3:0000:0000:8a2e:0370:7334",
   "NetworkInterfaceId":"eni-1bcb7772"
   "AvailabilityZoneId": "eus1-az2",
   "AvailabilityZoneName": "eu-south-1b",
   "VpcId": "vpc-08d45b31fa009a15e"
}
```

듀얼 스택 서브넷에서 사용 가능한 IPv4 주소에 탑재 대상 생성

다음 요청은 탑재 대상의 파일 시스템, 서브넷 및 주소 유형을 지정합니다. 대상은 VPC와 연결된 기본 보안 그룹을 사용하여 지정된 듀얼 스택 서브넷의 사용 가능한 IPv4 주소에서 생성됩니다.

샘플 요청

```
POST /2015-02-01/mount-targets HTTP/1.1
Host: elasticfilesystem.us-west-2.amazonaws.com
x-amz-date: 20140620T221118Z
Authorization: <...>
Content-Type: application/json
Content-Length: 160

{"SubnetId": "subnet-748c5d03", "FileSystemId": "fs-01234567"}
```

샘플 응답

```
HTTP/1.1 200 OK
x-amzn-RequestId: 01234567-89ab-cdef-0123-456789abcdef
Content-Type: application/json
Content-Length: 252

{
    "OwnerId": "251839141158",
    "MountTargetId": "fsmt-55a4413c",
    "FileSystemId": "fs-01234567",
    "SubnetId":"subnet-fd04ff94",
    "LifeCycleState": "available",
```

```
    "IpAddress": "172.31.22.183"
    "Ipv6Address": "2a05:d01a:419:8611:3e57:75ab:5719:b238",
    "NetworkInterfaceId": "eni-01234567",
    "AvailabilityZoneId": "eus1-az2",
    "AvailabilityZoneName": "eu-south-1b",
    "VpcId": "vpc-08d45b31fa009a15e"
}
```

듀얼 스택 서브넷의 특정 Ipv4 및 IPv6 주소에 탑재 대상 생성

다음 요청은 탑재 대상의 파일 시스템, 서브넷, 보안 그룹, IPv4 주소, IPv6 주소를 지정합니다. 대상은 듀얼 스택 서브넷의 지정된 IPv4 및 IPv6 주소에서 생성됩니다.

샘플 요청

```
POST /2015-02-01/mount-targets HTTP/1.1
Host: elasticfilesystem.us-west-2.amazonaws.com
x-amz-date: 20140620T221118Z
Authorization: <...>
Content-Type: application/json
Content-Length: 160

{
   "FileSystemId":"fs-01234567",
   "SubnetId":"subnet-01234567",
   "IpAddressType": "DUAL_STACK",
   "IpAddress": "10.0.1.25",
   "Ipv6Address":"2001:0db8:85a3:0000:0000:8a2e:0370:7334",
   "SecurityGroups":[
      "sg-01234567"
   ]
}
```

샘플 응답

```
HTTP/1.1 200 OK
x-amzn-RequestId: 01234567-89ab-cdef-0123-456789abcdef
Content-Type: application/json
Content-Length: 252

{
    "OwnerId": "231243201240",
    "MountTargetId": "fsmt-55a4413c",
```

```
    "FileSystemId": "fs-01234567",
    "SubnetId": "subnet-01234567",
    "LifeCycleState": "available",
    "IpAddress": "10.0.1.25",
    "Ipv6Address":"2001:0db8:85a3:0000:0000:8a2e:0370:7334",
    "NetworkInterfaceId": "eni-01234567",
    "AvailabilityZoneId": "eus1-az2",
    "AvailabilityZoneName": "eu-south-1b",
    "VpcId": "vpc-08d45b31fa009a15e"
}
```

참고

언어별 AWS SDKs

- AWS 명령줄 인터페이스 V2
- AWS .NET V4용 SDK
- AWS SDK for C++
- AWS Go v2용 SDK
- AWS Java V2용 SDK
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS PHP V3용 SDK
- AWS Python용 SDK
- AWS SDK for Ruby V3

CreateReplicationConfiguration

새 EFS 파일 시스템 또는 기존 EFS 파일 시스템에 대한 복제 구성을 생성합니다. 자세한 내용을 알아보려면 Amazon EFS 사용 설명서의 Amazon EFS 복제를 참조하세요. 다음 복제 구성은 다음을 지정합니다.

- 소스 파일 시스템 - 복제하려는 EFS 파일 시스템입니다.
- 대상 파일 시스템 - 소스 파일 시스템이 복제되는 대상 파일 시스템입니다. 복제 구성에는 대상 파일 시스템이 하나만 있을 수 있습니다.

> **Note**
>
> 파일 시스템은 하나의 복제 구성에만 포함될 수 있습니다.

복제 구성의 대상 파라미터는 새 파일 시스템에 복제하는지 기존 파일 시스템에 복제하는지 여부와 AWS 계정간에 복제하는지에 따라 달라집니다. 자세한 정보는 DestinationToCreate을 참조하세요.

이 작업에는 `elasticfilesystem:CreateReplicationConfiguration` 액션에 대한 권한이 필요합니다. 또한 파일 시스템을 복제하는 방법에 따라 다른 권한이 필요합니다. 자세한 내용은 Amazon EFS 사용 설명서의 복제를 위한 필수 권한을 참조하세요.

Request Syntax

```
POST /2015-02-01/file-systems/SourceFileSystemId/replication-configuration HTTP/1.1
Content-type: application/json

{
   "Destinations": [
      {
         "AvailabilityZoneName": "string",
         "FileSystemId": "string",
         "KmsKeyId": "string",
         "Region": "string",
         "RoleArn": "string"
      }
   ]
}
```

URI 요청 파라미터

요청은 다음 URI 파라미터를 사용합니다.

SourceFileSystemId

복제하려는 Amazon EFS 파일 시스템을 지정합니다. 이 파일 시스템은 이미 다른 복제 구성의 소스 또는 대상 파일 시스템일 수 없습니다.

길이 제약: 최대 길이는 128입니다.

패턴: `^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$`

필수 여부: 예

요청 본문

요청은 JSON 형식으로 다음 데이터를 받습니다.

Destinations

대상 구성 객체의 배열입니다. 대상 구성 객체는 하나만 지원됩니다.

타입: DestinationToCreate객체 배열

필수 항목 여부: 예

응답 구문

```
HTTP/1.1 200
Content-type: application/json

{
   "CreationTime": number,
   "Destinations": [
      {
         "FileSystemId": "string",
         "LastReplicatedTimestamp": number,
         "OwnerId": "string",
         "Region": "string",
```

```
            "RoleArn": "string",
            "Status": "string",
            "StatusMessage": "string"
        }
    ],
    "OriginalSourceFileSystemArn": "string",
    "SourceFileSystemArn": "string",
    "SourceFileSystemId": "string",
    "SourceFileSystemOwnerId": "string",
    "SourceFileSystemRegion": "string"
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반송합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

CreationTime

복제 구성을 만든 시기를 설명합니다.

유형: 타임스탬프

Destinations

대상 객체의 배열입니다. 대상 개체는 하나만 지원됩니다.

타입: Destination객체 배열

OriginalSourceFileSystemArn

복제 구성에 있는 원본 소스 EFS 파일 시스템의 Amazon 리소스 이름(ARN)입니다.

유형: 문자열

SourceFileSystemArn

복제 구성에 있는 현재 소스 파일 시스템의 Amazon 리소스 이름(ARN)입니다.

유형: 문자열

SourceFileSystemId

복제 중인 소스 Amazon EFS 파일 시스템의 ID입니다.

유형: 문자열

길이 제약: 최대 길이는 128입니다.

패턴: `^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$`

SourceFileSystemOwnerId

소스 파일 시스템이 AWS 계정 있는의 ID입니다.

유형: 문자열

길이 제약 조건: 최대 길이는 14입니다.

패턴: `^(\d{12})|(\d{4}-\d{4}-\d{4})$`

SourceFileSystemRegion

소스 EFS 파일 시스템이 있는 AWS 리전 입니다.

유형: 문자열

길이 제한: 최소 길이는 1. 최대 길이는 64.

패턴: `^[a-z]{2}-((iso[a-z]{0,1}-)|(gov-)){0,1}[a-z]+-{0,1}[0-9]{0,1}$`

오류

BadRequest

요청이 잘못되었거나 잘못된 파라미터 값 또는 필수 파라미터 누락 등의 오류가 있는 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야

할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 400

ConflictException

복제본의 소스 파일 시스템은 암호화되었지만 대상 파일 시스템은 암호화되지 않은 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 409

FileSystemLimitExceeded

AWS 계정 가 계정당 허용되는 최대 파일 시스템 수를 이미 생성한 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 403

FileSystemNotFound

지정된 `FileSystemId` 값이 요청자의에 없는 경우 반환됩니다 AWS 계정.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 404

IncorrectFileSystemLifeCycleState

파일 시스템의 수명 주기 상태가 “사용 가능”이 아닌 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 409

InsufficientThroughputCapacity

추가 처리량을 프로비저닝할 용량이 충분하지 않은 경우 반환됩니다. 프로비저닝된 처리량 모드에서 파일 시스템을 생성하려고 할 때, 기존 파일 시스템의 프로비저닝된 처리량을 늘리려고 할 때 또는 기존 파일 시스템을 버스팅 처리량에서 프로비저닝된 처리량 모드로 변경하려고 할 때 이 값이 반환될 수 있습니다. 나중에 다시 시도해 주세요.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 503

InternalServerError

서버 측에서 오류가 발생한 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 500

ReplicationNotFound

지정된 파일 시스템에 복제 구성이 없는 경우 반환됩니다.

ErrorCode

ReplicationNotFound

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야

할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 404

ThroughputLimitExceeded

처리량 한도인 1024MiB/s에 도달하여 처리량 모드 또는 프로비저닝된 처리량을 변경할 수 없는 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 400

UnsupportedAvailabilityZone

요청된 Amazon EFS 기능을 지정된 가용 영역에서 사용할 수 없는 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 400

ValidationException

요청이 이루어진 AWS 리전 에서 AWS Backup 서비스를 사용할 수 없는 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 400

참고

언어별 AWS SDKs

- AWS 명령줄 인터페이스 V2
- AWS .NET V4용 SDK
- AWS SDK for C++
- AWS Go용 SDK v2
- AWS Java V2용 SDK
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS PHP V3용 SDK
- AWS Python용 SDK
- AWS SDK for Ruby V3

CreateTags

> **Note**
>
> 더 이상 사용되지 않음 - `CreateTags`는 더 이상 사용되지 않으며 유지 관리되지 않습니다. EFS 리소스용 태그를 생성하려면 TagResource API 작업을 사용하세요.

파일 시스템과 연결된 태그를 생성하거나 덮어씁니다. 각 태그는 키-값 페어입니다. 요청에 지정된 태그 키가 파일 시스템에 이미 있는 경우 이 작업은 해당 값을 요청에 제공된 값으로 덮어씁니다. 파일 시스템에 `Name` 태그를 추가하면 Amazon EFS는 DescribeFileSystems 작업에 대한 응답으로 태그를 반환합니다.

이 작업에는 `elasticfilesystem:CreateTags` 작업에 대한 권한이 필요합니다.

Request Syntax

```
POST /2015-02-01/create-tags/FileSystemId HTTP/1.1
Content-type: application/json

{
   "Tags": [
      {
         "Key": "string",
         "Value": "string"
      }
   ]
}
```

URI 요청 파라미터

요청은 다음 URI 파라미터를 사용합니다.

FileSystemId

태그를 수정하려는 파일 시스템의 ID입니다(문자열). 이 작업은 태그만 수정하고 파일 시스템은 수정하지 않습니다.

길이 제약: 최대 길이는 128입니다.

패턴: `^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$`

필수 여부: 예

요청 본문

요청은 JSON 형식으로 다음 데이터를 받습니다.

Tags

추가할 Tag 객체 배열. 각 Tag 객체는 키-값 페어입니다.

타입: Tag객체 배열

필수 항목 여부: 예

응답 구문

```
HTTP/1.1 204
```

Response Elements

액션이 성공하면 해당 서비스는 빈 HTTP 본문과 함께 HTTP 204 응답을 되돌려줍니다.

Errors

BadRequest

요청이 잘못되었거나 잘못된 파라미터 값 또는 필수 파라미터 누락 등의 오류가 있는 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야

할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 400

FileSystemNotFound

지정된 `FileSystemId` 값이 요청자의에 없는 경우 반환됩니다 AWS 계정.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 404

InternalServerError

서버 측에서 오류가 발생한 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 500

참고

언어별 AWS SDKs

- AWS 명령줄 인터페이스 V2
- AWS .NET V4용 SDK
- AWS SDK for C++
- AWS Go용 SDK v2
- AWS Java V2용 SDK
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS PHP V3용 SDK
- AWS Python용 SDK
- AWS SDK for Ruby V3

DeleteAccessPoint

지정된 액세스 포인트를 삭제합니다. 삭제가 완료되면 새 클라이언트는 더 이상 액세스 포인트에 연결할 수 없습니다. 삭제 시 액세스 포인트에 연결된 클라이언트는 연결이 종료될 때까지 계속 작동합니다.

이 작업에는 `elasticfilesystem:DeleteAccessPoint` 액션에 대한 권한이 필요합니다.

Request Syntax

```
DELETE /2015-02-01/access-points/AccessPointId HTTP/1.1
```

URI 요청 파라미터

요청은 다음 URI 파라미터를 사용합니다.

AccessPointId

삭제할 액세스 포인트의 ID입니다.

길이 제약: 최대 길이는 128입니다.

패턴: `^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:access-point/fsap-[0-9a-f]{8,40}|fsap-[0-9a-f]{8,40})$`

필수 여부: 예

Request Body

해당 요청에는 본문이 없습니다.

Response Syntax

```
HTTP/1.1 204
```

Response Elements

액션이 성공하면 해당 서비스는 빈 HTTP 본문과 함께 HTTP 204 응답을 되돌려줍니다.

Errors

AccessPointNotFound

지정된 `AccessPointId` 값이 요청자의에 없는 경우 반환됩니다 AWS 계정.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 404

BadRequest

요청이 잘못되었거나 잘못된 파라미터 값 또는 필수 파라미터 누락 등의 오류가 있는 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 400

InternalServerError

서버 측에서 오류가 발생한 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 500

참고

언어별 AWS SDKs

- AWS 명령줄 인터페이스 V2
- AWS .NET V4용 SDK
- AWS SDK for C++
- AWS Go용 SDK v2
- AWS Java V2용 SDK
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS PHP V3용 SDK
- AWS Python용 SDK
- AWS SDK for Ruby V3

DeleteFileSystem

파일 시스템을 삭제하여 해당 콘텐츠에 대한 액세스를 영구적으로 차단합니다. 반환되면 파일 시스템은 더 이상 존재하지 않으며 삭제된 파일 시스템의 콘텐츠에 액세스할 수 없습니다.

EFS 파일 시스템을 삭제하려면 먼저 파일 시스템에 연결된 탑재 대상을 수동으로 삭제해야 합니다. 이 단계는 AWS 콘솔을 사용하여 파일 시스템을 삭제할 때 수행됩니다.

> **Note**
>
> EFS 복제 구성의 일부인 파일 시스템은 삭제할 수 없습니다. 복제 구성을 먼저 삭제해야 합니다.

사용 중인 파일 시스템은 삭제할 수 없습니다. 즉, 파일 시스템에 탑재 대상이 있는 경우 먼저 탑재 대상을 삭제해야 합니다. 자세한 내용은 DescribeMountTargets 및 DeleteMountTarget 섹션을 참조하세요.

> **Note**
>
> 파일 시스템 상태가 아직 `deleting`일 때 `DeleteFileSystem` 직접 호출이 반환됩니다. DescribeFileSystems 작업을 직접 호출하여 파일 시스템 삭제 상태를 확인할 수 있습니다. 그러면 계정의 파일 시스템 목록이 반환됩니다. 삭제된 파일 시스템의 파일 시스템 ID 또는 생성 토큰을 전달하면 DescribeFileSystems가 `404 FileSystemNotFound` 오류를 반환합니다.

이 작업에는 `elasticfilesystem:DeleteFileSystem` 액션에 대한 권한이 필요합니다.

Request Syntax

```
DELETE /2015-02-01/file-systems/FileSystemId HTTP/1.1
```

URI 요청 파라미터

요청은 다음 URI 파라미터를 사용합니다.

FileSystemId

삭제하려는 파일 시스템의 ID입니다.

길이 제약: 최대 길이는 128입니다.

패턴: `^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$`

필수 여부: 예

Request Body

해당 요청에는 본문이 없습니다.

Response Syntax

```
HTTP/1.1 204
```

Response Elements

액션이 성공하면 해당 서비스는 빈 HTTP 본문과 함께 HTTP 204 응답을 되돌려줍니다.

Errors

BadRequest

요청이 잘못되었거나 잘못된 파라미터 값 또는 필수 파라미터 누락 등의 오류가 있는 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 400

FileSystemInUse

파일 시스템에 탑재 대상이 있는 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 409

FileSystemNotFound

지정된 `FileSystemId` 값이 요청자의에 없는 경우 반환됩니다 AWS 계정.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 404

InternalServerError

서버 측에서 오류가 발생한 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 500

예제

파일 시스템 삭제

다음 예에서는 ID가 `fs-01234567`인 파일 시스템을 삭제하기 위해 `file-systems` 엔드포인트(`elasticfilesystem.us-west-2.amazonaws.com/2015-02-01/file-systems/fs-01234567`)에 DELETE 요청을 보냅니다.

샘플 요청

```
DELETE /2015-02-01/file-systems/fs-01234567 HTTP/1.1
Host: elasticfilesystem.us-west-2.amazonaws.com
x-amz-date: 20140622T233021Z
Authorization: <...>
```

샘플 응답

```
HTTP/1.1 204 No Content
x-amzn-RequestId: a2d125b3-7ebd-4d6a-ab3d-5548630bff33
Content-Length: 0
```

참고

언어별 AWS SDKs

- AWS 명령줄 인터페이스 V2
- AWS .NET V4용 SDK
- AWS SDK for C++
- AWS Go용 SDK v2

- AWS Java V2용 SDK
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS PHP V3용 SDK
- AWS Python용 SDK
- AWS SDK for Ruby V3

DeleteFileSystemPolicy

지정된 파일 시스템의 FileSystemPolicy를 삭제합니다. 기존 정책이 삭제되면 FileSystemPolicy 기본값이 적용됩니다. 기본 파일 시스템 정책에 대한 자세한 내용은 EFS에서의 리소스 기반 정책 사용을 참조하세요.

이 작업에는 elasticfilesystem:DeleteFileSystemPolicy 액션에 대한 권한이 필요합니다.

Request Syntax

```
DELETE /2015-02-01/file-systems/FileSystemId/policy HTTP/1.1
```

URI 요청 파라미터

요청은 다음 URI 파라미터를 사용합니다.

FileSystemId

FileSystemPolicy를 삭제할 EFS 파일 시스템을 지정합니다.

길이 제약: 최대 길이는 128입니다.

패턴: ^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$

필수 여부: 예

Request Body

해당 요청에는 본문이 없습니다.

Response Syntax

```
HTTP/1.1 200
```

Response Elements

작업이 성공하면 서비스가 비어 있는 HTTP 본문과 함께 HTTP 200 응답을 다시 전송합니다.

오류

BadRequest

요청이 잘못되었거나 잘못된 파라미터 값 또는 필수 파라미터 누락 등의 오류가 있는 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 400

FileSystemNotFound

지정된 `FileSystemId` 값이 요청자의에 없는 경우 반환됩니다 AWS 계정.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 404

IncorrectFileSystemLifeCycleState

파일 시스템의 수명 주기 상태가 “사용 가능”이 아닌 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 409

InternalServerError

서버 측에서 오류가 발생한 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 500

참고

언어별 AWS SDKs

- AWS 명령줄 인터페이스 V2
- AWS .NET V4용 SDK
- AWS SDK for C++
- AWS Go용 SDK v2

- AWS Java V2용 SDK
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS PHP V3용 SDK
- AWS Python용 SDK
- AWS SDK for Ruby V3

DeleteMountTarget

지정된 탑재 대상을 삭제합니다.

이 작업은 삭제되는 탑재 대상을 사용해서 파일 시스템의 모든 탑재를 강제로 해제하여, 이러한 탑대를 사용하는 인스턴스 또는 애플리케이션이 중단될 수 있습니다. 애플리케이션이 갑자기 중단되는 것을 방지하려면 가능하면 탑재 대상의 모든 탑재를 탑재 해제하는 것을 고려할 수 있습니다. 이 작업을 수행하면 관련 네트워크 인터페이스도 삭제됩니다. 커밋되지 않은 쓰기는 손실될 수 있지만 이 작업을 사용하여 탑재 대상을 해제해도 파일 시스템 자체는 손상되지 않습니다. 생성한 파일 시스템은 그대로 남아 있습니다. 다른 탑재 대상을 사용하여 VPC에 EC2 인스턴스를 탑재할 수 있습니다.

이 작업에는 파일 시스템에서 다음 작업에 대한 권한이 필요합니다.

- `elasticfilesystem:DeleteMountTarget`

> **Note**
>
> 탑재 대상 상태가 아직 `deleting`일 때 `DeleteMountTarget` 직접 호출이 반환됩니다. 해당 파일 시스템에 대한 탑재 대상 설명 목록을 반환하는 DescribeMountTargets 작업을 직접 호출하여 탑재 대상 삭제를 확인할 수 있습니다.

이 작업에는 탑재 대상의 네트워크 인터페이스에서 다음 Amazon EC2 작업에 대한 권한도 필요합니다.

- `ec2:DeleteNetworkInterface`

Request Syntax

```
DELETE /2015-02-01/mount-targets/MountTargetId HTTP/1.1
```

URI 요청 파라미터

요청은 다음 URI 파라미터를 사용합니다.

MountTargetId

삭제할 탑재 대상의 ID입니다(문자열).

길이 제약: 최소 길이는 13자. 최대 길이는 45입니다.

패턴: `^fsmt-[0-9a-f]{8,40}$`

필수 여부: 예

Request Body

해당 요청에는 본문이 없습니다.

Response Syntax

```
HTTP/1.1 204
```

Response Elements

액션이 성공하면 해당 서비스는 빈 HTTP 본문과 함께 HTTP 204 응답을 되돌려줍니다.

Errors

BadRequest

요청이 잘못되었거나 잘못된 파라미터 값 또는 필수 파라미터 누락 등의 오류가 있는 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 400

DependencyTimeout

요청을 처리하는 중 서비스 시간이 초과되었으므로 클라이언트가 호출을 다시 시도해야 합니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 504

InternalServerError

서버 측에서 오류가 발생한 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 500

MountTargetNotFound

호출자의 AWS 계정에 지정된 ID의 탑재 대상이 없는 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야

할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 404

예제

파일 시스템의 탑재 대상 제거

다음 예제는 특정 탑재 대상을 삭제하기 위해 DELETE 요청을 보냅니다.

샘플 요청

```
DELETE /2015-02-01/mount-targets/fsmt-9a13661e HTTP/1.1
Host: elasticfilesystem.us-west-2.amazonaws.com
x-amz-date: 20140622T232908Z
Authorization: <...>
```

샘플 응답

```
HTTP/1.1 204 No Content
x-amzn-RequestId: 01234567-89ab-cdef-0123-456789abcdef
```

참고

언어별 AWS SDKs

- AWS 명령줄 인터페이스 V2
- AWS .NET V4용 SDK
- AWS SDK for C++
- AWS Go용 SDK v2
- AWS Java V2용 SDK
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS PHP V3용 SDK
- AWS Python용 SDK

- AWS SDK for Ruby V3

DeleteReplicationConfiguration

복제 구성을 삭제합니다. 복제 구성을 삭제하면 복제 프로세스가 종료됩니다. 복제 구성을 삭제하면 대상 파일 시스템이 `Writeable` 상태가 되고 그 복제 덮어쓰기 보호가 다시 활성화됩니다. 자세한 내용은 복제 구성 삭제 섹션을 참조하십시오.

이 작업에는 `elasticfilesystem:DeleteReplicationConfiguration` 액션에 대한 권한이 필요합니다.

Request Syntax

```
DELETE /2015-02-01/file-systems/SourceFileSystemId/replication-configuration?
deletionMode=DeletionMode HTTP/1.1
```

URI 요청 파라미터

요청은 다음 URI 파라미터를 사용합니다.

DeletionMode

여러 AWS 계정 또는 여러 간에 복제할 때 AWS 리전 Amazon EFS는 기본적으로 소스 및 대상 계정 또는 리전(`ALL_CONFIGURATIONS`) 모두에서 복제 구성을 삭제합니다. Amazon EFS가 양쪽에서 복제 구성을 삭제하지 못하게 하는 구성 또는 권한 문제가 있는 경우 `LOCAL_CONFIGURATION_ONLY` 모드를 사용하여 로컬 측(삭제가 수행되는 계정 또는 리전)에서만 복제 구성을 삭제할 수 있습니다.

> **Note**
>
> Amazon EFS가 소스 및 대상 계정 또는 리전 모두에서 복제 구성을 삭제할 수 없는 경우에만 `LOCAL_CONFIGURATION_ONLY` 모드를 사용합니다. 로컬 구성을 삭제하면 구성을 다른 계정 또는 리전에서 복구할 수 없습니다.
> 또한 동일한 계정, 동일한 리전 복제에는이 모드를 사용하지 마세요. 이렇게 하면 BadRequest 예외 오류가 발생합니다.

유효한 값: `ALL_CONFIGURATIONS | LOCAL_CONFIGURATION_ONLY`

SourceFileSystemId

복제 구성의 소스 파일 시스템 ID입니다.

길이 제약: 최대 길이는 128입니다.

패턴: `^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$`

필수 여부: 예

Request Body

해당 요청에는 본문이 없습니다.

Response Syntax

```
HTTP/1.1 204
```

Response Elements

액션이 성공하면 해당 서비스는 빈 HTTP 본문과 함께 HTTP 204 응답을 되돌려줍니다.

Errors

BadRequest

요청이 잘못되었거나 잘못된 파라미터 값 또는 필수 파라미터 누락 등의 오류가 있는 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 400

FileSystemNotFound

지정된 `FileSystemId` 값이 요청자의에 없는 경우 반환됩니다 AWS 계정.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 404

InternalServerError

서버 측에서 오류가 발생한 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 500

ReplicationNotFound

지정된 파일 시스템에 복제 구성이 없는 경우 반환됩니다.

ErrorCode

ReplicationNotFound

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 404

참고

언어별 AWS SDKs

- AWS 명령줄 인터페이스 V2
- AWS .NET V4용 SDK
- AWS SDK for C++
- AWS Go용 SDK v2
- AWS Java V2용 SDK
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS PHP V3용 SDK
- AWS Python용 SDK
- AWS SDK for Ruby V3

DeleteTags

> **Note**
>
> 더 이상 사용되지 않음 - DeleteTags는 더 이상 사용되지 않으며 유지 관리되지 않습니다. EFS 리소스에서 태그를 제거하려면 UntagResource API 작업을 사용합니다.

파일 시스템에서 지정된 태그를 삭제합니다. DeleteTags 요청에 존재하지 않는 태그 키가 포함된 경우 Amazon EFS는 해당 태그 키를 무시하고 오류를 발생시키지 않습니다. 태그 및 관련 제한에 대한 자세한 내용은 AWS 결제 및 비용 관리 사용 설명서의 태그 제한을 참조하세요.

이 작업에는 elasticfilesystem:DeleteTags 액션에 대한 권한이 필요합니다.

Request Syntax

```
POST /2015-02-01/delete-tags/FileSystemId HTTP/1.1
Content-type: application/json

{
   "TagKeys": [ "string" ]
}
```

URI 요청 파라미터

요청은 다음 URI 파라미터를 사용합니다.

FileSystemId

태그를 삭제하려는 파일 시스템의 ID입니다(문자열).

길이 제약: 최대 길이는 128입니다.

패턴: ^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$

필수 여부: 예

요청 본문

요청은 JSON 형식으로 다음 데이터를 받습니다.

TagKeys

삭제할 태그 키의 목록입니다.

타입: 문자열 배열

배열 멤버: 최소수는 1개입니다. 최대수 50개.

길이 제약: 최소 길이 1. 최대 길이 128.

패턴: `^(?![aA]{1}[wW]{1}[sS]{1}:)([\p{L}\p{Z}\p{N}_.:/=+\-@]+)$`

필수 여부: 예

응답 구문

```
HTTP/1.1 204
```

Response Elements

액션이 성공하면 해당 서비스는 빈 HTTP 본문과 함께 HTTP 204 응답을 되돌려줍니다.

Errors

BadRequest

요청이 잘못되었거나 잘못된 파라미터 값 또는 필수 파라미터 누락 등의 오류가 있는 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 400

FileSystemNotFound

지정된 `FileSystemId` 값이 요청자의에 없는 경우 반환됩니다 AWS 계정.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 404

InternalServerError

서버 측에서 오류가 발생한 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 500

참고

언어별 AWS SDKs

- AWS 명령줄 인터페이스 V2

- AWS .NET V4용 SDK
- AWS SDK for C++
- AWS Go용 SDK v2
- AWS Java V2용 SDK
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS PHP V3용 SDK
- AWS Python용 SDK
- AWS SDK for Ruby V3

DescribeAccessPoints

AccessPointId가 제공된 경우 특정 Amazon EFS 액세스 포인트에 대한 설명을 반환합니다. EFS FileSystemId를 제공하면 해당 파일 시스템의 모든 액세스 포인트에 대한 설명이 반환됩니다. 요청에서 AccessPointId 또는 FileSystemId를 제공할 수 있지만 둘 다 제공할 수는 없습니다.

이 작업에는 elasticfilesystem:DescribeAccessPoints 액션에 대한 권한이 필요합니다.

Request Syntax

```
GET /2015-02-01/access-points?
AccessPointId=AccessPointId&FileSystemId=FileSystemId&MaxResults=MaxResults&NextToken=NextToken
 HTTP/1.1
```

URI 요청 파라미터

요청은 다음 URI 파라미터를 사용합니다.

AccessPointId

(선택 사항)응답에서 설명할 EFS 액세스 포인트를 지정합니다. FileSystemId와는 상호 배타적입니다.

길이 제약: 최대 길이는 128입니다.

패턴: ^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:access-point/fsap-[0-9a-f]{8,40}|fsap-[0-9a-f]{8,40})$

FileSystemId

(선택 사항)FileSystemId를 제공하는 경우 EFS는 해당 파일 시스템에 대한 모든 액세스 포인트를 반환하며, AccessPointId와는 상호 배타적입니다.

길이 제약: 최대 길이는 128입니다.

패턴: ^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$

MaxResults

(선택 사항)파일 시스템의 모든 액세스 포인트를 검색할 때, 선택적으로 MaxItems 파라미터를 지정하여 응답으로 반환되는 객체 수를 제한할 수 있습니다. 기본값은 100입니다.

유효 범위: 최소값 1.

NextToken

응답에 페이지가 매겨진 경우 NextToken이 나타납니다. 후속 요청에서 NextMarker를 사용하여 액세스 포인트 설명의 다음 페이지를 가져올 수 있습니다.

길이 제약: 최소 길이 1. 최대 길이 128.

패턴: .+

요청 본문

해당 요청에는 본문이 없습니다.

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
   "AccessPoints": [
      {
         "AccessPointArn": "string",
         "AccessPointId": "string",
         "ClientToken": "string",
         "FileSystemId": "string",
         "LifeCycleState": "string",
         "Name": "string",
         "OwnerId": "string",
         "PosixUser": {
            "Gid": number,
            "SecondaryGids": [ number ],
            "Uid": number
         },
         "RootDirectory": {
            "CreationInfo": {
               "OwnerGid": number,
               "OwnerUid": number,
               "Permissions": "string"
            },
            "Path": "string"
         },
```

```
         "Tags": [
            {
               "Key": "string",
               "Value": "string"
            }
         ]
      }
   ],
   "NextToken": "string"
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반송합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

AccessPoints

액세스 포인트 설명 배열입니다.

타입: AccessPointDescription객체 배열

NextToken

응답에 반환된 것보다 많은 액세스 포인트가 있는 경우 제시하세요. 후속 요청에서 NextMarker를 사용하여 추가 설명을 가져올 수 있습니다.

유형: 문자열

길이 제약: 최소 길이는 1. 최대 길이 128.

패턴: `.+`

오류

AccessPointNotFound

지정된 `AccessPointId` 값이 요청자의에 없는 경우 반환됩니다 AWS 계정.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 404

BadRequest

요청이 잘못되었거나 잘못된 파라미터 값 또는 필수 파라미터 누락 등의 오류가 있는 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 400

FileSystemNotFound

지정된 `FileSystemId` 값이 요청자의에 없는 경우 반환됩니다 AWS 계정.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인

오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 404

InternalServerError

서버 측에서 오류가 발생한 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 500

참고

언어별 AWS SDKs

- AWS 명령줄 인터페이스 V2
- AWS .NET V4용 SDK
- AWS SDK for C++
- AWS Go용 SDK v2
- AWS Java V2용 SDK
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS PHP V3용 SDK
- AWS Python용 SDK
- AWS SDK for Ruby V3

DescribeAccountPreferences

현재에서 요청을 하는 사용자와 AWS 계정 연결된에 대한 계정 기본 설정 설정을 반환합니다 AWS 리전.

Request Syntax

```
GET /2015-02-01/account-preferences HTTP/1.1
Content-type: application/json

{
   "MaxResults": number,
   "NextToken": "string"
}
```

URI 요청 파라미터

요청은 URI 파라미터를 사용하지 않습니다.

요청 본문

요청은 JSON 형식으로 다음 데이터를 받습니다.

MaxResults

(선택 사항)계정 설정을 가져올 때 선택적으로 MaxItems 파라미터를 지정하여 응답으로 반환되는 객체 수를 제한할 수 있습니다. 기본 값은 100입니다.

타입: 정수

유효 범위: 최소값 1.

필수 여부: 아니요

NextToken

(선택 사항)응답 페이로드가 페이지로 구분된 경우 후속 요청에서 NextToken을 사용하여 AWS 계정 설정의 다음 페이지를 가져올 수 있습니다.

유형: 문자열

길이 제약: 최소 길이는 1. 최대 길이 128.

패턴: .+

필수 여부: 아니요

응답 구문

```
HTTP/1.1 200
Content-type: application/json

{
   "NextToken": "string",
   "ResourceIdPreference": {
      "ResourceIdType": "string",
      "Resources": [ "string" ]
   }
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반송합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

NextToken

응답으로 반환된 것보다 많은 레코드가 있는 경우 제시하세요. 후속 요청에서 NextToken을 사용하여 추가 설명을 가져올 수 있습니다.

유형: 문자열

길이 제약: 최소 길이는 1. 최대 길이 128.

패턴: .+

ResourceIdPreference

현재에서 요청을 하는 사용자와 AWS 계정 연결된에 대한 리소스 ID 기본 설정을 설명합니다 AWS 리전.

타입: ResourceIdPreference 객체

오류

InternalServerError

서버 측에서 오류가 발생한 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 500

참고

언어별 AWS SDKs

- AWS 명령줄 인터페이스 V2
- AWS .NET V4용 SDK
- AWS SDK for C++
- AWS Go용 SDK v2
- AWS Java V2용 SDK
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS PHP V3용 SDK
- AWS Python용 SDK
- AWS SDK for Ruby V3

DescribeBackupPolicy

지정된 EFS 파일 시스템의 백업 정책을 반환합니다.

Request Syntax

```
GET /2015-02-01/file-systems/FileSystemId/backup-policy HTTP/1.1
```

URI 요청 파라미터

요청은 다음 URI 파라미터를 사용합니다.

FileSystemId

BackupPolicy를 검색할 EFS 파일 시스템을 지정합니다.

길이 제약: 최대 길이는 128입니다.

패턴: `^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$`

필수 여부: 예

Request Body

해당 요청에는 본문이 없습니다.

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
   "BackupPolicy": {
      "Status": "string"
   }
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반송합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

BackupPolicy

자동 백업의 켜짐 또는 꺼짐을 나타내는 파일 시스템의 백업 정책을 설명합니다.

타입: BackupPolicy 객체

오류

BadRequest

요청이 잘못되었거나 잘못된 파라미터 값 또는 필수 파라미터 누락 등의 오류가 있는 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 400

FileSystemNotFound

지정된 `FileSystemId` 값이 요청자의에 없는 경우 반환됩니다 AWS 계정.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인

오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 404

InternalServerError

서버 측에서 오류가 발생한 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 500

PolicyNotFound

지정된 IAM 정책 또는 리소스 정책을 찾거나 액세스할 수 없는 경우 반환됩니다. 수행 중인 작업에 따라 EFS 파일 시스템에 기본 파일 시스템 정책이 적용되거나 백업 정책이 지정되지 않은 경우에도 오류가 반환될 수 있습니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 404

ValidationException

요청이 이루어진 AWS 리전 에서 AWS Backup 서비스를 사용할 수 없는 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 400

참고

언어별 AWS SDKs

- AWS 명령줄 인터페이스 V2
- AWS .NET V4용 SDK
- AWS SDK for C++
- AWS Go용 SDK v2
- AWS Java V2용 SDK
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS PHP V3용 SDK
- AWS Python용 SDK
- AWS SDK for Ruby V3

DescribeFileSystemPolicy

지정된 EFS 파일 시스템에 대한 FileSystemPolicy를 반환합니다.

이 작업에는 elasticfilesystem:DescribeFileSystemPolicy 액션에 대한 권한이 필요합니다.

Request Syntax

```
GET /2015-02-01/file-systems/FileSystemId/policy HTTP/1.1
```

URI 요청 파라미터

요청은 다음 URI 파라미터를 사용합니다.

FileSystemId

FileSystemPolicy를 검색할 EFS 파일 시스템을 지정합니다.

길이 제약: 최대 길이는 128입니다.

패턴: ^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$

필수 여부: 예

Request Body

해당 요청에는 본문이 없습니다.

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
   "FileSystemId": "string",
   "Policy": "string"
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반송합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

FileSystemId

`FileSystemPolicy`가 적용되는 EFS 파일 시스템을 지정합니다.

유형: 문자열

길이 제약: 최대 길이는 128입니다.

패턴: `^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$`

Policy

EFS 파일 시스템을 위한 JSON 형식의 `FileSystemPolicy`입니다.

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이는 20000입니다.

패턴: `[\s\S]+`

오류

BadRequest

요청이 잘못되었거나 잘못된 파라미터 값 또는 필수 파라미터 누락 등의 오류가 있는 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야

할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 400

FileSystemNotFound

지정된 `FileSystemId` 값이 요청자의에 없는 경우 반환됩니다 AWS 계정.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 404

InternalServerError

서버 측에서 오류가 발생한 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 500

PolicyNotFound

지정된 IAM 정책 또는 리소스 정책을 찾거나 액세스할 수 없는 경우 반환됩니다. 수행 중인 작업에 따라 EFS 파일 시스템에 기본 파일 시스템 정책이 적용되거나 백업 정책이 지정되지 않은 경우에도 오류가 반환될 수 있습니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 404

예제

예

이 예에서는 DescribeFileSystemPolicy의 한 가지 사용법을 보여줍니다.

샘플 요청

```
GET /2015-02-01/file-systems/fs-01234567/policy HTTP/1.1
```

샘플 응답

```
{
    "FileSystemId": "fs-01234567",
    "Policy": "{
        "Version": "2012-10-17",
        "Id": "efs-policy-wizard-cdef0123-aaaa-6666-5555-444455556666",
        "Statement": [
            {
                "Sid": "efs-statement-abcdef01-1111-bbbb-2222-111122224444",
```

```
                "Effect" : "Deny",
                "Principal": {
                "AWS": "*"
                },
                "Action": "*",
                "Resource": "arn:aws:elasticfilesystem:us-east-2:111122223333:file-
system/fs-01234567",
                "Condition": {
                "Bool": {
                    "aws:SecureTransport": "false"
                    }
                }
            },
            {
                "Sid": "efs-statement-01234567-aaaa-3333-4444-111122223333",
                "Effect": "Allow",
                "Principal": {
                    "AWS": "*"
                },
                "Action": [
                    "elasticfilesystem:ClientMount",
                    "elasticfilesystem:ClientWrite"
                ],
                "Resource" : "arn:aws:elasticfilesystem:us-east-2:111122223333:file-
system/fs-01234567"
            }
        ]
    }
}
```

참고

언어별 AWS SDKs

- AWS 명령줄 인터페이스 V2
- AWS .NET V4용 SDK
- AWS SDK for C++
- AWS Go용 SDK v2
- AWS Java V2용 SDK
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin

- AWS PHP V3용 SDK
- AWS Python용 SDK
- AWS SDK for Ruby V3

DescribeFileSystems

파일 시스템 CreationToken 또는 FileSystemId가 제공된 경우 특정 Amazon EFS 파일 시스템에 대한 설명을 반환합니다. 그렇지 않으면 호출하려는 엔드포인트 AWS 리전 의 AWS 계정 에서 호출자의가 소유한 모든 파일 시스템에 대한 설명이 반환됩니다.

모든 파일 시스템 설명을 검색할 때 선택적으로 MaxItems 파라미터를 지정하여 응답의 설명 수를 제한할 수 있습니다. 이 숫자는 자동으로 100으로 설정됩니다. 파일 시스템 설명이 더 남아 있는 경우 Amazon EFS는 응답으로 불투명 토큰인 NextMarker를 반환합니다. 이 경우 Marker 요청 파라미터를 NextMarker 값으로 설정하여 후속 요청을 보내야 합니다.

파일 시스템 설명 목록을 검색하기 위해 이 작업을 반복 프로세스에서 사용합니다. 반복 프로세스에서는 Marker를 사용하지 않고 DescribeFileSystems를 먼저 직접 호출한 다음 Marker 파라미터를 이전 응답의 NextMarker 값으로 설정한 다음 응답에 NextMarker가 없을 때까지 계속 직접 호출합니다.

한 번의 DescribeFileSystems 직접 호출에 대한 응답으로 반환되는 파일 시스템의 순서와 다중 직접 호출 반복의 응답에서 반환되는 파일 시스템의 순서는 지정되지 않았습니다.

이 작업에는 elasticfilesystem:DescribeFileSystems 액션에 대한 권한이 필요합니다.

Request Syntax

```
GET /2015-02-01/file-systems?
CreationToken=CreationToken&FileSystemId=FileSystemId&Marker=Marker&MaxItems=MaxItems
 HTTP/1.1
```

URI 요청 파라미터

요청은 다음 URI 파라미터를 사용합니다.

CreationToken

(선택 사항)이 생성 토큰(문자열)이 있는 파일 시스템으로 목록을 제한합니다. Amazon EFS 파일 시스템 생성시 생성 토큰을 지정합니다.

길이 제한: 최소 길이는 1. 최대 길이는 64.

패턴: .+

FileSystemId

(선택 사항)설명을 검색하려는 파일 시스템의 ID입니다(문자열).

길이 제약: 최대 길이는 128입니다.

패턴: `^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$`

Marker

(선택 사항)이전 DescribeFileSystems 작업에서 반환된 불투명한 페이지 매김 토큰입니다(문자열). 존재하는 경우, 반환 직접 호출이 중단된 부분부터 목록을 계속하도록 지정합니다.

길이 제약: 최소 길이 1. 최대 길이 128.

패턴: `.+`

MaxItems

(선택 사항)응답에서 반환할 최대 파일 시스템 수를 지정합니다(정수). 이 숫자는 자동으로 100으로 설정됩니다. 파일 시스템이 100개 이상인 경우 응답은 페이지당 100페이지로 분류됩니다.

유효 범위: 최소값 1.

Request Body

해당 요청에는 본문이 없습니다.

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
   "FileSystems": [
      {
         "AvailabilityZoneId": "string",
         "AvailabilityZoneName": "string",
         "CreationTime": number,
         "CreationToken": "string",
         "Encrypted": boolean,
         "FileSystemArn": "string",
         "FileSystemId": "string",
         "FileSystemProtection": {
            "ReplicationOverwriteProtection": "string"
         },
```

```
         "KmsKeyId": "string",
         "LifeCycleState": "string",
         "Name": "string",
         "NumberOfMountTargets": number,
         "OwnerId": "string",
         "PerformanceMode": "string",
         "ProvisionedThroughputInMibps": number,
         "SizeInBytes": {
            "Timestamp": number,
            "Value": number,
            "ValueInArchive": number,
            "ValueInIA": number,
            "ValueInStandard": number
         },
         "Tags": [
            {
               "Key": "string",
               "Value": "string"
            }
         ],
         "ThroughputMode": "string"
      }
   ],
   "Marker": "string",
   "NextMarker": "string"
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반송합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

FileSystems

파일 시스템 설명 배열입니다.

타입: FileSystemDescription객체 배열

Marker

요청에서 호출자가 제공한 경우 제시하세요(문자열).

유형: 문자열

길이 제약: 최소 길이는 1. 최대 길이 128.

패턴: `.+`

NextMarker

응답에 반환된 것보다 많은 파일 시스템이 있는 경우 표시됩니다(문자열). 후속 요청에서 `NextMarker`를 사용하여 설명을 가져올 수 있습니다.

유형: 문자열

길이 제약: 최소 길이는 1. 최대 길이 128.

패턴: `.+`

오류

BadRequest

요청이 잘못되었거나 잘못된 파라미터 값 또는 필수 파라미터 누락 등의 오류가 있는 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 400

FileSystemNotFound

지정된 `FileSystemId` 값이 요청자의에 없는 경우 반환됩니다 AWS 계정.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 404

InternalServerError

서버 측에서 오류가 발생한 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 500

예제

10개 파일 시스템 목록을 검색합니다.

다음 예제에서는 `file-systems` 엔드포인트(`elasticfilesystem.us-west-2.amazonaws.com/2015-02-01/file-systems`)에 GET 요청을 보냅니다. 요청은 파일 시스템 설명 수를 10개로 제한하는 `MaxItems` 쿼리 파라미터를 지정합니다.

샘플 요청

```
GET /2015-02-01/file-systems?MaxItems=10 HTTP/1.1
Host: elasticfilesystem.us-west-2.amazonaws.com
x-amz-date: 20140622T191208Z
```

```
Authorization: <...>
```

샘플 응답

```
HTTP/1.1 200 OK
x-amzn-RequestId: 01234567-89ab-cdef-0123-456789abcdef
Content-Type: application/json
Content-Length: 499
{
   "FileSystems":[
      {
         "OwnerId":"251839141158",
         "CreationToken":"MyFileSystem1",
         "FileSystemId":"fs-01234567",
         "PerformanceMode" : "generalPurpose",
         "CreationTime":"1403301078",
         "LifeCycleState":"created",
         "Name":"my first file system",
         "NumberOfMountTargets":1,
         "SizeInBytes":{
            "Timestamp": 1403301078,
            "Value": 29313618372,
            "ValueInArchive": 201156,
            "ValueInIA": 675432,
            "ValueInStandard": 29312741784
         }
      }
   ]
}
```

참고

언어별 AWS SDKs

- AWS 명령줄 인터페이스 V2
- AWS .NET V4용 SDK
- AWS SDK for C++
- AWS Go용 SDK v2
- AWS Java V2용 SDK
- AWS SDK for JavaScript V3

- AWS SDK for Kotlin
- AWS PHP V3용 SDK
- AWS Python용 SDK
- AWS SDK for Ruby V3

DescribeLifecycleConfiguration

지정된 EFS 파일 시스템에 대한 현재 `LifecycleConfiguration` 객체를 반환합니다. Lifecycle Management는 `LifecycleConfiguration` 객체를 사용하여 스토리지 클래스 간에 파일을 이동할 시기를 식별합니다. `LifecycleConfiguration` 객체가 없는 파일 시스템의 경우 호출은 응답에 빈 배열을 반환합니다.

이 작업에는 `elasticfilesystem:DescribeLifecycleConfiguration` 작업에 대한 권한이 필요합니다.

Request Syntax

```
GET /2015-02-01/file-systems/FileSystemId/lifecycle-configuration HTTP/1.1
```

URI 요청 파라미터

요청은 다음 URI 파라미터를 사용합니다.

FileSystemId

`LifecycleConfiguration` 객체를 검색하려는 파일 시스템의 ID입니다(문자열).

길이 제약: 최대 길이는 128입니다.

패턴: `^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$`

필수 여부: 예

Request Body

해당 요청에는 본문이 없습니다.

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
   "LifecyclePolicies": [
```

```
      {
         "TransitionToArchive": "string",
         "TransitionToIA": "string",
         "TransitionToPrimaryStorageClass": "string"
      }
   ]
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반송합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

LifecyclePolicies

수명 주기 관리 정책들의 배열. EFS는 파일 시스템당 최대 하나의 정책을 지원합니다.

타입: LifecyclePolicy객체 배열

배열 멤버: 최대 항목 수는 3개입니다.

오류

BadRequest

요청이 잘못되었거나 잘못된 파라미터 값 또는 필수 파라미터 누락 등의 오류가 있는 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 400

FileSystemNotFound

지정된 `FileSystemId` 값이 요청자의에 없는 경우 반환됩니다 AWS 계정.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 404

InternalServerError

서버 측에서 오류가 발생한 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 500

예제

파일 시스템의 수명 주기 구성을 검색합니다.

다음 요청은 지정된 파일 시스템의 `LifecycleConfiguration` 객체를 검색합니다.

샘플 요청

```
GET /2015-02-01/file-systems/fs-01234567/lifecycle-configuration HTTP/1.1
Host: elasticfilesystem.us-west-2.amazonaws.com
x-amz-date: 20181120T221118Z
Authorization: <...>
```

샘플 응답

```
HTTP/1.1 200 OK
        x-amzn-RequestId: 01234567-89ab-cdef-0123-456789abcdef
        Content-Type: application/json
        Content-Length: 86
{
  "LifecyclePolicies": [
    {
        "TransitionToArchive": "AFTER_270_DAYS"
    },
    {
        "TransitionToIA": "AFTER_14_DAYS"
    },
    {
        "TransitionToPrimaryStorageClass": "AFTER_1_ACCESS"
    }
  ]
}
```

참고

언어별 AWS SDKs

- AWS 명령줄 인터페이스 V2
- AWS .NET V4용 SDK
- AWS SDK for C++
- AWS Go용 SDK v2
- AWS Java V2용 SDK
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin

- AWS PHP V3용 SDK
- AWS Python용 SDK
- AWS SDK for Ruby V3

DescribeMountTargets

파일 시스템에 대한 모든 현재 탑재 대상 또는 특정 탑재 대상의 설명을 반환합니다. 현재 탑재 대상을 모두 요청하는 경우 응답에서 반환되는 탑재 대상의 순서는 지정되지 않습니다.

이 작업을 수행하려면 `FileSystemId`에서 지정한 파일 시스템 ID 또는 `MountTargetId`에서 지정한 탑재 대상의 파일 시스템에 대한 `elasticfilesystem:DescribeMountTargets` 작업 권한이 필요합니다.

해당 CLI 명령은 `describe-mount-targets`입니다.

Request Syntax

```
GET /2015-02-01/mount-targets?
AccessPointId=AccessPointId&FileSystemId=FileSystemId&Marker=Marker&MaxItems=MaxItems&MountTarg
 HTTP/1.1
```

URI 요청 파라미터

요청은 다음 URI 파라미터를 사용합니다.

AccessPointId

(선택 사항)탑재 대상을 나열하려는 액세스 포인트의 ID입니다. `FileSystemId` 또는 `MountTargetId`가 요청에 포함되지 않은 경우 요청에 포함되어야 합니다. 액세스 포인트 ID 또는 ARN을 입력으로 받아들입니다.

길이 제약: 최대 길이는 128입니다.

패턴: `^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:access-point/fsap-[0-9a-f]{8,40}|fsap-[0-9a-f]{8,40})$`

FileSystemId

(선택 사항)탑재 대상을 나열하려는 파일 시스템의 ID (문자열). `AccessPointId` 또는 `MountTargetId`가 포함되지 않은 경우 요청에 포함되어야 합니다. 파일 시스템 ID 또는 ARN을 입력으로 받아들입니다.

길이 제약: 최대 길이는 128입니다.

패턴: `^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$`

Marker

(선택 사항)이전 DescribeMountTargets 작업에서 반환된 불투명한 페이지 매김 토큰입니다(문자열). 존재하는 경우, 반환 직접 호출이 중단된 부분부터 목록을 계속하도록 지정합니다.

길이 제약: 최소 길이 1. 최대 길이 128.

패턴: .+

MaxItems

(선택 사항)응답에 반환될 최대 탑재 대상 수입니다. 현재 이 수는 자동으로 10으로 설정되며 다른 값은 무시됩니다. 탑재 대상이 100개가 넘는 경우 응답은 페이지당 100페이지로 분류됩니다.

유효 범위: 최소값 1.

MountTargetId

(선택 사항)설명하려는 탑재 대상의 ID입니다(문자열). FileSystemId가 포함되지 않은 경우 요청에 포함해야 합니다. 탑재 대상 ID 또는 ARN을 입력으로 받아들입니다.

길이 제약: 최소 길이는 13자. 최대 길이는 45입니다.

패턴: ^fsmt-[0-9a-f]{8,40}$

요청 본문

해당 요청에는 본문이 없습니다.

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
   "Marker": "string",
   "MountTargets": [
      {
         "AvailabilityZoneId": "string",
         "AvailabilityZoneName": "string",
         "FileSystemId": "string",
         "IpAddress": "string",
```

```
            "Ipv6Address": "string",
            "LifeCycleState": "string",
            "MountTargetId": "string",
            "NetworkInterfaceId": "string",
            "OwnerId": "string",
            "SubnetId": "string",
            "VpcId": "string"
        }
    ],
    "NextMarker": "string"
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반송합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

Marker

요청에 `Marker`가 포함된 경우 응답은 이 필드에 해당 값을 반환합니다.

유형: 문자열

길이 제약: 최소 길이는 1. 최대 길이 128.

패턴: `.+`

MountTargets

파일 시스템의 탑재 대상을 `MountTargetDescription` 객체의 배열로 반환합니다.

타입: MountTargetDescription객체 배열

NextMarker

값이 있는 경우 반환할 탑재 대상이 더 많습니다. 후속 요청 시 요청에 `Marker`를 제공하여 다음 탑재 대상 세트를 검색할 수 있습니다.

유형: 문자열

길이 제약: 최소 길이는 1. 최대 길이 128.

패턴: `.+`

오류

AccessPointNotFound

지정된 `AccessPointId` 값이 요청자의에 없는 경우 반환됩니다 AWS 계정.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 404

BadRequest

요청이 잘못되었거나 잘못된 파라미터 값 또는 필수 파라미터 누락 등의 오류가 있는 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 400

FileSystemNotFound

지정된 `FileSystemId` 값이 요청자의에 없는 경우 반환됩니다 AWS 계정.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 404

InternalServerError

서버 측에서 오류가 발생한 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 500

MountTargetNotFound

호출자의 AWS 계정에 지정된 ID의 탑재 대상이 없는 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야

할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 404

예제

파일 시스템용으로 생성된 탑재 대상에 대한 설명을 검색합니다.

다음 요청은 지정된 파일 시스템에 대해 생성된 탑재 대상에 대한 설명을 검색합니다.

샘플 요청

```
GET /2015-02-01/mount-targets?FileSystemId=fs-01234567 HTTP/1.1
Host: elasticfilesystem.us-west-2.amazonaws.com
x-amz-date: 20140622T191252Z
Authorization: <...>
```

샘플 응답

```
HTTP/1.1 200 OK
x-amzn-RequestId: 01234567-89ab-cdef-0123-456789abcdef
Content-Type: application/json
Content-Length: 357

{
   "MountTargets":[
      {
         "OwnerId":"251839141158",
         "MountTargetId":"fsmt-01234567",
         "FileSystemId":"fs-01234567",
         "SubnetId":"subnet-01234567",
         "LifeCycleState":"added",
         "IpAddress":"10.0.2.42",
         "NetworkInterfaceId":"eni-1bcb7772"
      }
   ]
}
```

참고

언어별 AWS SDKs

- AWS 명령줄 인터페이스 V2
- AWS .NET V4용 SDK
- AWS SDK for C++
- AWS Go용 SDK v2
- AWS Java V2용 SDK
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS PHP V3용 SDK
- AWS Python용 SDK
- AWS SDK for Ruby V3

DescribeMountTargetSecurityGroups

현재 탑재 대상에 대해 유효한 보안 그룹을 반환합니다. 이 작업을 수행하려면 탑재 대상의 네트워크 인터페이스가 생성되고 탑재 대상의 수명 주기 상태는 `deleted`가 아니어야 합니다.

이 작업에는 다음 작업에 대한 권한이 필요합니다.

- 탑재 대상의 파일 시스템에서의 `elasticfilesystem:DescribeMountTargetSecurityGroups` 작업
- 탑재 대상의 네트워크 인터페이스에서의 `ec2:DescribeNetworkInterfaceAttribute` 작업

Request Syntax

```
GET /2015-02-01/mount-targets/MountTargetId/security-groups HTTP/1.1
```

URI 요청 파라미터

요청은 다음 URI 파라미터를 사용합니다.

MountTargetId

보안 그룹을 검색할 마운트 대상의 ID입니다.

길이 제약: 최소 길이는 13자. 최대 길이는 45입니다.

패턴: `^fsmt-[0-9a-f]{8,40}$`

필수 여부: 예

Request Body

해당 요청에는 본문이 없습니다.

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
```

```
   "SecurityGroups": [ "string" ]
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반송합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

SecurityGroups

보안 그룹의 배열입니다.

유형: 문자열 어레이

배열 멤버: 최대 항목 수는 100개입니다.

길이 제약: 최소 길이는 11입니다. 최대 길이는 43입니다.

패턴: `^sg-[0-9a-f]{8,40}`

오류

BadRequest

요청이 잘못되었거나 잘못된 파라미터 값 또는 필수 파라미터 누락 등의 오류가 있는 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 400

IncorrectMountTargetState

탑재 대상이 작업에 적합한 올바른 상태가 아닌 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 409

InternalServerError

서버 측에서 오류가 발생한 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 500

MountTargetNotFound

호출자의 AWS 계정에 지정된 ID의 탑재 대상이 없는 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 404

예제

파일 시스템에 적용되는 보안 그룹을 검색합니다.

다음 예에서는 탑재 대상과 관련된 네트워크 인터페이스에 적용되는 보안 그룹을 검색합니다.

샘플 요청

```
GET /2015-02-01/mount-targets/fsmt-9a13661e/security-groups HTTP/1.1
Host: elasticfilesystem.us-west-2.amazonaws.com
x-amz-date: 20140620T223513Z
Authorization: <...>
```

샘플 응답

```
HTTP/1.1 200 OK
x-amzn-RequestId: 01234567-89ab-cdef-0123-456789abcdef
Content-Length: 57

{
"SecurityGroups" : [
"sg-188d9f74"
]
}
```

참고

언어별 AWS SDKs

- AWS 명령줄 인터페이스 V2

- AWS .NET V4용 SDK
- AWS SDK for C++
- AWS Go용 SDK v2
- AWS Java V2용 SDK
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS PHP V3용 SDK
- AWS Python용 SDK
- AWS SDK for Ruby V3

DescribeReplicationConfigurations

특정 파일 시스템의 복제 구성을 검색합니다. 파일 시스템을 지정하지 않으면의 AWS 계정 에 대한 모든 복제 구성이 검색 AWS 리전 됩니다.

Request Syntax

```
GET /2015-02-01/file-systems/replication-configurations?
FileSystemId=FileSystemId&MaxResults=MaxResults&NextToken=NextToken HTTP/1.1
```

URI 요청 파라미터

요청은 다음 URI 파라미터를 사용합니다.

FileSystemId

파일 시스템 ID를 제공하여 특정 파일 시스템의 복제 구성을 검색할 수 있습니다. 교차 계정, 교차 리전 복제의 경우 계정은 자체 리전의 파일 시스템에 대한 복제 구성만 설명할 수 있습니다.

길이 제약: 최대 길이는 128입니다.

패턴: `^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$`

MaxResults

(선택 사항)응답에서 반환되는 객체 수를 제한하려면 `MaxItems` 파라미터를 지정할 수 있습니다. 기본값은 100입니다.

유효 범위: 최소값 1.

NextToken

응답에 페이지가 매겨진 경우 `NextToken`이 나타납니다. 후속 요청에서 다음 출력 페이지를 가져오는 데 `NextToken`을 사용할 수 있습니다.

길이 제약: 최소 길이 1. 최대 길이 128.

패턴: `.+`

요청 본문

해당 요청에는 본문이 없습니다.

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
   "NextToken": "string",
   "Replications": [
      {
         "CreationTime": number,
         "Destinations": [
            {
               "FileSystemId": "string",
               "LastReplicatedTimestamp": number,
               "OwnerId": "string",
               "Region": "string",
               "RoleArn": "string",
               "Status": "string",
               "StatusMessage": "string"
            }
         ],
         "OriginalSourceFileSystemArn": "string",
         "SourceFileSystemArn": "string",
         "SourceFileSystemId": "string",
         "SourceFileSystemOwnerId": "string",
         "SourceFileSystemRegion": "string"
      }
   ]
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반송합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

NextToken

이전 응답의 NextToken를 후속 요청에서 사용하여 추가 설명을 가져올 수 있습니다.

유형: 문자열

길이 제약: 최소 길이는 1. 최대 길이 128.

패턴: `.+`

Replications

반환된 복제 구성 컬렉션입니다.

타입: ReplicationConfigurationDescription 객체 배열

오류

BadRequest

요청이 잘못되었거나 잘못된 파라미터 값 또는 필수 파라미터 누락 등의 오류가 있는 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 400

FileSystemNotFound

지정된 `FileSystemId` 값이 요청자의에 없는 경우 반환됩니다 AWS 계정.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인

오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 404

InternalServerError

서버 측에서 오류가 발생한 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 500

ReplicationNotFound

지정된 파일 시스템에 복제 구성이 없는 경우 반환됩니다.

ErrorCode

ReplicationNotFound

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 404

ValidationException

요청이 이루어진 AWS 리전 에서 AWS Backup 서비스를 사용할 수 없는 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 400

참고

언어별 AWS SDKs

- AWS 명령줄 인터페이스 V2
- AWS .NET V4용 SDK
- AWS SDK for C++
- AWS Go용 SDK v2
- AWS Java V2용 SDK
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS PHP V3용 SDK
- AWS Python용 SDK
- AWS SDK for Ruby V3

DescribeTags

> **Note**
>
> 더 이상 사용되지 않음 - 이 DescribeTags 작업은 더 이상 사용되지 않으며 유지 관리되지 않습니다. EFS 리소스와 연결된 태그를 보려면 ListTagsForResource API 작업을 사용하세요.

파일 시스템과 연결된 태그를 반환합니다. 한 번의 DescribeTags 직접 호출에 대한 응답에서 반환되는 태그의 순서와 다중 직접 호출 반복(페이지 매김을 사용하는 경우)의 응답에서 반환되는 태그의 순서는 지정되지 않습니다.

이 작업에는 elasticfilesystem:DescribeTags 액션에 대한 권한이 필요합니다.

Request Syntax

```
GET /2015-02-01/tags/FileSystemId/?Marker=Marker&MaxItems=MaxItems HTTP/1.1
```

URI 요청 파라미터

요청은 다음 URI 파라미터를 사용합니다.

FileSystemId

태그 세트를 가져올 파일 시스템의 ID입니다.

길이 제약: 최대 길이는 128입니다.

패턴: `^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$`

필수 여부: 예

Marker

(선택 사항)이전 DescribeTags 작업에서 반환된 불투명한 페이지 매김 토큰(문자열). 존재하는 경우, 이전 직접 호출이 중단된 부분부터 목록을 계속하도록 지정합니다.

길이 제약: 최소 길이 1. 최대 길이 128.

패턴: `.+`

MaxItems

(선택 사항)응답에 반환될 최대 파일 시스템 태그 수입니다. 현재 이 숫자는 자동으로 100으로 설정되며 다른 값은 무시됩니다. 태그가 100개가 넘는 경우 응답은 페이지당 100페이지로 분류됩니다.

유효 범위: 최소값 1.

Request Body

해당 요청에는 본문이 없습니다.

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
   "Marker": "string",
   "NextMarker": "string",
   "Tags": [
      {
         "Key": "string",
         "Value": "string"
      }
   ]
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반송합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

Marker

요청에 `Marker`가 포함된 경우 응답은 이 필드에 해당 값을 반환합니다.

유형: 문자열

길이 제약: 최소 길이는 1. 최대 길이 128.

패턴: `.+`

NextMarker

값이 있는 경우 반환할 태그가 더 많습니다. 후속 요청에서 다음 태그 세트를 검색하기 위해 다음 요청의 `Marker` 파라미터의 값으로 `NextMarker`의 값을 제공할 수 있습니다.

유형: 문자열

길이 제약: 최소 길이는 1. 최대 길이 128.

패턴: `.+`

Tags

파일 시스템과 연결된 태그를 `Tag` 객체의 배열로 반환합니다.

타입: Tag 객체 배열

오류

BadRequest

요청이 잘못되었거나 잘못된 파라미터 값 또는 필수 파라미터 누락 등의 오류가 있는 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 400

FileSystemNotFound

지정된 `FileSystemId` 값이 요청자의에 없는 경우 반환됩니다 AWS 계정.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 404

InternalServerError

서버 측에서 오류가 발생한 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 500

예제

파일 시스템과 연결된 태그 검색

다음 요청은 지정된 파일 시스템과 연결된 태그(키-값 페어)를 검색합니다.

샘플 요청

```
GET /2015-02-01/tags/fs-01234567/ HTTP/1.1
```

```
Host: elasticfilesystem.us-west-2.amazonaws.com
x-amz-date: 20140620T215404Z
Authorization: <...>
```

샘플 응답

```
HTTP/1.1 200 OK
x-amzn-RequestId: 01234567-89ab-cdef-0123-456789abcdef
Content-Type: application/json
Content-Length: 288

{
    "Tags":[
        {
            "Key":"Name",
            "Value":"my first file system"
        },
        {
            "Key":"Fleet",
            "Value":"Development"
        },
        {
            "Key":"Developer",
            "Value":"Alice"
        }
    ]
}
```

참고

언어별 AWS SDKs

- AWS 명령줄 인터페이스 V2
- AWS .NET V4용 SDK
- AWS SDK for C++
- AWS Go용 SDK v2
- AWS Java V2용 SDK
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS PHP V3용 SDK

- AWS Python용 SDK
- AWS SDK for Ruby V3

ListTagsForResource

최상위 EFS 리소스에 대한 모든 태그를 나열합니다. 태그를 검색하려는 리소스에 대한 ID를 입력해야 합니다.

이 작업에는 `elasticfilesystem:DescribeAccessPoints` 액션에 대한 권한이 필요합니다.

Request Syntax

```
GET /2015-02-01/resource-tags/ResourceId?MaxResults=MaxResults&NextToken=NextToken
 HTTP/1.1
```

URI 요청 파라미터

요청은 다음 URI 파라미터를 사용합니다.

MaxResults

(선택 사항)응답에 반환될 최대 태그 객체 수를 지정합니다. 기본값은 100입니다.

유효 범위: 최소값 1.

NextToken

(선택 사항)응답 페이로드가 페이지로 구분된 경우 후속 요청에서 액세스 포인트 설명의 다음 페이지를 가져오는 데 `NextToken`을 사용할 수 있습니다.

길이 제약: 최소 길이 1. 최대 길이 128.

패턴: `.+`

ResourceId

태그를 검색할 EFS 리소스를 지정합니다. 이 API 엔드포인트를 사용하여 EFS 파일 시스템 및 액세스 포인트에 대한 태그를 검색할 수 있습니다.

길이 제약: 최대 길이는 128입니다.

패턴: `^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:(access-point/fsap|file-system/fs)-[0-9a-f]{8,40}|fs(ap)?-[0-9a-f]{8,40})$`

필수 여부: 예

Request Body

해당 요청에는 본문이 없습니다.

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
   "NextToken": "string",
   "Tags": [
      {
         "Key": "string",
         "Value": "string"
      }
   ]
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반송합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

NextToken

응답 페이로드가 페이징된 경우 NextToken이 존재합니다. 후속 요청에서 액세스 포인트 설명의 다음 페이지를 가져오는 데 NextToken을 사용할 수 있습니다.

유형: 문자열

길이 제약: 최소 길이는 1. 최대 길이 128.

패턴: .+

Tags

지정된 EFS 리소스에 대한 태그의 배열입니다.

타입: Tag 객체 배열

오류

AccessPointNotFound

지정된 `AccessPointId` 값이 요청자의에 없는 경우 반환됩니다 AWS 계정.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 404

BadRequest

요청이 잘못되었거나 잘못된 파라미터 값 또는 필수 파라미터 누락 등의 오류가 있는 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 400

FileSystemNotFound

지정된 `FileSystemId` 값이 요청자의에 없는 경우 반환됩니다 AWS 계정.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 404

InternalServerError

서버 측에서 오류가 발생한 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 500

참고

언어별 AWS SDKs

- AWS 명령줄 인터페이스 V2
- AWS .NET V4용 SDK
- AWS SDK for C++
- AWS Go용 SDK v2

- AWS Java V2용 SDK
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS PHP V3용 SDK
- AWS Python용 SDK
- AWS SDK for Ruby V3

ModifyMountTargetSecurityGroups

탑재 대상에 대해 유효한 보안 그룹 세트를 수정합니다.

탑재 대상을 생성하면 Amazon EFS는 새 네트워크 인터페이스도 생성합니다. 자세한 내용은 CreateMountTarget 단원을 참조하십시오. 이 작업은 탑재 대상과 연결된 네트워크 인터페이스에 적용되는 보안 그룹을 요청에 제공된 `SecurityGroups`로 대체합니다. 이 작업을 수행하려면 탑재 대상의 네트워크 인터페이스가 생성되고 탑재 대상의 수명 주기 상태는 `deleted`가 아니어야 합니다.

이 작업에는 다음 작업에 대한 권한이 필요합니다.

- 탑재 대상의 파일 시스템에서의 `elasticfilesystem:ModifyMountTargetSecurityGroups` 작업
- 탑재 대상의 네트워크 인터페이스에서의 `ec2:ModifyNetworkInterfaceAttribute` 작업

Request Syntax

```
PUT /2015-02-01/mount-targets/MountTargetId/security-groups HTTP/1.1
Content-type: application/json

{
   "SecurityGroups": [ "string" ]
}
```

URI 요청 파라미터

요청은 다음 URI 파라미터를 사용합니다.

MountTargetId

수정하려는 보안 그룹이 있는 탑재 대상의 ID.

길이 제약: 최소 길이는 13자. 최대 길이는 45입니다.

패턴: `^fsmt-[0-9a-f]{8,40}$`

필수 여부: 예

요청 본문

요청은 JSON 형식으로 다음 데이터를 받습니다.

SecurityGroups

VPC 보안 그룹 ID의 배열.

유형: 문자열 어레이

배열 멤버: 최대 항목 수는 100개입니다.

길이 제약: 최소 길이는 11입니다. 최대 길이는 43입니다.

패턴: `^sg-[0-9a-f]{8,40}`

필수 여부: 아니요

응답 구문

```
HTTP/1.1 204
```

Response Elements

액션이 성공하면 해당 서비스는 빈 HTTP 본문과 함께 HTTP 204 응답을 되돌려줍니다.

Errors

BadRequest

요청이 잘못되었거나 잘못된 파라미터 값 또는 필수 파라미터 누락 등의 오류가 있는 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 400

IncorrectMountTargetState

탑재 대상이 작업에 적합한 올바른 상태가 아닌 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 409

InternalServerError

서버 측에서 오류가 발생한 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 500

MountTargetNotFound

호출자의 AWS 계정에 지정된 ID의 탑재 대상이 없는 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 404

SecurityGroupLimitExceeded

요청에 지정된 `SecurityGroups`의 수가 계정 할당량에 기반한 한도를 초과할 경우 반환됩니다. 일부 보안 그룹을 삭제하거나 계정 할당량 상향을 요청하세요. 자세한 내용은 Amazon VPC 사용 설명서의 Amazon VPC 할당량을 참조하세요(보안 그룹 테이블 참조).

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 400

SecurityGroupNotFound

지정된 보안 그룹 중 하나가 서브넷의 Virtual Private Cloud(VPC)에 존재하지 않는 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야

할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 400

예

탑재 대상의 보안 그룹 교체

다음 예는 탑재 대상과 연결된 네트워크 인터페이스에 적용되는 보안 그룹을 대체합니다.

샘플 요청

```
PUT /2015-02-01/mount-targets/fsmt-9a13661e/security-groups HTTP/1.1
Host: elasticfilesystem.us-west-2.amazonaws.com
x-amz-date: 20140620T223446Z
Authorization: <...>
Content-Type: application/json
Content-Length: 57

{
  "SecurityGroups" : [
  "sg-188d9f74"
  ]
}
```

샘플 응답

```
HTTP/1.1 204 No Content
x-amzn-RequestId: 01234567-89ab-cdef-0123-456789abcdef
```

참고

언어별 AWS SDKs

- AWS 명령줄 인터페이스 V2
- AWS .NET V4용 SDK
- AWS SDK for C++
- AWS Go용 SDK v2

- AWS Java V2용 SDK
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS PHP V3용 SDK
- AWS Python용 SDK
- AWS SDK for Ruby V3

PutAccountPreferences

이 작업을 사용하면 새 EFS 파일 시스템 및 탑재 대상 리소스에 대해 긴 17자(63비트) 또는 짧은 8자(32비트) 리소스 ID를 사용하도록 현재 AWS 리전 의 계정 기본 설정을 설정할 수 있습니다. 모든 기존 리소스 ID는 변경의 영향을 받지 않습니다. 긴 리소스 ID로 EFS가 전환함에 따라 옵트인 기간 동안 ID 기본 설정을 지정할 수 있습니다. 자세한 내용은 Amazon EFS 리소스 ID 관리를 참조하세요.

> **Note**
>
> 2021년 10월부터 짧은 8자 형식의 리소스 ID를 사용하도록 계정 기본 설정을 지정하려고 하면 오류가 발생합니다. 오류가 발생하여 파일 시스템 및 탑재 대상 리소스에 짧은 IDs 사용해야 하는 경우 AWS 지원팀에 문의하세요.

Request Syntax

```
PUT /2015-02-01/account-preferences HTTP/1.1
Content-type: application/json

{
   "ResourceIdType": "string"
}
```

URI 요청 파라미터

요청은 URI 파라미터를 사용하지 않습니다.

요청 본문

요청은 JSON 형식으로 다음 데이터를 받습니다.

ResourceIdType

현재에서 사용자의에 대해 설정할 EFS 리소스 ID 기본 설정을 AWS 리전LONG_ID (17자) 또는 SHORT_ID (8자) AWS 계정로 지정합니다.

> **Note**
>
> 2021년 10월부터 계정 기본 설정을 SHORT_ID로 지정할 때 오류 메시지가 표시됩니다. 오류가 발생하여 파일 시스템 및 탑재 대상 리소스에 짧은 IDs 사용해야 하는 경우 AWS 지원 팀에 문의하세요.

타입: 문자열

유효 값: LONG_ID | SHORT_ID

필수 여부: 예

응답 구문

```
HTTP/1.1 200
Content-type: application/json

{
   "ResourceIdPreference": {
      "ResourceIdType": "string",
      "Resources": [ "string" ]
   }
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반송합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

ResourceIdPreference

현재에서 리소스 유형과 사용자의 AWS 계정에 대한 ID 기본 설정을 설명합니다 AWS 리전.

타입: ResourceIdPreference 객체

오류

BadRequest

요청이 잘못되었거나 잘못된 파라미터 값 또는 필수 파라미터 누락 등의 오류가 있는 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 400

InternalServerError

서버 측에서 오류가 발생한 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 500

참고

언어별 AWS SDKs

- AWS 명령줄 인터페이스 V2
- AWS .NET V4용 SDK
- AWS SDK for C++
- AWS Go용 SDK v2
- AWS Java V2용 SDK
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS PHP V3용 SDK
- AWS Python용 SDK
- AWS SDK for Ruby V3

PutBackupPolicy

파일 시스템의 백업 정책을 업데이트합니다. 이 작업을 사용하면 파일 시스템의 자동 백업을 시작하거나 중지할 수 있습니다.

Request Syntax

```
PUT /2015-02-01/file-systems/FileSystemId/backup-policy HTTP/1.1
Content-type: application/json

{
   "BackupPolicy": {
      "Status": "string"
   }
}
```

URI 요청 파라미터

요청은 다음 URI 파라미터를 사용합니다.

FileSystemId

백업 정책을 업데이트할 EFS 파일 시스템을 지정합니다.

길이 제약: 최대 길이는 128입니다.

패턴: `^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$`

필수 여부: 예

요청 본문

요청은 JSON 형식으로 다음 데이터를 받습니다.

BackupPolicy

`PutBackupPolicy` 요청에 포함된 백업 정책.

타입: BackupPolicy객체

필수 여부: 예

응답 구문

```
HTTP/1.1 200
Content-type: application/json

{
   "BackupPolicy": {
      "Status": "string"
   }
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반송합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

BackupPolicy

자동 백업의 켜짐 또는 꺼짐을 나타내는 파일 시스템의 백업 정책을 설명합니다.

타입: BackupPolicy 객체

오류

BadRequest

요청이 잘못되었거나 잘못된 파라미터 값 또는 필수 파라미터 누락 등의 오류가 있는 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인

오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 400

FileSystemNotFound

지정된 `FileSystemId` 값이 요청자의에 없는 경우 반환됩니다 AWS 계정.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 404

IncorrectFileSystemLifeCycleState

파일 시스템의 수명 주기 상태가 “사용 가능”이 아닌 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 409

InternalServerError

서버 측에서 오류가 발생한 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 500

ValidationException

요청이 수행된 AWS 리전 에서 AWS Backup 서비스를 사용할 수 없는 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 400

참고

언어별 AWS SDKs

- AWS 명령줄 인터페이스 V2
- AWS .NET V4용 SDK
- AWS SDK for C++
- AWS Go용 SDK v2

- AWS Java V2용 SDK
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS PHP V3용 SDK
- AWS Python용 SDK
- AWS SDK for Ruby V3

PutFileSystemPolicy

Amazon EFS 파일 시스템에 Amazon EFS `FileSystemPolicy`를 적용합니다. 파일 시스템 정책은 IAM 리소스 기반 정책이며 여러 정책 명령문을 포함할 수 있습니다. 파일 시스템에는 항상 정확히 하나의 파일 시스템 정책이 있으며, 이는 기본 정책일 수도 있고 이 API 작업을 사용하여 설정되거나 업데이트된 명시적 정책일 수도 있습니다. EFS 파일 시스템 정책은 20,000자로 제한됩니다. 명시적 정책이 설정되면 기본 정책보다 우선합니다. 기본 파일 시스템 정책에 대한 자세한 내용은 기본 EFS 파일 시스템 정책 섹션을 참조하십시오.

> **Note**
>
> EFS 파일 시스템 정책은 20,000자로 제한됩니다.

이 작업에는 `elasticfilesystem:PutFileSystemPolicy` 액션에 대한 권한이 필요합니다.

Request Syntax

```
PUT /2015-02-01/file-systems/FileSystemId/policy HTTP/1.1
Content-type: application/json

{
   "BypassPolicyLockoutSafetyCheck": boolean,
   "Policy": "string"
}
```

URI 요청 파라미터

요청은 다음 URI 파라미터를 사용합니다.

FileSystemId

`FileSystemPolicy`를 생성 또는 업데이트하려는 EFS 파일 시스템의 ID입니다.

길이 제약: 최대 길이는 128입니다.

패턴: `^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$`

필수 여부: 예

요청 본문

요청은 JSON 형식으로 다음 데이터를 받습니다.

BypassPolicyLockoutSafetyCheck

(선택 사항)FileSystemPolicy 잠금 안전 검사를 우회할지 여부를 지정하는 부울입니다. 잠금 안전 검사는 요청의 정책이 요청을 하는 IAM 보안 주체를 잠글지 아니면 이 파일 시스템에 대해 향후 PutFileSystemPolicy 요청을 하지 못하도록 할지 결정합니다. 요청하는 IAM 보안 주체가 이 파일 시스템에서 후속 PutFileSystemPolicy 요청을 하지 않도록 하려는 경우에만 BypassPolicyLockoutSafetyCheck를 True로 설정합니다. 기본값은 False입니다.

유형: 부울

필수 항목 여부: 아니요

Policy

생성하려는 FileSystemPolicy입니다. JSON 형식의 정책 정의를 수락합니다. EFS 파일 시스템 정책은 20,000자로 제한됩니다. 파일 시스템 정책을 구성하는 요소에 대해 자세히 알아보려면 Amazon EFS 내 리소스 기반 정책 섹션을 참조하십시오.

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이는 20000입니다.

패턴: [\s\S]+

필수 여부: 예

응답 구문

```
HTTP/1.1 200
Content-type: application/json

{
   "FileSystemId": "string",
   "Policy": "string"
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반송합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

FileSystemId

`FileSystemPolicy`가 적용되는 EFS 파일 시스템을 지정합니다.

유형: 문자열

길이 제약: 최대 길이는 128입니다.

패턴: `^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$`

Policy

EFS 파일 시스템을 위한 JSON 형식의 `FileSystemPolicy`입니다.

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이는 20000입니다.

패턴: `[\s\S]+`

오류

BadRequest

요청이 잘못되었거나 잘못된 파라미터 값 또는 필수 파라미터 누락 등의 오류가 있는 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 400

FileSystemNotFound

지정된 `FileSystemId` 값이 요청자의에 없는 경우 반환됩니다 AWS 계정.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 404

IncorrectFileSystemLifeCycleState

파일 시스템의 수명 주기 상태가 “사용 가능”이 아닌 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 409

InternalServerError

서버 측에서 오류가 발생한 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 500

InvalidPolicyException

FileSystemPolicy가 잘못되었거나 잘못된 파라미터 값 또는 필수 파라미터 누락 등의 오류가 있는 경우 반환됩니다. 정책 록아웃 안전 검사 오류가 발생한 경우에 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 400

예

EFS FileSystemPolicy 생성

다음 요청은 모든 AWS 보안 주체FileSystemPolicy가 읽기 및 쓰기 권한으로 지정된 EFS 파일 시스템을 탑재할 수 있도록 허용하는를 생성합니다.

샘플 요청

```
PUT /2015-02-01/file-systems/fs-01234567/file-system-policy HTTP/1.1
{
    "Version": "2012-10-17",
    "Statement": [
```

```
        {
            "Effect": "Allow",
            "Action": [
                "elasticfilesystem:ClientMount",
                "elasticfilesystem:ClientWrite"
            ],
            "Principal": {
                "AWS": ["*"]
            },
        }
    ]
}
```

샘플 응답

```
{
    "Version": "2012-10-17",
    "Id": "1",
    "Statement": [
        {
            "Sid": "efs-statement-abcdef01-1111-bbbb-2222-111122224444",
            "Effect": "Allow",
            "Action": [
                "elasticfilesystem:ClientMount",
                "elasticfilesystem:ClientWrite"
            ],
            "Principal": {
                "AWS": ["*"]
            },
            "Resource":"arn:aws:elasticfilesystem:us-east-1:111122233334444:file-
system/fs-01234567"
        }
    ]
}
```

참고

언어별 AWS SDKs

- AWS 명령줄 인터페이스 V2
- AWS .NET V4용 SDK
- AWS SDK for C++

- AWS Go용 SDK v2
- AWS Java V2용 SDK
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS PHP V3용 SDK
- AWS Python용 SDK
- AWS SDK for Ruby V3

PutLifecycleConfiguration

이 작업을 사용하면 파일 시스템의 스토리지를 관리할 수 있습니다. `LifecycleConfiguration`은 다음을 정의하는 하나 이상의 `LifecyclePolicy` 객체로 구성됩니다.

- **`TransitionToIA`** - 파일 시스템의 파일을 기본 스토리지(Standard 스토리지 클래스)에서 Inrequent Access(IA) 스토리지로 이동하는 시점입니다.
- **`TransitionToArchive`** - 파일 시스템의 파일을 현재 스토리지 클래스(IA 또는 Standard 스토리지)에서 Archive 스토리지로 이동하는 시점입니다.

  IA 스토리지로 전환하기 전에는 파일 시스템을 Archive 스토리지로 전환할 수 없습니다. 따라서 TransitionToArchive는 설정되지 않았거나 TransitionToIA 이후여야 합니다.

  > **Note**
  >
  > Archive 스토리지 클래스는 탄력적 처리량 모드 및 범용 성능 모드를 사용하는 파일 시스템에만 사용할 수 있습니다.

- **`TransitionToPrimaryStorageClass`** - IA 또는 Archive 스토리지에서 액세스된 후 파일 시스템의 파일을 기본 스토리지(Standard 스토리지 클래스)로 다시 이동할지 여부입니다.

자세한 내용은 파일 시스템 스토리지 관리 섹션을 참조하십시오.

각 Amazon EFS 파일 시스템은 파일 시스템의 모든 파일에 적용되는 하나의 수명 주기 구성을 지원합니다. 지정된 파일 시스템에 대한 `LifecycleConfiguration` 객체가 이미 존재하는 경우 `PutLifecycleConfiguration` 직접 호출을 통해 기존 구성이 수정됩니다. 요청 본문에 빈 `LifecyclePolicies` 배열을 `PutLifecycleConfiguration`으로 직접 호출하면 기존 `LifecycleConfiguration`이 모두 삭제됩니다. 요청에서 다음을 지정합니다.

- 수명 주기 관리를 활성화, 비활성화 또는 수정하려는 파일 시스템의 ID
- 파일을 IA 스토리지로, Archive 스토리지로, 다시 기본 스토리지로 이동하는 시점을 정의하는 `LifecyclePolicy` 객체의 `LifecyclePolicies` 배열입니다.

> **Note**
>
> Amazon EFS에서는 각 `LifecyclePolicy` 객체에 단일 전환만 포함해야 하므로 `LifecyclePolicies` 배열을 별도의 `LifecyclePolicy` 객체로 구성해야 합니다. 자세한 내용을 알아보려면 다음 섹션의 예제 요청을 참조하세요.

이 작업에는 `elasticfilesystem:PutLifecycleConfiguration` 작업에 대한 권한이 필요합니다.

암호화된 파일 시스템에 `LifecycleConfiguration` 객체를 적용하려면 암호화된 파일 시스템을 생성할 때와 동일한 AWS Key Management Service 권한이 필요합니다.

Request Syntax

```
PUT /2015-02-01/file-systems/FileSystemId/lifecycle-configuration HTTP/1.1
Content-type: application/json

{
   "LifecyclePolicies": [
      {
         "TransitionToArchive": "string",
         "TransitionToIA": "string",
         "TransitionToPrimaryStorageClass": "string"
      }
   ]
}
```

URI 요청 파라미터

요청은 다음 URI 파라미터를 사용합니다.

FileSystemId

`LifecycleConfiguration` 객체(문자열)를 생성할 파일 시스템의 ID입니다.

길이 제약: 최대 길이는 128입니다.

패턴: `^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$`

필수 여부: 예

요청 본문

요청은 JSON 형식으로 다음 데이터를 받습니다.

LifecyclePolicies

파일 시스템의 `LifecycleConfiguration` 객체를 정의하는 `LifecyclePolicy` 객체의 배열입니다. `LifecycleConfiguration` 객체는 수명 주기 관리에 다음 사항을 알립니다.

- **`TransitionToIA`** - 파일 시스템의 파일을 기본 스토리지(Standard 스토리지 클래스)에서 Inrequent Access(IA) 스토리지로 이동하는 시점입니다.
- **`TransitionToArchive`** - 파일 시스템의 파일을 현재 스토리지 클래스(IA 또는 Standard 스토리지)에서 Archive 스토리지로 이동하는 시점입니다.

  IA 스토리지로 전환하기 전에는 파일 시스템을 Archive 스토리지로 전환할 수 없습니다. 따라서 TransitionToArchive는 설정되지 않았거나 TransitionToIA 이후여야 합니다.

  > **Note**
  >
  > Archive 스토리지 클래스는 탄력적 처리량 모드 및 범용 성능 모드를 사용하는 파일 시스템에만 사용할 수 있습니다.

- **`TransitionToPrimaryStorageClass`** - IA 또는 Archive 스토리지에서 액세스된 후 파일 시스템의 파일을 기본 스토리지(Standard 스토리지 클래스)로 다시 이동할지 여부입니다.

> **Note**
>
> `put-lifecycle-configuration` CLI 명령 또는 `PutLifecycleConfiguration` API 작업을 사용할 때, Amazon EFS에서는 각 `LifecyclePolicy` 객체에 단일 전환만 포함하도록 요구합니다. 즉, 요청 본문에서 `LifecyclePolicies`는 각 스토리지 전환 시 하나의 객체인 `LifecyclePolicy` 객체의 배열로 구조화되어야 합니다. 자세한 내용을 알아보려면 다음 섹션의 예제 요청을 참조하세요.

타입: LifecyclePolicy객체 배열

배열 멤버: 최대 항목 수는 3개입니다.

필수 항목 여부: 예

응답 구문

```
HTTP/1.1 200
Content-type: application/json

{
   "LifecyclePolicies": [
      {
         "TransitionToArchive": "string",
         "TransitionToIA": "string",
         "TransitionToPrimaryStorageClass": "string"
      }
   ]
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반송합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

LifecyclePolicies

수명 주기 관리 정책들의 배열. EFS는 파일 시스템당 최대 하나의 정책을 지원합니다.

타입: LifecyclePolicy객체 배열

배열 멤버: 최대 항목 수는 3개입니다.

오류

BadRequest

요청이 잘못되었거나 잘못된 파라미터 값 또는 필수 파라미터 누락 등의 오류가 있는 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 400

FileSystemNotFound

지정된 `FileSystemId` 값이 요청자의에 없는 경우 반환됩니다 AWS 계정.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 404

IncorrectFileSystemLifeCycleState

파일 시스템의 수명 주기 상태가 “사용 가능”이 아닌 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인

오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 409

InternalServerError

서버 측에서 오류가 발생한 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 500

예제

수명 주기 구성 생성

다음 예제에서는 `PutLifecycleConfiguration` 액션을 사용하여 `LifecyclePolicy` 객체를 만듭니다. 이 예시에서는 EFS에 다음을 수행하도록 지시하는 수명 주기 정책을 생성합니다.

- Standard 스토리지에서 지난 30일 동안 액세스하지 않은 파일 시스템 내 파일을 모두 IA 스토리지로 이동합니다.
- Standard 스토리지에서 지난 90일 동안 액세스하지 않은 파일 시스템 내 파일을 모두 Archive 스토리지로 이동합니다.
- IA 또는 Archive 스토리지에서 액세스한 파일은 다시 Standard 스토리지로 이동합니다. Archive 스토리지 클래스는 탄력적 처리량 모드 및 범용 성능 모드를 사용하는 파일 시스템에만 사용할 수 있습니다.

자세한 내용은 EFS 스토리지 클래스 및 파일 시스템 스토리지 관리 섹션을 참조하십시오.

샘플 요청

```
PUT /2015-02-01/file-systems/fs-0123456789abcdefb/lifecycle-configuration HTTP/1.1
Host: elasticfilesystem.us-west-2.amazonaws.com
x-amz-date: 20181122T232908Z
Authorization: <...>
Content-type: application/json
Content-Length: 86

{
   "LifecyclePolicies": [
      {
         "TransitionToArchive": "AFTER_90_DAYS"
      },
      {
         "TransitionToIA": "AFTER_30_DAYS"
      },
      {
         "TransitionToPrimaryStorage": "AFTER_1_ACCESS"
      }
   ]
}
```

샘플 응답

```
HTTP/1.1 200 OK
x-amzn-RequestId: 01234567-89ab-cdef-0123-456789abcdef
Content-type: application/json
Content-Length: 86

{
    "LifecyclePolicies": [
      {
         "TransitionToArchive": "AFTER_90_DAYS"
      },
      {
         "TransitionToIA": "AFTER_30_DAYS"
      },
      {
         "TransitionToPrimaryStorage": "AFTER_1_ACCESS"
      }
    ]
```

```
}
```

put-lifecycle-configuration CLI 요청 예제

이 예제는 PutLifecycleConfiguration의 한 가지 사용법을 보여줍니다.

샘플 요청

```
aws efs put-lifecycle-configuration \
   --file-system-id fs-0123456789abcdefb \
   --lifecycle-policies "[{"TransitionToArchive":"AFTER_90_DAYS"},
     {"TransitionToIA":"AFTER_30_DAYS"},
     {"TransitionToPrimaryStorageClass":"AFTER_1_ACCESS"}]
   --region us-west-2 \
   --profile adminuser
```

샘플 응답

```
{
   "LifecyclePolicies": [
       {
           "TransitionToArchive": "AFTER_90_DAYS"
       },
       {
           "TransitionToIA": "AFTER_30_DAYS"
       },
       {
           "TransitionToPrimaryStorageClass": "AFTER_1_ACCESS"
       }
   ]
}
```

수명 주기 관리 비활성화

다음 예제에서는 지정된 파일 시스템의 수명 주기 관리를 비활성화합니다.

샘플 요청

```
PUT /2015-02-01/file-systems/fs-01234567/lifecycle-configuration HTTP/1.1
Host: elasticfilesystem.us-west-2.amazonaws.com
```

```
x-amz-date: 20181122T232908Z
Authorization: <...>
Content-type: application/json
Content-Length: 86

{
   "LifecyclePolicies": [ ]
}
```

샘플 응답

```
HTTP/1.1 200 OK
x-amzn-RequestId: 01234567-89ab-cdef-0123-456789abcdef
Content-type: application/json
Content-Length: 86

{
   "LifecyclePolicies": [ ]
}
```

참고

언어별 AWS SDKs

- AWS 명령줄 인터페이스 V2
- AWS .NET V4용 SDK
- AWS SDK for C++
- AWS Go용 SDK v2
- AWS Java V2용 SDK
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS PHP V3용 SDK
- AWS Python용 SDK
- AWS SDK for Ruby V3

TagResource

EFS 리소스용 태그를 생성합니다. 이 API 작업을 사용하여 EFS 파일 시스템 및 액세스 포인트를 위한 태그를 생성할 수 있습니다.

이 작업에는 `elasticfilesystem:TagResource` 액션에 대한 권한이 필요합니다.

Request Syntax

```
POST /2015-02-01/resource-tags/ResourceId HTTP/1.1
Content-type: application/json

{
   "Tags": [
      {
         "Key": "string",
         "Value": "string"
      }
   ]
}
```

URI 요청 파라미터

요청은 다음 URI 파라미터를 사용합니다.

ResourceId

태그를 생성하려는 EFS 리소스를 지정하는 ID입니다.

길이 제약: 최대 길이는 128입니다.

패턴: `^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:(access-point/fsap|file-system/fs)-[0-9a-f]{8,40}|fs(ap)?-[0-9a-f]{8,40})$`

필수 여부: 예

요청 본문

요청은 JSON 형식으로 다음 데이터를 받습니다.

Tags

추가할 Tag 객체 배열. 각 Tag 객체는 키-값 페어입니다.

타입: Tag객체 배열

필수 항목 여부: 예

응답 구문

```
HTTP/1.1 200
```

Response Elements

작업이 성공하면 서비스가 비어 있는 HTTP 본문과 함께 HTTP 200 응답을 다시 전송합니다.

오류

AccessPointNotFound

지정된 AccessPointId 값이 요청자의에 없는 경우 반환됩니다 AWS 계정.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 404

BadRequest

요청이 잘못되었거나 잘못된 파라미터 값 또는 필수 파라미터 누락 등의 오류가 있는 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 400

FileSystemNotFound

지정된 `FileSystemId` 값이 요청자의에 없는 경우 반환됩니다 AWS 계정.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 404

InternalServerError

서버 측에서 오류가 발생한 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 500

예제

파일 시스템에 태그 생성

다음 요청은 지정된 파일 시스템에 세 개의 태그("key1", "key2", 및 "key3")를 생성합니다.

샘플 요청

```
POST /2015-02-01/tag-resource/fs-01234567 HTTP/1.1
Host: elasticfilesystem.us-west-2.amazonaws.com
x-amz-date: 20140620T221118Z
Authorization: <...>
Content-Type: application/json
Content-Length: 160

{
    "Tags": [
        {
            "Key": "key1",
            "Value": "value1"
        },
        {
            "Key": "key2",
            "Value": "value2"
        },
        {
            "Key": "key3",
            "Value": "value3"
        }
    ]
}
```

샘플 응답

```
HTTP/1.1 204 no content
x-amzn-RequestId: 01234567-89ab-cdef-0123-456789abcdef
```

참고

언어별 AWS SDKs

- AWS 명령줄 인터페이스 V2
- AWS .NET V4용 SDK
- AWS SDK for C++
- AWS Go용 SDK v2
- AWS Java V2용 SDK
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS PHP V3용 SDK
- AWS Python용 SDK
- AWS SDK for Ruby V3

UntagResource

EFS 리소스에서 태그를 제거합니다. 이 API 작업을 사용하여 EFS 파일 시스템 및 액세스 포인트를 위한 태그를 제거할 수 있습니다.

이 작업에는 `elasticfilesystem:UntagResource` 액션에 대한 권한이 필요합니다.

Request Syntax

```
DELETE /2015-02-01/resource-tags/ResourceId?tagKeys=TagKeys HTTP/1.1
```

URI 요청 파라미터

요청은 다음 URI 파라미터를 사용합니다.

ResourceId

태그를 제거할 EFS 리소스를 지정합니다.

길이 제약: 최대 길이는 128입니다.

패턴: `^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:(access-point/fsap|file-system/fs)-[0-9a-f]{8,40}|fs(ap)?-[0-9a-f]{8,40})$`

필수 여부: 예

TagKeys

지정된 EFS 리소스에서 제거하려는 키-값 태그 페어 키입니다.

배열 멤버: 최소수는 1개입니다. 최대수 50개.

길이 제약: 최소 길이 1. 최대 길이 128.

패턴: `^(?![aA]{1}[wW]{1}[sS]{1}:)([\p{L}\p{Z}\p{N}_.:/=+\-@]+)$`

필수 여부: 예

Request Body

해당 요청에는 본문이 없습니다.

Response Syntax

```
HTTP/1.1 200
```

Response Elements

작업이 성공하면 서비스가 비어 있는 HTTP 본문과 함께 HTTP 200 응답을 다시 전송합니다.

오류

AccessPointNotFound

지정된 `AccessPointId` 값이 요청자의에 없는 경우 반환됩니다 AWS 계정.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 404

BadRequest

요청이 잘못되었거나 잘못된 파라미터 값 또는 필수 파라미터 누락 등의 오류가 있는 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야

할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 400

FileSystemNotFound

지정된 `FileSystemId` 값이 요청자의에 없는 경우 반환됩니다 AWS 계정.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 404

InternalServerError

서버 측에서 오류가 발생한 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 500

참고

언어별 AWS SDKs

- AWS 명령줄 인터페이스 V2
- AWS .NET V4용 SDK
- AWS SDK for C++
- AWS Go용 SDK v2
- AWS Java V2용 SDK
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS PHP V3용 SDK
- AWS Python용 SDK
- AWS SDK for Ruby V3

UpdateFileSystem

기존 파일 시스템의 처리량 모드 또는 프로비저닝된 처리량을 업데이트합니다.

Request Syntax

```
PUT /2015-02-01/file-systems/FileSystemId HTTP/1.1
Content-type: application/json

{
   "ProvisionedThroughputInMibps": number,
   "ThroughputMode": "string"
}
```

URI 요청 파라미터

요청은 다음 URI 파라미터를 사용합니다.

FileSystemId

업데이트할 파일 시스템의 ID입니다.

길이 제약: 최대 길이는 128입니다.

패턴: `^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$`

필수 여부: 예

요청 본문

요청은 JSON 형식으로 다음 데이터를 받습니다.

ProvisionedThroughputInMibps

(선택 사항)생성 중인 파일 시스템에 대해 프로비저닝하고자 하는 처리량입니다. 초당 메비바이트(MiBps)로 측정됩니다. `ThroughputMode`이 `provisioned`로 설정된 경우 필수입니다. 유효한 값은 1~3414MiBps이며 상한은 리전에 따라 다릅니다. 이 제한을 늘리려면에 문의하세요 지원. 자세한 내용을 알아보려면 Amazon Elastic File System 사용 설명서의 증가할 수 있는 Amazon EFS 할당량을 참조하세요.

유형: Double

유효한 범위: 최소값은 1.0입니다.

필수 여부: 아니요

ThroughputMode

(선택 사항)파일 시스템의 처리량 모드를 업데이트합니다. 처리량 모드를 업데이트하지 않는 경우 요청에 이 값을 제공할 필요가 없습니다. ThroughputMode를 provisioned로 변경하면 ProvisionedThroughputInMibps의 값도 설정해야 합니다.

타입: 문자열

유효 값: bursting | provisioned | elastic

필수 항목 여부: 아니요

응답 구문

```
HTTP/1.1 202
Content-type: application/json

{
   "AvailabilityZoneId": "string",
   "AvailabilityZoneName": "string",
   "CreationTime": number,
   "CreationToken": "string",
   "Encrypted": boolean,
   "FileSystemArn": "string",
   "FileSystemId": "string",
   "FileSystemProtection": {
      "ReplicationOverwriteProtection": "string"
   },
   "KmsKeyId": "string",
   "LifeCycleState": "string",
   "Name": "string",
   "NumberOfMountTargets": number,
   "OwnerId": "string",
   "PerformanceMode": "string",
   "ProvisionedThroughputInMibps": number,
   "SizeInBytes": {
      "Timestamp": number,
```

```
      "Value": number,
      "ValueInArchive": number,
      "ValueInIA": number,
      "ValueInStandard": number
   },
   "Tags": [
      {
         "Key": "string",
         "Value": "string"
      }
   ],
   "ThroughputMode": "string"
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 202 응답을 다시 전송합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

AvailabilityZoneId

파일 시스템이 위치한 가용 영역의 고유하고 일관된 식별자로, One Zone 파일 시스템에만 유효합니다. 예를 들어 `use1-az1`는 us-east-1의 가용 영역 ID AWS 리전이며 모든에서 위치가 동일합니다 AWS 계정.

유형: 문자열

AvailabilityZoneName

파일 시스템이 있는 AWS 가용 영역을 설명하며 One Zone 파일 시스템에만 유효합니다. 자세한 내용을 알아보려면 Amazon EFS 사용 설명서의 EFS 스토리지 클래스 사용을 참조하세요.

유형: 문자열

길이 제한: 최소 길이는 1. 최대 길이는 64.

패턴: `.+`

CreationTime

파일 시스템이 생성된 시간(초)입니다(1970-01-01T00:00:00Z 이후).

유형: 타임스탬프

CreationToken

요청에 지정된 불투명한 문자열입니다.

유형: 문자열

길이 제한: 최소 길이는 1. 최대 길이는 64.

패턴: `.+`

Encrypted

true인 경우 파일 시스템이 암호화되었음을 나타내는 부울 값입니다.

유형: Boolean

FileSystemArn

EFS 파일 시스템의 Amazon 리소스 이름(ARN)으로서 `arn:aws:elasticfilesystem:region:account-id:file-system/file-system-id` 형식입니다. 샘플 데이터를 사용한 예: `arn:aws:elasticfilesystem:us-west-2:1111333322228888:file-system/fs-01234567`

유형: 문자열

FileSystemId

Amazon EFS에서 할당한 파일 시스템의 ID입니다.

유형: 문자열

길이 제약: 최대 길이는 128입니다.

패턴: `^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$`

FileSystemProtection

파일 시스템의 보호를 설명합니다.

유형: FileSystemProtectionDescription객체

KmsKeyId

암호화된 파일 시스템을 보호하는 데 AWS KMS key 사용되는의 ID입니다.

유형: 문자열

길이 제약: 최대 길이 2048.

패턴: `^([0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}|mrk-[0-9a-f]{32}|alias/[a-zA-Z0-9/_-]+|(arn:aws[-a-z]*:kms:[a-z0-9-]+:\d{12}:((key/[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12})|(key/mrk-[0-9a-f]{32})|(alias/[a-zA-Z0-9/_-]+))))$`

LifeCycleState

파일 시스템의 수명 주기 단계입니다.

타입: 문자열

유효 값: `creating | available | updating | deleting | deleted | error`

Name

태그를 포함하여 파일 시스템에 `Name` 태그를 추가할 수 있습니다. 자세한 내용은 CreateFileSystem 단원을 참조하십시오. 파일 시스템에 `Name` 태그가 있는 경우 Amazon EFS는 이 필드에 값을 반환합니다.

유형: 문자열

길이 제약: 최대 길이 256.

패턴: `^([\p{L}\p{Z}\p{N}_.:/=+\-@]*)$`

NumberOfMountTargets

파일 시스템에 있는 탑재 대상의 현재 수. 자세한 내용은 CreateMountTarget 단원을 참조하십시오.

유형: 정수

유효 범위: 최소값 0.

OwnerId

파일 시스템을 AWS 계정 생성한 입니다.

유형: 문자열

길이 제약 조건: 최대 길이는 14입니다.

패턴: ^(\d{12})|(\d{4}-\d{4}-\d{4})$

PerformanceMode

파일 시스템의 성능 모드입니다.

타입: 문자열

유효 값: generalPurpose | maxIO

ProvisionedThroughputInMibps

MiBps로 측정되는 프로비저닝된 파일 시스템 처리량을 정의합니다. provisioned로 설정된 ThroughputMode을 사용하는 파일 시스템에 유효합니다.

유형: Double

유효한 범위: 최소값은 1.0입니다.

SizeInBytes

파일 시스템에서 해당 Value 필드에 저장된 데이터의 가장 최근 알려진 측정 크기(바이트) 및 해당 Timestamp 필드에서 해당 크기가 결정된 시간입니다. Timestamp 값은 1970-01-01T00:00:00Z 이후의 정수 시간(초)입니다. 이 SizeInBytes 값은 파일 시스템의 일관된 스냅샷 크기를 나타내지는 않지만 파일 시스템에 쓰기가 없는 경우 최종적으로 일관성을 유지합니다. 즉, 몇 시간 이상 파일 시스템을 수정하지 않은 경우에만 SizeInBytes가 실제 크기를 나타냅니다. 그렇지 않으면 값이 특정 시점의 파일 시스템 크기와 정확히 일치하지 않습니다.

유형: FileSystemSize객체

Tags

파일 시스템과 연결된 태그로, Tag 객체 배열로 표시됩니다.

타입: Tag객체 배열

ThroughputMode

파일 시스템의 처리량 모드를 표시합니다. 자세한 내용을 알아보려면 Amazon Elastic File System 사용 설명서의 처리량 모드를 참조하세요.

타입: 문자열

유효 값: bursting | provisioned | elastic

오류

BadRequest

요청이 잘못되었거나 잘못된 파라미터 값 또는 필수 파라미터 누락 등의 오류가 있는 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 400

FileSystemNotFound

지정된 `FileSystemId` 값이 요청자의에 없는 경우 반환됩니다 AWS 계정.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 404

IncorrectFileSystemLifeCycleState

파일 시스템의 수명 주기 상태가 “사용 가능”이 아닌 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 409

InsufficientThroughputCapacity

추가 처리량을 프로비저닝할 용량이 충분하지 않은 경우 반환됩니다. 프로비저닝된 처리량 모드에서 파일 시스템을 생성하려고 할 때, 기존 파일 시스템의 프로비저닝된 처리량을 늘리려고 할 때 또는 기존 파일 시스템을 버스팅 처리량에서 프로비저닝된 처리량 모드로 변경하려고 할 때 이 값이 반환될 수 있습니다. 나중에 다시 시도해 주세요.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 503

InternalServerError

서버 측에서 오류가 발생한 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 500

ThroughputLimitExceeded

처리량 한도인 1024MiB/s에 도달하여 처리량 모드 또는 프로비저닝된 처리량을 변경할 수 없는 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 400

TooManyRequests

처리량 모드를 변경하거나 프로비저닝된 처리량 값을 낮추기 전에 최소 24시간을 기다리지 않으면 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인

오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 429

참고 항목

언어별 AWS SDKs

- AWS 명령줄 인터페이스 V2
- AWS .NET V4용 SDK
- AWS SDK for C++
- AWS Go용 SDK v2
- AWS Java V2용 SDK
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS PHP V3용 SDK
- AWS Python용 SDK
- AWS SDK for Ruby V3

UpdateFileSystemProtection

파일 시스템의 보호를 업데이트합니다.

이 작업에는 `elasticfilesystem:UpdateFileSystemProtection` 액션에 대한 권한이 필요합니다.

Request Syntax

```
PUT /2015-02-01/file-systems/FileSystemId/protection HTTP/1.1
Content-type: application/json

{
   "ReplicationOverwriteProtection": "string"
}
```

URI 요청 파라미터

요청은 다음 URI 파라미터를 사용합니다.

FileSystemId

업데이트할 파일 시스템의 ID입니다.

길이 제약: 최대 길이는 128입니다.

패턴: `^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$`

필수 여부: 예

요청 본문

요청은 JSON 형식으로 다음 데이터를 받습니다.

ReplicationOverwriteProtection

파일 시스템의 복제 덮어쓰기 보호 상태입니다.

- ENABLED – 파일 시스템은 복제 구성의 대상 파일 시스템으로 사용할 수 없습니다. 파일 시스템은 쓰기 가능합니다. 복제 덮어쓰기 보호는 기본적으로 ENABLED 상태입니다.

- DISABLED – 파일 시스템은 복제 구성의 대상 파일 시스템으로 사용할 수 있습니다. 파일 시스템은 읽기 전용이며 EFS 복제를 통해서만 수정할 수 있습니다.
- REPLICATING – 파일 시스템은 복제 구성의 대상 파일 시스템으로 사용 중입니다. 파일 시스템은 읽기 전용이며 EFS 복제를 통해서만 수정됩니다.

복제 구성을 삭제하면 파일 시스템의 복제 덮어쓰기 보호가 다시 활성화되고 파일 시스템은 쓰기 가능해집니다.

타입: 문자열

유효 값: ENABLED | DISABLED | REPLICATING

필수 항목 여부: 아니요

응답 구문

```
HTTP/1.1 200
Content-type: application/json

{
   "ReplicationOverwriteProtection": "string"
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반송합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

ReplicationOverwriteProtection

파일 시스템의 복제 덮어쓰기 보호 상태입니다.

- ENABLED – 파일 시스템은 복제 구성의 대상 파일 시스템으로 사용할 수 없습니다. 파일 시스템은 쓰기 가능합니다. 복제 덮어쓰기 보호는 기본적으로 ENABLED 상태입니다.
- DISABLED – 파일 시스템은 복제 구성의 대상 파일 시스템으로 사용할 수 있습니다. 파일 시스템은 읽기 전용이며 EFS 복제를 통해서만 수정할 수 있습니다.
- REPLICATING – 파일 시스템은 복제 구성의 대상 파일 시스템으로 사용 중입니다. 파일 시스템은 읽기 전용이며 EFS 복제를 통해서 수정됩니다.

복제 구성을 삭제하면 파일 시스템의 복제 덮어쓰기 보호가 다시 활성화되고 파일 시스템은 쓰기 가능해집니다.

타입: 문자열

유효 값: `ENABLED | DISABLED | REPLICATING`

오류

BadRequest

요청이 잘못되었거나 잘못된 파라미터 값 또는 필수 파라미터 누락 등의 오류가 있는 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 400

FileSystemNotFound

지정된 `FileSystemId` 값이 요청자의에 없는 경우 반환됩니다 AWS 계정.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인

오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 404

IncorrectFileSystemLifeCycleState

파일 시스템의 수명 주기 상태가 “사용 가능”이 아닌 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 409

InsufficientThroughputCapacity

추가 처리량을 프로비저닝할 용량이 충분하지 않은 경우 반환됩니다. 프로비저닝된 처리량 모드에서 파일 시스템을 생성하려고 할 때, 기존 파일 시스템의 프로비저닝된 처리량을 늘리려고 할 때 또는 기존 파일 시스템을 버스팅 처리량에서 프로비저닝된 처리량 모드로 변경하려고 할 때 이 값이 반환될 수 있습니다. 나중에 다시 시도해 주세요.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 503

InternalServerError

서버 측에서 오류가 발생한 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 500

ReplicationAlreadyExists

파일 시스템이 복제 구성에 이미 포함되어 있는 경우 반환됩니다.>

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 409

ThroughputLimitExceeded

처리량 한도인 1024MiB/s에 도달하여 처리량 모드 또는 프로비저닝된 처리량을 변경할 수 없는 경우 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 400

TooManyRequests

처리량 모드를 변경하거나 프로비저닝된 처리량 값을 낮추기 전에 최소 24시간을 기다리지 않으면 반환됩니다.

ErrorCode

오류 코드는 오류 상태를 식별하는 고유한 문자열입니다. 이 코드는 오류를 감지하여 유형별로 처리하는 프로그램에서 판독하고 이해하기 위한 것입니다.

Message

오류 메시지는 오류 상태에 관한 일반적인 설명을 담고 있으며 영어로 제공됩니다. 이는 기계가 아닌 사람을 위한 것입니다. 간단한 프로그램은 자체적으로 확인되지 않거나 어떻게 처리해야 할지 모르는 오류 상태가 발생하면 최종 사용자에게 직접 메시지를 표시합니다. 좀 더 포괄적인 오류 처리 기능을 갖고 있고 적절히 국제화된 고급 프로그램은 오류 메시지를 무시할 가능성이 큽니다.

HTTP 상태 코드: 429

참고 항목

언어별 AWS SDKs

- AWS 명령줄 인터페이스 V2
- AWS .NET V4용 SDK
- AWS SDK for C++
- AWS Go용 SDK v2
- AWS Java V2용 SDK
- AWS SDK for JavaScript V3

- AWS SDK for Kotlin
- AWS PHP V3용 SDK
- AWS Python용 SDK
- AWS SDK for Ruby V3

데이터 타입

다음 데이터 타입이 지원됩니다.

- AccessPointDescription
- BackupPolicy
- CreationInfo
- Destination
- DestinationToCreate
- FileSystemDescription
- FileSystemProtectionDescription
- FileSystemSize
- LifecyclePolicy
- MountTargetDescription
- PosixUser
- ReplicationConfigurationDescription
- ResourceIdPreference
- RootDirectory
- Tag

AccessPointDescription

EFS 파일 시스템 액세스 포인트에 대한 설명을 제공합니다.

내용

AccessPointArn

액세스 포인트와 연결된 고유한 Amazon 리소스 이름(ARN)입니다.

유형: 문자열

길이 제약: 최대 길이는 128입니다.

패턴: `^arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:access-point/fsap-[0-9a-f]{8,40}$`

필수 여부: 아니요

AccessPointId

Amazon EFS에서 할당한 액세스 포인트의 ID입니다.

유형: 문자열

길이 제약: 최대 길이는 128입니다.

패턴: `^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:access-point/fsap-[0-9a-f]{8,40}|fsap-[0-9a-f]{8,40})$`

필수 여부: 아니요

ClientToken

멱등성 생성을 보장하기 위해 요청에 지정된 불투명한 문자열입니다.

유형: 문자열

길이 제한: 최소 길이는 1. 최대 길이는 64.

패턴: `.+`

필수 여부: 아니요

FileSystemId

액세스 포인트가 적용되는 EFS 파일 시스템의 ID입니다.

유형: 문자열

길이 제약: 최대 길이는 128입니다.

패턴: `^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$`

필수 여부: 아니요

LifeCycleState

액세스 포인트의 수명 주기 단계를 식별합니다.

타입: 문자열

유효 값: `creating | available | updating | deleting | deleted | error`

필수 여부: 아니요

Name

이 액세스 포인트의 이름입니다. `Name` 태그의 값입니다.

유형: 문자열

필수 여부: 아니요

OwnerId

액세스 포인트 리소스를 소유하는 AWS 계정을 식별합니다.

유형: 문자열

길이 제약 조건: 최대 길이는 14입니다.

패턴: `^(\d{12})|(\d{4}-\d{4}-\d{4})$`

필수 여부: 아니요

PosixUser

액세스 포인트를 사용하여 NFS 클라이언트가 수행하는 모든 파일 작업에 사용되는 액세스 포인트에서 사용자 ID, 그룹 ID 및 보조 그룹 ID를 포함한 전체 POSIX ID입니다.

유형: PosixUser객체

필수 여부: 아니요

RootDirectory

액세스 포인트가 액세스 포인트를 사용하는 NFS 클라이언트에 루트 디렉터리로 노출하는 EFS 파일 시스템의 디렉터리입니다.

유형: RootDirectory객체

필수 여부: 아니요

Tags

액세스 포인트와 관련된 태그는 태그 객체의 배열로 표시됩니다.

타입: Tag객체 배열

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

BackupPolicy

자동 일일 백업을 생성하는 데 사용되는 파일 시스템의 백업 정책입니다. status 값이 ENABLED인 경우 파일 시스템이 자동으로 백업됩니다. 자세한 내용은 자동 백업을 참조하세요.

내용

Status

파일 시스템의 백업 정책 상태를 설명합니다.

- **ENABLED** - EFS는 파일 시스템을 자동으로 백업합니다.
- **ENABLING** - EFS는 파일 시스템에 대한 자동 백업을 켭니다.
- **DISABLED** - 파일 시스템에 대한 자동 백업을 끕니다.
- **DISABLING** - EFS는 파일 시스템에 대한 자동 백업을 끕니다.

타입: 문자열

유효 값: `ENABLED | ENABLING | DISABLED | DISABLING`

필수 여부: 예

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

CreationInfo

지정된 `RootDirectory` > `Path`가 존재하지 않는 경우 필요합니다. 액세스 포인트의 `RootDirectory` > `Path`에 적용할 POSIX ID 및 권한을 지정합니다. 액세스 포인트 루트 디렉터리가 없는 경우 클라이언트가 액세스 포인트에 연결할 때 EFS는 이러한 설정을 사용하여 액세스 포인트 루트 디렉터리를 만듭니다. `CreationInfo`를 지정할 때 모든 속성에 대한 값을 포함해야 합니다.

Amazon EFS는 사용자가 CreationInfo(OwNuID, ownGID 및 디렉터리에 대한 권한)를 제공한 경우에만 루트 디렉터리를 생성합니다. 이 정보를 제공하지 않으면 Amazon EFS는 루트 디렉터리를 생성하지 않습니다. 루트 디렉터리가 없으면 액세스 포인트를 사용하는 탑재 시도가 실패합니다.

> **Important**
>
> `CreationInfo`를 지정하지 않고 지정된 `RootDirectory`가 없는 경우 액세스 포인트를 사용하여 파일 시스템을 탑재하려는 시도가 실패합니다.

내용

OwnerGid

`RootDirectory`에 적용할 POSIX 그룹 ID를 지정합니다. 0에서 2^32(4294967295) 사이의 값을 허용합니다.

타입: Long

유효 범위: 최소값은 0입니다. 최댓값은 4294967295입니다.

필수 여부: 예

OwnerUid

`RootDirectory`에 적용할 POSIX 사용자 ID를 지정합니다. 0에서 2^32(4294967295) 사이의 값을 허용합니다.

타입: Long

유효 범위: 최소값은 0입니다. 최댓값은 4294967295입니다.

필수 여부: 예

Permissions

`RootDirectory`에 적용할 POSIX 권한을 파일의 모드 비트를 나타내는 8진수 형식으로 지정합니다.

유형: 문자열

길이 제약 조건: 최소 길이는 3입니다. 최대 길이는 4입니다.

패턴: `^[0-7]{3,4}$`

필수 여부: 예

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Destination

복제 구성의 대상 파일 시스템을 설명합니다.

내용

FileSystemId

대상 Amazon EFS 파일 시스템의 ID입니다.

유형: 문자열

길이 제약: 최대 길이는 128입니다.

패턴: `^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$`

필수 여부: 예

Region

대상 파일 시스템이 위치한 AWS 리전입니다.

유형: 문자열

길이 제한: 최소 길이는 1. 최대 길이는 64.

패턴: `^[a-z]{2}-((iso[a-z]{0,1}-)|(gov-)){0,1}[a-z]+-{0,1}[0-9]{0,1}$`

필수 여부: 예

Status

복제 구성의 상태를 설명합니다. 복제 상태에 대한 자세한 내용은 Amazon EFS 사용 설명서의 복제 세부 정보 보기를 참조하세요.

타입: 문자열

유효 값: `ENABLED | ENABLING | DELETING | ERROR | PAUSED | PAUSING`

필수 여부: 예

LastReplicatedTimestamp

대상 파일 시스템에서 가장 최근의 동기화가 성공적으로 완료된 시간입니다. 이 시간 이전에 발생한 소스 파일 시스템의 모든 데이터 변경 사항은 대상 파일 시스템에 성공적으로 복제되었습니다. 이 시간 이후에 발생한 모든 변경 사항은 완전히 복제되지 않을 수 있습니다.

유형: 타임스탬프

필수 여부: 아니요

OwnerId

대상 파일 시스템이 있는 AWS 계정의 ID입니다.

유형: 문자열

길이 제약 조건: 최대 길이는 14입니다.

패턴: `^(\d{12})|(\d{4}-\d{4}-\d{4})$`

필수 여부: 아니요

RoleArn

Amazon EFS가 대신 복제를 수행할 수 있도록 허용하는 소스 계정의 IAM 역할의 Amazon 리소스 이름(ARN)입니다. 이는 동일한 계정 복제의 경우 선택 사항이고 교차 계정 복제에는 필요합니다.

유형: 문자열

길이 제약: 최대 길이 2048.

패턴: `arn:(aws[a-zA-Z-]*)?:iam::\d{12}:role/?[a-zA-Z_0-9+=,.@\-_/]+`

필수 여부: 아니요

StatusMessage

복제 대상 구성의 PAUSED 또는 ERRROR 상태에 대한 세부 정보를 제공하는 메시지입니다. 복제 상태 메시지에 대한 자세한 내용은 Amazon EFS 사용 설명서의 복제 세부 정보 보기를 참조하세요.

유형: 문자열

필수 항목 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS SDK for C++
- AWS SDK for Java V2

- AWS SDK for Ruby V3

DestinationToCreate

복제 구성의 새로운 또는 기존 대상 파일 시스템을 설명합니다.

- 새 파일 시스템으로 복제하려면 대상 파일 시스템의 File System ID를 지정하지 마세요. Amazon EFS는 새롭고 빈 파일 시스템을 생성합니다. One Zone 스토리지를 위해 파일 시스템을 생성할 가용 영역을 지정합니다. 기본 KMS 키 이외의 AWS Key Management Service 키를 사용하려면 지정합니다. 자세한 내용은 Amazon EFS 사용 설명서의 새 Amazon EFS 파일 시스템으로 복제 구성을 참조하세요.

  > **Note**
  >
  > 파일 시스템 생성 후에는 KMS 키나 성능 모드를 변경할 수 없습니다.

- 소스 파일 시스템과 동일한 계정에 있는 기존 파일 시스템으로 복제하려면, 복제 대상 파일 시스템의 ID 또는 Amazon 리소스 이름(ARN)을 제공해야 합니다. 파일 시스템의 복제 덮어쓰기 보호 기능을 비활성화해야 합니다. 자세한 내용은 Amazon EFS 사용 설명서의 기존 파일 시스템에 복제를 참조하세요.
- 파일 시스템을 소스 파일 시스템과 다른 계정에 있는 파일 시스템으로 복제하는 경우(교차 계정 복제) 파일 시스템의 ARN과 Amazon EFS가 대상 계정에서 복제를 수행할 수 있도록 허용하는 IAM 역할을 제공해야 합니다. 파일 시스템의 복제 덮어쓰기 보호 기능을 비활성화해야 합니다. 자세한 내용을 알아보려면 Amazon EFS 사용 설명서의 AWS 계정 간 복제를 참조하세요.

내용

AvailabilityZoneName

One Zone 스토리지를 사용하는 파일 시스템을 생성하려면 대상 파일 시스템을 생성할 가용 영역 이름을 지정합니다.

유형: 문자열

길이 제한: 최소 길이는 1. 최대 길이는 64.

패턴: `.+`

필수 여부: 아니요

FileSystemId

대상에 사용할 파일 시스템의 ID 또는 ARN입니다. 교차 계정 복제의 경우 ARN이어야 합니다. 파일 시스템의 복제 덮어쓰기 보호를 비활성화해야 합니다. ID 또는 ARN을 지정하지 않으면 새 파일 시스템이 생성됩니다.

> **Note**
>
> 기존 파일 시스템에 대한 복제를 처음 구성할 때 Amazon EFS는 원본 파일 시스템의 데이터와 일치하도록 대상 파일 시스템에 데이터를 쓰거나 대상 파일 시스템에서 기존 데이터를 제거합니다. 대상 파일 시스템의 데이터를 변경하지 않으려면 대신 새 파일 시스템에 복제해야 합니다. 자세한 내용은 https://docs.aws.amazon.com/efs/latest/ug/create-replication.html 섹션을 참조하세요.

유형: 문자열

길이 제약: 최대 길이는 128입니다.

패턴: `^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$`

필수 여부: 아니요

KmsKeyId

대상 파일 시스템을 암호화하는 데 사용할 AWS Key Management Service(AWS KMS) 키를 지정합니다. KMS 키를 지정하지 않으면 Amazon EFS에서는 Amazon EFS, `/aws/elasticfilesystem`의 기본 KMS 키를 사용합니다. 이 ID는 다음 형식 중 하나에 해당할 수 있습니다.

- 키 ID - 키의 고유 식별자입니다(예: `1234abcd-12ab-34cd-56ef-1234567890ab`).
- ARN - 키의 ARN입니다. 예: `arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab`.
- 키 별칭 - 키에 대해 이전에 생성된 표시 이름입니다(예: `alias/projectKey1`).
- 키 별칭 ARN - 키 별칭의 ARN입니다(예: `arn:aws:kms:us-west-2:444455556666:alias/projectKey1`).

유형: 문자열

길이 제약: 최대 길이 2048.

패턴: `^([0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}|mrk-[0-9a-f]{32}|alias/[a-zA-Z0-9/_-]+|(arn:aws[-a-z]*:kms:[a-z0-9-]+:\d{12}:((key/[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12})|(key/mrk-[0-9a-f]{32})|(alias/[a-zA-Z0-9/_-]+))))$`

필수 여부: 아니요

Region

리전 스토리지를 사용하는 파일 시스템을 생성하려면 대상 파일 시스템을 생성할 AWS 리전을 지정하세요. 소스 파일 시스템을 소유한 AWS 계정에 대해 리전을 활성화해야 합니다. 자세한 내용은 AWS 일반 참조 안내서의 AWS 리전 관리 섹션을 참조하십시오.

유형: 문자열

길이 제한: 최소 길이는 1. 최대 길이는 64.

패턴: `^[a-z]{2}-((iso[a-z]{0,1}-)|(gov-)){0,1}[a-z]+-{0,1}[0-9]{0,1}$`

필수 여부: 아니요

RoleArn

Amazon EFS가 대신 복제를 수행할 수 있도록 허용하는 소스 계정의 IAM 역할의 Amazon 리소스 이름(ARN)입니다. 이는 동일한 계정 복제의 경우 선택 사항이고 교차 계정 복제에는 필요합니다.

유형: 문자열

길이 제약: 최대 길이 2048.

패턴: `arn:(aws[a-zA-Z-]*)?:iam::\d{12}:role/?[a-zA-Z_0-9+=,.@\-_/]+`

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

FileSystemDescription

파일 시스템 설명 배열입니다.

내용

CreationTime

파일 시스템이 생성된 시간(초)입니다(1970-01-01T00:00:00Z 이후).

유형: 타임스탬프

필수 여부: 예

CreationToken

요청에 지정된 불투명한 문자열입니다.

유형: 문자열

길이 제한: 최소 길이는 1. 최대 길이는 64.

패턴: `.+`

필수 여부: 예

FileSystemId

Amazon EFS에서 할당한 파일 시스템의 ID입니다.

유형: 문자열

길이 제약: 최대 길이는 128입니다.

패턴: `^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$`

필수 여부: 예

LifeCycleState

파일 시스템의 수명 주기 단계입니다.

타입: 문자열

유효 값: `creating | available | updating | deleting | deleted | error`

필수 여부: 예

NumberOfMountTargets

파일 시스템에 있는 탑재 대상의 현재 수. 자세한 내용은 CreateMountTarget 섹션을 참조하세요.

유형: 정수

유효한 범위: 최소값 0.

필수 여부: 예

OwnerId

파일 시스템을 생성한 AWS 계정입니다.

유형: 문자열

길이 제약 조건: 최대 길이는 14입니다.

패턴: `^(\d{12})|(\d{4}-\d{4}-\d{4})$`

필수 여부: 예

PerformanceMode

파일 시스템의 성능 모드입니다.

타입: 문자열

유효 값: `generalPurpose | maxIO`

필수 여부: 예

SizeInBytes

파일 시스템에서 해당 `Value` 필드에 저장된 데이터의 가장 최근 알려진 측정 크기(바이트) 및 해당 `Timestamp` 필드에서 해당 크기가 결정된 시간입니다. `Timestamp` 값은 1970-01-01T00:00:00Z 이후의 정수 시간(초)입니다. 이 `SizeInBytes` 값은 파일 시스템의 일관된 스냅샷 크기를 나타내지는 않지만 파일 시스템에 쓰기가 없는 경우 최종적으로 일관성을 유지합니다. 즉, 몇 시간 이상 파일 시스템을 수정하지 않은 경우에만 `SizeInBytes`가 실제 크기를 나타냅니다. 그렇지 않으면 값이 특정 시점의 파일 시스템 크기와 정확히 일치하지 않습니다.

타입: FileSystemSize객체

필수 여부: 예

Tags

파일 시스템과 연결된 태그로, Tag 객체 배열로 표시됩니다.

유형: Tag객체 어레이

필수 여부: 예

AvailabilityZoneId

파일 시스템이 위치한 가용 영역의 고유하고 일관된 식별자로, One Zone 파일 시스템에만 유효합니다. 예를 들어, use1-az1은 us-east-1 AWS 리전의 가용 영역 ID이고 모든 AWS 계정 계정에서 동일한 위치입니다.

유형: 문자열

필수 여부: 아니요

AvailabilityZoneName

파일 시스템이 위치한 AWS 가용 영역을 설명하며, One Zone 파일 시스템에만 유효합니다. 자세한 내용을 알아보려면 Amazon EFS 사용 설명서의 EFS 스토리지 클래스 사용을 참조하세요.

유형: 문자열

길이 제한: 최소 길이는 1. 최대 길이는 64.

패턴: .+

필수 여부: 아니요

Encrypted

true인 경우 파일 시스템이 암호화되었음을 나타내는 부울 값입니다.

유형: 부울

필수 여부: 아니요

FileSystemArn

EFS 파일 시스템의 Amazon 리소스 이름(ARN)으로서 arn:aws:elasticfilesystem:*region*:*account-id*:file-system/*file-system-id* 형식입니다. 샘플 데이터를 사용한 예: arn:aws:elasticfilesystem:us-west-2:1111333322228888:file-system/fs-01234567

유형: 문자열

필수 여부: 아니요

FileSystemProtection

파일 시스템의 보호를 설명합니다.

유형: FileSystemProtectionDescription객체

필수 여부: 아니요

KmsKeyId

암호화된 파일 시스템을 보호하는 데 사용할 AWS KMS key의 ID입니다.

유형: 문자열

길이 제약: 최대 길이 2048.

패턴: `^([0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}|mrk-[0-9a-f]{32}|alias/[a-zA-Z0-9/_-]+|(arn:aws[-a-z]*:kms:[a-z0-9-]+:\d{12}:((key/[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12})|(key/mrk-[0-9a-f]{32})|(alias/[a-zA-Z0-9/_-]+))))$`

필수 여부: 아니요

Name

태그를 포함하여 파일 시스템에 Name 태그를 추가할 수 있습니다. 자세한 내용은 CreateFileSystem 섹션을 참조하세요. 파일 시스템에 Name 태그가 있는 경우 Amazon EFS는 이 필드에 값을 반환합니다.

유형: 문자열

길이 제약: 최대 길이 256.

패턴: `^([\p{L}\p{Z}\p{N}_.:/=+\-@]*)$`

필수 여부: 아니요

ProvisionedThroughputInMibps

MiBps로 측정되는 프로비저닝된 파일 시스템 처리량을 정의합니다. provisioned로 설정된 ThroughputMode을 사용하는 파일 시스템에 유효합니다.

유형: Double

유효한 범위: 최소값은 1.0입니다.

필수 여부: 아니요

ThroughputMode

파일 시스템의 처리량 모드를 표시합니다. 자세한 내용을 알아보려면 Amazon Elastic File System 사용 설명서의 처리량 모드를 참조하세요.

타입: 문자열

유효 값: `bursting | provisioned | elastic`

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

FileSystemProtectionDescription

파일 시스템의 보호를 설명합니다.

내용

ReplicationOverwriteProtection

파일 시스템의 복제 덮어쓰기 보호 상태입니다.

- ENABLED – 파일 시스템은 복제 구성의 대상 파일 시스템으로 사용할 수 없습니다. 파일 시스템은 쓰기 가능합니다. 복제 덮어쓰기 보호는 기본적으로 ENABLED 상태입니다.
- DISABLED – 파일 시스템은 복제 구성의 대상 파일 시스템으로 사용할 수 있습니다. 파일 시스템은 읽기 전용이며 EFS 복제를 통해서만 수정할 수 있습니다.
- REPLICATING – 파일 시스템은 복제 구성의 대상 파일 시스템으로 사용 중입니다. 파일 시스템은 읽기 전용이며 EFS 복제를 통해서 수정됩니다.

복제 구성을 삭제하면 파일 시스템의 복제 덮어쓰기 보호가 다시 활성화되고 파일 시스템은 쓰기 가능해집니다.

타입: 문자열

유효 값: `ENABLED | DISABLED | REPLICATING`

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

FileSystemSize

파일 시스템에서 해당 `Value` 필드에 저장된 데이터의 가장 최근 알려진 측정 크기(바이트) 및 해당 `Timestamp` 필드에서 해당 크기가 결정된 시간입니다. 이 값은 파일 시스템의 일관된 스냅샷 크기를 나타내지는 않지만 파일 시스템에 쓰기가 없는 경우 최종적으로 일관성을 유지합니다. 즉, 몇 시간 이상 파일 시스템을 수정하지 않은 경우에만 이 값이 실제 크기를 나타냅니다. 그렇지 않으면 이 값이 특정 시점의 파일 시스템 크기와 정확히 일치하지 않을 수 있습니다.

내용

Value

파일 시스템에 저장된 데이터의 가장 최근 알려진 측정 크기(바이트).

타입: Long

유효 범위: 최소값은 0입니다.

필수 여부: 예

Timestamp

`Value` 필드에 반환된 데이터 크기가 결정된 시간입니다. 값은 1970-01-01T00:00:00Z 이후의 정수 시간(초)입니다.

유형: 타임스탬프

필수 여부: 아니요

ValueInArchive

Archive 스토리지 클래스에 저장된 데이터의 가장 최근 알려진 측정 크기(바이트)입니다.

타입: Long

유효 범위: 최소값은 0입니다.

필수 여부: 아니요

ValueInIA

Infrequent Access 스토리지 클래스에 저장된 데이터의 최근 알려진 측정 크기(바이트)입니다.

타입: Long

유효 범위: 최소값은 0입니다.

필수 여부: 아니요

ValueInStandard

Standard 스토리지 클래스에 저장된 데이터의 가장 최근 알려진 측정 크기(바이트)입니다.

타입: Long

유효 범위: 최소값은 0입니다.

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

LifecyclePolicy

스토리지 클래스 안팎으로 파일을 전환할 시기를 지정하는 수명 주기 관리에서 사용하는 정책에 대해 설명합니다. 자세한 내용은 파일 시스템 스토리지 관리 섹션을 참조하십시오.

> **Note**
>
> `put-lifecycle-configuration` CLI 명령 또는 `PutLifecycleConfiguration` API 작업을 사용할 때, Amazon EFS에서는 각 `LifecyclePolicy` 객체에 단일 전환만 포함하도록 요구합니다. 즉, 요청 본문에서 `LifecyclePolicies`는 각 전환에 대해 하나의 객체인 `LifecyclePolicy` 객체의 배열로 구조화되어야 합니다. 자세한 내용은 PutLifecycleConfiguration의 요청 예제를 참조하세요.

내용

TransitionToArchive

파일을 Archive 스토리지로 이동할 기본 스토리지(Standard 스토리지 클래스)에서 마지막으로 액세스한 후 경과한 일수입니다. 디렉터리의 내용 나열과 같은 메타데이터 조작은 파일 액세스 이벤트로 간주되지 않습니다.

타입: 문자열

유효 값: `AFTER_1_DAY | AFTER_7_DAYS | AFTER_14_DAYS | AFTER_30_DAYS | AFTER_60_DAYS | AFTER_90_DAYS | AFTER_180_DAYS | AFTER_270_DAYS | AFTER_365_DAYS`

필수 여부: 아니요

TransitionToIA

Infrequent Access(IA) 스토리지로 이동할 기본 스토리지(Standard 스토리지 클래스)에서 마지막으로 액세스한 후 경과한 일수입니다. 디렉터리의 내용 나열과 같은 메타데이터 조작은 파일 액세스 이벤트로 간주되지 않습니다.

타입: 문자열

유효 값: `AFTER_7_DAYS | AFTER_14_DAYS | AFTER_30_DAYS | AFTER_60_DAYS | AFTER_90_DAYS | AFTER_1_DAY | AFTER_180_DAYS | AFTER_270_DAYS | AFTER_365_DAYS`

필수 여부: 아니요

TransitionToPrimaryStorageClass

IA 또는 Archive 스토리지에서 파일을 액세스한 후 파일을 기본 스토리지로 다시 이동할지 여부입니다. 디렉터리의 내용 나열과 같은 메타데이터 조작은 파일 액세스 이벤트로 간주되지 않습니다.

타입: 문자열

유효 값: `AFTER_1_ACCESS`

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

MountTargetDescription

탑재 대상에 대한 설명을 제공합니다.

내용

FileSystemId

탑재 대상으로 의도된 파일 시스템의 ID입니다.

유형: 문자열

길이 제약: 최대 길이는 128입니다.

패턴: `^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$`

필수 여부: 예

LifeCycleState

탑재 대상의 생명 주기 상태입니다.

타입: 문자열

유효 값: `creating | available | updating | deleting | deleted | error`

필수 여부: 예

MountTargetId

시스템에서 할당한 탑재 대상 ID.

유형: 문자열

길이 제약: 최소 길이는 13입니다. 최대 길이는 45입니다.

패턴: `^fsmt-[0-9a-f]{8,40}$`

필수 여부: 예

SubnetId

탑재 대상의 서브넷의 ID입니다.

유형: 문자열

길이 제약: 최소 길이는 15입니다. 최대 길이는 47입니다.

패턴: `^subnet-[0-9a-f]{8,40}$`

필수 여부: 예

AvailabilityZoneId

탑재 대상이 위치한 가용 영역의 고유하고 일관된 식별자입니다. 예를 들어, `use1-az1`는 us-east-1 리전의 AZ ID이고, 모든 AWS 계정에서 위치가 동일합니다.

유형: 문자열

필수 여부: 아니요

AvailabilityZoneName

탑재 대상이 위치한 가용 영역의 이름입니다. AWS 계정별로 이름에 가용 영역을 독립적으로 매핑합니다. 예를 들어 AWS 계정의 `us-east-1a` 가용 영역은 다른 AWS 계정에 대한 `us-east-1a`와 위치가 동일하지 않을 수 있습니다.

유형: 문자열

길이 제한: 최소 길이는 1. 최대 길이는 64.

패턴: `.+`

필수 여부: 아니요

IpAddress

탑재 대상의 IPv4 주소입니다.

유형: 문자열

길이 제약: 최소 길이는 7입니다. 최대 길이는 15입니다.

패턴: `^[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}$`

필수 여부: 아니요

Ipv6Address

탑재 대상의 IPv6 주소입니다.

유형: 문자열

길이 제약 조건: 최소 길이는 3입니다. 최대 길이 39.

필수 여부: 아니요

NetworkInterfaceId

탑재 대상을 생성할 때 Amazon EFS가 생성한 네트워크 인터페이스의 ID입니다.

유형: 문자열

필수 여부: 아니요

OwnerId

리소스를 소유한 AWS 계정 ID입니다.

유형: 문자열

길이 제약 조건: 최대 길이는 14입니다.

패턴: `^(\d{12})|(\d{4}-\d{4}-\d{4})$`

필수 여부: 아니요

VpcId

탑재 대상이 구성된 Virtual Private Cloud(VPC) ID입니다.

유형: 문자열

필수 항목 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

PosixUser

액세스 포인트를 사용하여 NFS 클라이언트가 수행하는 모든 파일 시스템 작업에 사용되는 액세스 포인트에서 사용자 ID, 그룹 ID 및 보조 그룹 ID를 포함한 전체 POSIX ID입니다.

내용

Gid

이 액세스 포인트를 사용하는 모든 파일 시스템 작업에 사용되는 POSIX 그룹 ID입니다.

타입: Long

유효 범위: 최소값은 0입니다. 최댓값은 4294967295입니다.

필수 여부: 예

Uid

이 액세스 포인트를 사용하는 모든 파일 시스템 작업에 사용되는 POSIX 사용자 ID입니다.

타입: Long

유효 범위: 최소값은 0입니다. 최댓값은 4294967295입니다.

필수 여부: 예

SecondaryGids

이 액세스 포인트를 사용하는 모든 파일 시스템 작업에 사용되는 보조 POSIX 그룹 ID입니다.

유형: Long 배열

배열 구성원: 최소수는 0개입니다. 최대 항목 수는 16개입니다.

유효 범위: 최소값은 0입니다. 최댓값은 4294967295입니다.

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS SDK for C++

- AWS SDK for Java V2
- AWS SDK for Ruby V3

ReplicationConfigurationDescription

특정 파일 시스템의 복제 구성을 설명합니다.

내용

CreationTime

복제 구성이 생성된 시기를 설명합니다.

유형: 타임스탬프

필수 여부: 예

Destinations

대상 객체의 배열입니다. 대상 개체는 하나만 지원됩니다.

타입: Destination객체 배열

필수 여부: 예

OriginalSourceFileSystemArn

복제 구성에 있는 원본 소스 EFS 파일 시스템의 Amazon 리소스 이름(ARN)입니다.

유형: 문자열

필수 항목 여부: 예

SourceFileSystemArn

복제 구성에 있는 현재 소스 파일 시스템의 Amazon 리소스 이름(ARN)입니다.

유형: 문자열

필수 항목 여부: 예

SourceFileSystemId

복제 중인 소스 Amazon EFS 파일 시스템의 ID입니다.

유형: 문자열

길이 제약: 최대 길이는 128입니다.

패턴: `^(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:file-system/fs-[0-9a-f]{8,40}|fs-[0-9a-f]{8,40})$`

필수 여부: 예

SourceFileSystemRegion

소스 EFS 파일 시스템이 위치한 AWS 리전입니다.

유형: 문자열

길이 제한: 최소 길이는 1. 최대 길이는 64.

패턴: `^[a-z]{2}-((iso[a-z]{0,1}-)|(gov-)){0,1}[a-z]+-{0,1}[0-9]{0,1}$`

필수 여부: 예

SourceFileSystemOwnerId

소스 파일 시스템이 있는 AWS 계정의 ID입니다.

유형: 문자열

길이 제약 조건: 최대 길이는 14입니다.

패턴: `^(\d{12})|(\d{4}-\d{4}-\d{4})$`

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ResourceIdPreference

현재 AWS 리전에서 사용자의 AWS 계정의 리소스 유형 및 ID 기본 설정을 설명합니다.

내용

ResourceIdType

LONG_ID(17자) 또는 SHORT_ID(8자)의 EFS 리소스 ID 기본 설정을 식별합니다.

타입: 문자열

유효 값: LONG_ID | SHORT_ID

필수 여부: 아니요

Resources

ID 기본 설정이 적용되는 Amazon EFS 리소스 FILE_SYSTEM 및 MOUNT_TARGET을 식별합니다.

유형: 문자열 배열

유효 값: FILE_SYSTEM | MOUNT_TARGET

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

RootDirectory

액세스 포인트가 액세스를 제공하는 Amazon EFS 파일 시스템의 디렉터리를 지정합니다. 액세스 포인트는 액세스 포인트를 사용하는 애플리케이션에 지정된 파일 시스템 경로를 파일 시스템의 루트 디렉터리로 노출합니다. 액세스 포인트를 사용하는 NFS 클라이언트는 액세스 포인트의 `RootDirectory` 및 하위 디렉터리에 있는 데이터만 액세스할 수 있습니다.

내용

CreationInfo

(선택 사항) 액세스 포인트의 `RootDirectory`에 적용할 POSIX ID 및 권한을 지정합니다. 지정된 `RootDirectory` > `Path`가 없으면 클라이언트가 액세스 포인트에 연결할 때 EFS는 `CreationInfo` 설정을 사용하여 루트 디렉터리를 만듭니다. `CreationInfo`를 지정할 때 모든 속성에 대한 값을 제공해야 합니다.

> **Important**
>
> `CreationInfo`를 지정하지 않고 지정된 `RootDirectory` > `Path`가 없는 경우 액세스 포인트를 사용하여 파일 시스템을 마운트하려는 시도가 실패합니다.

유형: CreationInfo객체

필수 여부: 아니요

Path

EFS 파일 시스템에 액세스하기 위해 액세스 포인트를 사용하여 NFS 클라이언트에 루트 디렉터리로 표시할 EFS 파일 시스템의 경로를 지정합니다. 경로에는 최대 4개의 하위 디렉터리가 있을 수 있습니다. 지정된 경로가 없는 경우 `CreationInfo`를 제공해야 합니다.

유형: 문자열

길이 제약: 최소 길이는 1. 최대 길이는 100.

패턴: `^(\/|(\/(?!\.)+[^$#<>;`|&?{}^*/\n]+){1,4})$`

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Tag

태그는 키-값 쌍입니다. 허용되는 문자는 문자, 공백 및 UTF-8로 표시할 수 있는 숫자와 다음 문자입니다: `+ - = . _ : /`.

내용

Key

태그 키(문자열)입니다. 키는 `aws:`로 시작할 수 없습니다.

유형: 문자열

길이 제약: 최소 길이는 1. 최대 길이 128.

패턴: `^(?![aA]{1}[wW]{1}[sS]{1}:)([\p{L}\p{Z}\p{N}_.:/=+\-@]+)$`

필수 여부: 예

Value

태그 키의 값입니다.

유형: 문자열

길이 제약: 최대 길이 256.

패턴: `^([\p{L}\p{Z}\p{N}_.:/=+\-@]*)$`

필수 여부: 예

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

문서 기록

- API 버전: 2015-02-01
- 최종 설명서 업데이트: 2025년 6월 9일

다음 표는 2018년 7월 이후 Amazon Elastic File System 사용 설명서에서 변경된 중요 사항을 기술한 것입니다. 문서 업데이트에 대한 알림을 받으려면 RSS 피드를 구독하시면 됩니다.

변경 사항	설명	날짜
IPv6 지원	인터넷 프로토콜 버전 6(IPv6)은 Amazon EFS Service API 및 탑재 대상에서 지원됩니다. Amazon EFS API에 대한 자세한 내용은 Amazon EFS API를 참조하세요. 탑재 대상에 대한 자세한 내용은 탑재 대상 관리를 참조하세요.	2025년 6월 9일
특정의 탄력적 처리량 증가 AWS 리전	아시아 태평양(뭄바이), 아시아 태평양(서울), 아시아 태평양(시드니), 유럽(런던) 리전에서 탄력적 처리량을 사용하는 파일 시스템의 최대 읽기 처리량이 20GiBps에서 60GiBps로 증가했습니다. 자세한 내용은 증가할 수 있는 Amazon EFS 할당량을 참조하세요.	2025년 5월 2일
특정의 탄력적 처리량 증가 AWS 리전	특정 리전에서 탄력적 처리량을 사용하는 파일 시스템의 최대 읽기 처리량은 60GiBps입니다. 다른 모든 리전에서는 탄력적 처리량을 사용하는 파일 시스템의 읽기 처리량 제한이 10에서 20GiBps로 증가했습니다.	2025년 4월 10일

	자세한 내용은 증가할 수 있는 Amazon EFS 할당량을 참조하세요.	
액세스 포인트 할당량 증가	단일 파일 시스템이 가질 수 있는 최대 액세스 포인트 수가 1,000개에서 10,000개로 늘어났습니다. 이 한도를 늘리도록 요청할 수도 있습니다. 자세한 내용은 증가할 수 있는 리소스 할당량을 참조하세요.	2025년 2월 10일
백업 및 복원 속도 개선	백업 및 복원 수행 속도가 개선되었습니다. 자세한 내용은 백업 성능을 참조하세요.	2025년 1월 8일
요청에 따른 IOPS 할당량 증가	이제 탄력적 처리량 모드를 사용하여 EFS 파일 시스템의 IOPS 증가를 요청할 수 있습니다. 자세한 내용은 증가할 수 있는 Amazon EFS 할당량을 참조하세요.	2024년 11월 26일
교차 계정 복제 지원	Amazon EFS는 교차 계정 복제를 지원합니다. 자세한 내용은 AWS 계정간에 EFS 파일 시스템 복제를 참조하세요.	2024년 11월 19일
기존 AWS 관리형 정책 업데이트	`ReplicationRead` , `ReplicationWrite` , `iam:PassRole` 이 `AmazonElasticFileSystemFullAccess` 정책에 추가됩니다. 자세한 내용은 AmazonElasticFileSystemFullAccess를 참조하세요.	2024년 11월 7일

기존 AWS 관리형 정책 업데이트	`ReplicationRead` 및 `ReplicationWrite` 가 `AmazonElasticFileSystemServiceRolePolicy` 에 추가되었습니다. 자세한 내용은 AmazonElasticFileSystemServiceRolePolicy를 참조하세요.	2024년 11월 7일
기존 AWS 관리형 정책 업데이트	기존 `AmazonElasticFileSystemReadOnlyAccess` 정책에 권한 `ReplicationRead` 가 추가되었습니다. 자세한 내용은 AmazonElasticFileSystemReadOnlyAccess를 참조하세요.	2024년 11월 7일
탄력적 처리량 한도 증가	탄력적 처리량 한도가 특정 리전의 경우 60GiBps(초당 기비바이트) AWS 리전 로 증가하고, 다른 모든 리전의 경우 10GiBps로 증가했습니다. 자세한 내용은 각 AWS 리전에 연결된 모든 클라이언트에 대한 총 기본 탄력적 처리량 단원을 참조하세요.	2024년 10월 14일
기존 AWS 관리형 정책 업데이트	이제 선택적 `Sid`(문 ID) 요소가 `AmazonElasticFileSystemReadOnlyAccess` 정책 설명에 포함됩니다. `Sid` 값은 `ElasticFileSystemReadOnlyAccess` 입니다. `Sid` 정책 요소에 대한 자세한 내용은 IAM JSON 정책 요소: Sid를 참조하세요.	2024년 8월 7일

탄력적 처리량 한도 증가	특정 AWS 리전의 탄력적 처리량 한도가 증가했습니다. 자세한 내용은 각 AWS 리전에 연결된 모든 클라이언트에 대한 총 기본 탄력적 처리량 단원을 참조하세요.	2024년 7월 31일
탑재 대상 할당량 증가	각 가상 프라이빗 클라우드(VPC)의 최대 탑재 대상 수가 400개에서 1,400개로 증가했습니다. 자세한 내용은 변경할 수 없는 Amazon EFS 리소스 할당량을 참조하세요.	2024년 5월 15일
Elastic File System의 통합 처리량 한도 증가	탄력적 처리량을 사용하고 버전 2.0 이상의 Amazon EFS 클라이언트(amazon-efs-utils 버전) 또는 Amazon EFS CSI 드라이버(aws-efs-csi-driver)를 사용하여 탑재된 파일 시스템의 경우 최대 읽기 및 쓰기 결합 처리량은 1,500MiBps입니다. 자세한 내용은 Amazon EFS 성능의 성능 요약 표를 참조하세요.	2024년 4월 30일
탄력적 처리량 한도 증가	특정 AWS 리전의 탄력적 처리량 한도가 증가했습니다. 자세한 내용은 각 AWS 리전에 연결된 모든 클라이언트에 대한 총 기본 탄력적 처리량 단원을 참조하세요.	2024년 3월 13일

IOPS 증가됨	탄력적 처리량을 사용하는 파일 시스템은 자주 액세스하지 않는 데이터에 대해 최대 90,000회의 읽기를 수행할 수 있습니다. 자세한 내용은 성능 개요를 참조하세요.	2024년 1월 22일
기존 AWS 관리형 정책 업데이트	기존 AmazonElasticFileSystemFullAccess 정책에 `elasticfilesystem:UpdateFileSystemProtection` 권한이 추가되어 보안 주체가 파일 시스템의 보호를 업데이트할 수 있도록 합니다. 자세한 내용은 AWS 관리형 정책에 대한 Amazon EFS 업데이트를 참조하세요.	2023년 11월 27일
기존 파일 시스템에 복제	이제 파일 시스템을 기존 파일 시스템에 복제할 수 있으므로 페일백 목적으로 파일 시스템 간에 변경 내용을 쉽게 동기화할 수 있습니다. 자세한 내용은 대상 파일 시스템 섹션을 참조하십시오.	2023년 11월 27일
파일 시스템 보호 추가됨	파일 시스템에 복제 덮어쓰기 보호가 추가되었으며 기본적으로 활성화되어 있습니다. 보호 기능은 파일 시스템이 복제 구성의 대상으로 사용되는 것을 방지합니다. 자세한 내용은 파일 시스템 보호 섹션을 참조하십시오.	2023년 11월 27일

새 스토리지 클래스, 파일 시스템 유형 및 수명 주기 정책	Amazon EFS는 이제 EFS Archive 스토리지 클래스, 파일 시스템 유형 및 Archive로 전환 수명 주기 정책을 제공합니다. 자세한 내용은 파일 시스템 유형 및 스토리지 클래스 섹션을 참조하십시오.	2023년 11월 26일
IOPS 증가됨	탄력적 처리량 파일 시스템은 이제 자주 액세스하지 않는 데이터에 대해 최대 65,000개의 읽기 및 50,000개의 쓰기 작업 IOPS를 지원하며, 이제 자주 액세스하는 데이터에 대해 250,000개의 읽기 IOPS를 지원합니다. 자세한 내용은 성능 개요를 참조하세요.	2023년 11월 26일
소스 파일 시스템에서 복제 구성을 삭제합니다.	이제 소스 파일 시스템에서 복제 구성을 삭제할 수 있습니다. 자세한 내용은 복제 구성 삭제를 참조하세요.	2023년 9월 19일
추가 AWS 리전 지원 추가	이제 이스라엘(텔아비브) 리전에서 모든 사용자가 Amazon EFS를 사용할 수 있습니다.	2023년 8월 7일
범용 모드 파일 시스템의 성능 향상	Amazon EFS 범용 모드 파일 시스템은 이제 초당 최대 55,000개의 읽기 작업과 25,000개의 쓰기 작업을 지원합니다. 자세한 내용은 Amazon EFS 파일 시스템 할당량 섹션을 참조하세요.	2023년 8월 3일

프로비저닝된 처리량 한도 증가	특정에 대해 프로비저닝된 처리량 한도가 증가했습니다 AWS 리전. 자세한 내용은 각의 연결된 모든 클라이언트에 대한 총 기본 프로비저닝 처리량 AWS 리전을 참조하세요.	2023년 6월 21일
EFS 복제를 위한 확장된 리전 지원	이제 EFS를 사용할 수 있는 모든 AWS 리전 에서 EFS 복제를 사용할 수 있습니다. 자세한 내용은 Amazon EFS 복제를 참조하세요.	2023년 4월 28일
탄력적 처리량 한도 증가	특정 AWS 리전의 탄력적 처리량 한도가 증가했습니다. 자세한 내용은 각의 연결된 모든 클라이언트에 대한 총 기본 탄력적 처리량 표를 참조하세요 AWS 리전.	2023년 4월 17일
기본 처리량 모드가 버스팅에서 탄력적으로 변경됨	파일 시스템의 기본(및 권장) 처리량 모드는 이제 버스팅 대신 탄력적입니다. 자세한 내용은 처리량 모드 섹션을 참조하세요.	2023년 4월 13일
추가 AWS 리전 지원 추가	이제 아시아 태평양(멜버른) 리전에서 모든 사용자가 Amazon EFS를 사용할 수 있습니다.	2023년 8월 12일
macOS Ventura에 대한 지원 추가	이제 macOS Ventura에서 실행되는 EC2 Mac 인스턴스에 Amazon EFS를 설치할 수 있습니다. 자세한 내용은 지원되는 배포를 참조하세요.	2023년 4월 10일

추가 AWS 리전 지원 추가	이제 아시아 태평양(하이데라바드) 리전에서 모든 사용자가 Amazon EFS를 사용할 수 있습니다.	2023년 2월 16일
추가 AWS 리전 지원 추가	이제 유럽(스페인) AWS 리전에서 모든 사용자가 Amazon EFS를 사용할 수 있습니다.	2023년 1월 19일
파일 시스템의 액세스 포인트 제한 증가	단일 파일 시스템이 가질 수 있는 최대 액세스 포인트 수가 120개에서 1,000개로 늘어났습니다. 자세한 내용은 리소스 할당량을 참조하세요.	2023년 1월 17일
추가 AWS 리전 지원 추가	이제 유럽(취리히)의 모든 사용자가 Amazon EFS를 사용할 수 있습니다 AWS 리전.	2022년 12월 15일
1일 수명 주기 정책에 대한 지원 추가	이제 IA 수명 주기 정책으로 전환할 1일을 선택할 수 있습니다. 자세한 내용은 수명 주기 정책 사용을 참조하세요.	2022년 11월 27일
읽기 및 쓰기 지연 시간 감소	One Zone 스토리지와 Standard 스토리지 파일 시스템 모두에서 파일 데이터 읽기 및 쓰기 지연 시간이 단축되었습니다. 자세한 내용은 성능 개요를 참조하세요.	2022년 11월 27일
추가 처리량 모드 추가	탄력적 처리량 모드가 Amazon EFS 파일 시스템의 처리량 옵션으로 추가되었습니다. 자세한 내용은 탄력적 처리량 단원을 참조하세요.	2022년 11월 27일

추가 AWS 리전 지원 추가	중동(UAE) 리전에서 이제 Amazon EFS를 사용할 수 있습니다.	2022년 10월 17일
EFS 복제에 대한 지원이 추가됨	Amazon EFS 는 EFS 복제가 소켓 및 명명된 파이프 또는 FIFOs.	2022년 9월 15일
연결당 파일 잠금 수 제한이 늘어났습니다.	연결당 파일 잠금 수가 8192개에서 65,536개로 증가했습니다. 자세한 내용은 NFS 클라이언트 할당량을 참조하세요.	2022년 5월 4일
파일 잠금을 사용하는 프로세스에 대한 제한이 제거되었습니다.	Amazon EFS 는 단일 인스턴스에서 최대 256개의 프로세스가 파일 잠금을 동시에 사용할 수 있는 이전 제한을 제거했습니다. 자세한 내용은 NFS 클라이언트 할당량을 참조하세요.	2022년 5월 4일
추가 AWS 리전 지원 추가	이제 아시아 태평양(자카르타) AWS 리전에서 Amazon EFS를 사용할 수 있습니다.	2022년 1월 27일
EFS 복제에 대한 지원이 추가됨	EFS 복제를 사용하여 EFS 파일 시스템의 데이터 및 메타데이터를 선택한 AWS 리전 의 다른 EFS 파일 시스템에 복제합니다. 자세한 내용은 Amazon EFS 복제를 참조하세요.	2022년 1월 25일
파일 시스템 및 탑재 대상 리소스는 17자 리소스 ID 형식을 사용합니다.	이제 새 Amazon EFS 파일 시스템 및 탑재 대상 리소스에 17자 ID가 할당됩니다. 자세한 내용은 Amazon EFS 리소스 작업을 참조하세요.	2021년 10월 22일

EFS Intelligent-Tiering에 대한 지원 추가	EFS Intelligent-Tiering은 EFS 수명 주기 관리를 사용하여 파일 액세스 패턴을 모니터링하며, 해당 Infrequent Access(IA) 스토리지 클래스 간에 파일을 자동으로 전환하도록 설계되었습니다. 자세한 내용을 알아보려면 EFS Intelligent-Tiering 및 EFS 수명 주기 관리를 참조하세요.	2021년 9월 2일
17자 리소스 ID 형식 테스트에 대한 지원이 추가되었습니다.	Amazon EFS는 2021년 10월 1일에 파일 시스템 및 탑재 대상에 대해 8자 ID를 사용하던 것을 17자 ID로 전환합니다. 이 전환 중에 옵트인하여 17자 리소스 IDs를 AWS 리전 기준으로 사용할 수 있습니다. 자세한 내용은 리소스 ID를 참조하세요.	2021년 5월 5일
Amazon EFS 탑재 도우미를 사용하여 다른 가용 영역에서 One Zone 파일 시스템을 탑재하기 위한 지원이 추가되었습니다.	이제 EFS 탑재 도우미를 사용하여 One Zone 스토리지 클래스를 사용하는 Amazon EFS 파일 시스템을 다른 가용 영역에 있는 EC2 인스턴스에 마운트할 수 있습니다. 새 az 옵션을 사용하여 Amazon EFS 파일 시스템의 가용 영역을 지정할 수 있습니다. 자세한 내용은 One Zone 스토리지 클래스를 사용한 파일 시스템 탑재를 참조하세요.	2021년 4월 6일

EFS One Zone 스토리지 클래스에 대한 지원 추가	Amazon EFS One Zone 스토리지 클래스는 AWS 리전의 단일 가용 영역 내에 데이터를 중복 저장합니다. EFS One Zone 및 One Zone-Infrequent Access(One Zone-IA) 스토리지 클래스는 EFS Standard 및 Standard-IA 스토리지 클래스의 다중 AZ 복원력이 필요하지 않은 데이터를 저장하기 위한 비용 효율적인 옵션입니다. 자세한 내용은 EFS 스토리지 클래스 사용 섹션을 참조하세요.	2021년 3월 9일
추가 AWS 리전 지원 추가	이제 아시아 태평양(오사카) AWS 리전에서 Amazon EFS를 사용할 수 있습니다.	2021년 3월 3일
macOS Big Sur를 실행하는 Amazon EC2 macOS 인스턴스에 대한 지원이 추가되었습니다.	이제 EFS 탑재 도우미를 사용하거나 NFS 탑재 명령을 사용하여 macOS Big Sur를 실행하는 EC2 macOS 인스턴스에서 Amazon EFS 파일 시스템을 탑재할 수 있습니다. 자세한 내용은 EFS 탑재 도우미를 사용한 탑재 또는 EFS 탑재 도우미를 사용하지 않고 파일 시스템 탑재를 참조하세요.	2021년 2월 23일
AWS GovCloud (US) 리전에서 새로운 Amazon EFS 콘솔 사용 가능	이제에서 새 Amazon EFS 콘솔을 사용할 수 있습니다 AWS GovCloud (US) AWS 리전.	2021년 2월 10일

새로운 Amazon EFS CloudWatch 지표 MeteredIOBytes 에 대한 지원이 추가되었습니다.	MeteredIOBytes 를 사용하여 각 파일 시스템 작업(데이터 읽기, 쓰기 및 메타데이터 작업 포함)에 필요한 측정된 바이트 수를 측정할 수 있습니다. 읽기 작업은 다른 작업의 3분의 1 비율로 측정됩니다. 자세한 내용은 Amazon EFS에 대한 Amazon CloudWatch 지표를 참조하세요.	2021년 1월 28일
Amazon EFS는 파일 시스템 읽기 처리량을 300% 증가시킵니다.	Amazon EFS 파일 시스템은 이제 다른 요청의 3분의 1 속도로 읽기 요청을 측정합니다.	2021년 1월 28일
새로운 Amazon EFS CloudWatch 지표 StorageBytes 에 대한 지원이 추가되었습니다.	StorageBytes 를 사용하여 Standard 및 Infrequent Access 스토리지 클래스에 저장된 데이터의 양을 포함하여 파일 시스템의 크기(바이트)를 측정하고 모니터링할 수 있습니다. 자세한 내용은 Amazon EFS에 대한 Amazon CloudWatch 지표를 참조하세요.	2021년 1월 11일
Amazon EFS 파일 시스템에 액세스하는 AWS Transfer Family 데 사용	AWS Transfer Family 를 사용하여 Amazon EFS 파일 시스템으로 또는 Amazon EFS 파일 시스템에서 파일을 전송할 수 있습니다. 자세한 내용은 EFS 파일 시스템의 파일에 액세스하는 AWS Transfer Family 데 사용을 참조하세요.	2021년 1월 6일

AWS Systems Manager 를 사용하여 Amazon EFS 클라이언트 관리(amazon-efs-utils)	AWS Systems Manager 를 사용하여 EC2 인스턴스에 Amazon EFS 클라이언트 (amazon-efs-utils)를 자동으로 설치하거나 업데이트 할 수 있습니다. 자세한 내용은 AWS Systems Manager를 사용하여 Amazon EFS 클라이언트 자동 설치 또는 업데이트를 참조하세요.	2020년 9월 29일
암호화된 EFS 파일 시스템 생성 적용	elasticfilesystem: Encrypted AWS Identity and Access Management (IAM) 조건 키를 사용하여 사용자가 유휴 시 암호화된 Amazon EFS 파일 시스템을 생성하도록 할 수 있습니다. 자세한 내용은 연습: 저장 시 암호화된 Amazon EFS 파일 시스템을 생성하도록 강제를 참조하세요.	2020년 9월 16일
Amazon EFS의 클라이언트당 처리량 100% 증가	EFS는 이제 클라이언트당 처리량을 최대 500MB/s까지 지원하며, 이는 이전 제한인 250MB/s에서 100% 증가한 수치입니다. 자세한 내용은 Amazon EFS 파일 시스템 할당량을 참조하세요.	2020년 7월 23일
Amazon EFS 파일 시스템의 자동 일일 백업에 대한 지원이 추가되었습니다.	이제 EFS 콘솔을 사용하여 파일 시스템을 생성할 때 자동 일일 백업이 기본적으로 활성화됩니다. 자세한 내용은 Amazon EFS AWS Backup 에서 사용을 참조하세요.	2020년 7월 16일

새로운 빠른 생성 워크플로는 Amazon EFS 파일 시스템 생성을 간소화합니다.	EFS 콘솔의 빠른 생성 옵션을 사용하면 버튼 하나로 서비스 권장 설정을 사용하여 EFS 파일 시스템을 생성할 수 있습니다. 자세한 내용은 EFS 파일 시스템 생성을 참조하세요.	2020년 7월 16일
이제 새 Amazon EFS 콘솔을 사용할 수 있습니다.	새로운 EFS 콘솔을 사용하면 Amazon EFS를 더 쉽게 사용하고 EFS 파일 시스템 관리를 간소화할 수 있습니다.	2020년 7월 16일
Amazon EFS는 파일 시스템 최소 처리량을 증가시킵니다.	버스팅 처리량을 사용하는 Amazon EFS 파일 시스템의 최소 처리량은 이제 1MiB/s입니다. 자세한 내용은 처리량 모드 섹션을 참조하세요.	2020년 6월 30일
범용 모드 파일 시스템의 성능 향상	Amazon EFS 범용 모드 파일 시스템이 앞서 7,000회로 제한되었던 읽기 작업을 400% 증가하여 초당 최대 35,000회를 지원하게 됩니다. 자세한 내용은 Amazon EFS 파일 시스템 할당량 섹션을 참조하세요.	2020년 4월 1일
추가 AWS 리전 지원 추가	이제 베이징 및 닝샤의 모든 사용자가 Amazon EFS를 사용할 수 있습니다 AWS 리전.	2020년 1월 22일

NFS 클라이언트에 대한 IAM 권한 부여 지원 추가	이제 AWS Identity and Access Management (IAM)을 사용하여 Amazon EFS 파일 시스템에 대한 NFS 액세스를 관리할 수 있습니다. 자세한 내용은 AWS IAM을 사용하여 Amazon EFS에 대한 NFS 액세스 제어를 참조하세요.	2020년 1월 13일
EFS 액세스 포인트에 대한 지원 추가	Amazon EFS 액세스 포인트는 Amazon EFS 파일 시스템에 대한 애플리케이션별 진입점으로, 공유 데이터 세트에 대한 애플리케이션 액세스를 쉽게 관리할 수 있도록 합니다. 자세한 내용은, Amazon EFS 액세스 포인트로 작업하기를 참조하세요.	2020년 1월 13일
AWS Backup 부분 복원에 대한 지원이 추가되었습니다.	이제 전체 복구 시점을 복원하는 것 외에도 부분 복원을 사용하여 특정 파일 및 디렉터리를 복원할 수 있습니다. 자세한 내용은 Amazon EFS AWS Backup 에서 사용을 참조하세요.	2020년 1월 13일
IAM 서비스 연결 역할에 대한 지원 추가	Amazon EFS에서는 이제 IAM에 기반한 서비스 연결 역할을 사용하므로 필요한 권한을 자동으로 추가하여 EFS를 보다 쉽게 설정할 수 있습니다. 자세한 내용은 Amazon EFS에 서비스 연결 역할 사용을 참조하세요.	2019년 12월 10일

추가 AWS 리전 지원 추가	이제 유럽(스톡홀름)의 모든 사용자가 Amazon EFS를 사용할 수 있습니다 AWS 리전.	2019년 11월 20일
추가 AWS 리전 지원 추가	이제 아시아 태평양(홍콩)의 모든 사용자가 Amazon EFS를 사용할 수 있습니다 AWS 리전.	2019년 11월 20일
추가 AWS 리전 지원 추가	이제 남아메리카(상파울루)의 모든 사용자가 Amazon EFS를 사용할 수 있습니다 AWS 리전.	2019년 11월 20일
추가 AWS 리전 지원 추가	이제 중동(바레인)의 모든 사용자가 Amazon EFS를 사용할 수 있습니다 AWS 리전.	2019년 11월 20일
새로운 7일 수명 주기 관리 정책이 추가됨	이제 수명 주기 관리에 7일 후 데이터를 비용 효율적인 Infrequent Access 스토리지 클래스로 이동하는 추가 정책이 포함되었습니다. 자세한 내용은 EFS 수명 주기 관리를 참조하세요.	2019년 11월 6일
인터페이스 VPC 엔드포인트에 대한 지원이 추가됨	VPC(Virtual Private Cloud)와 Amazon EFS 간에 프라이빗 연결을 설정하여 EFS API를 호출할 수 있습니다. 자세한 내용은 VPC 엔드포인트 작업을 참조하세요.	2019년 10월 22일

새 EC2 인스턴스를 시작할 때 EFS 파일 시스템을 탑재합니다.	이제 EC2 Launch Instance Wizard에서 시작 시 EFS 파일 시스템을 탑재하도록 새 Amazon EC2 인스턴스를 구성할 수 있습니다. 자세한 내용은 2단계를 참조하세요. EC2 리소스를 만들고 EC2 인스턴스를 시작합니다.	2019년 10월 17일
Service Quotas 지원 추가	이제 Service Quotas 콘솔에서 모든 Amazon EFS 한도를 볼 수 있습니다. 자세한 내용은 Amazon EFS 한도를 참조하세요.	2019년 9월 10일
새로운 수명 주기 관리 정책 추가	수명 주기 관리를 사용할 때 이제 4개 수명 주기 정책 중 하나를 선택하여 파일이 비용 효율적인 Infrequent Access 스토리지 클래스로 전환되는 시기를 정의할 수 있습니다. 자세한 내용은 EFS 수명 주기 관리를 참조하세요.	2019년 7월 9일
이제 모든 EFS 파일 시스템에서 EFS 수명 주기 관리 사용 가능	이제 모든 EFS 파일 시스템에서 EFS 수명 주기 관리 기능을 사용할 수 있습니다. 파일 시스템이 생성되는 시기에 따른 이전 제한은 이제 제거됩니다. 자세한 내용은 EFS 수명 주기 관리를 참조하세요.	2019년 7월 9일
추가 AWS 리전 지원 추가	이제 유럽(파리)의 모든 사용자가 Amazon EFS를 사용할 수 있습니다 AWS 리전.	2019년 6월 12일

추가 AWS 리전 지원 추가	이제 아시아 태평양(뭄바이)의 모든 사용자가 Amazon EFS를 사용할 수 있습니다 AWS 리전.	2019년 6월 5일
추가 AWS 리전 지원 추가	이제 캐나다(중부)의 모든 사용자가 Amazon EFS를 사용할 수 있습니다 AWS 리전.	2019년 5월 1일
API 업데이트: 이제 CreateFileSystem 작업 페이로드에 태그가 포함됩니다.	이제 AWS API 및 CLI CreateFileSystem 작업을 사용하여 Amazon EFS 파일 시스템을 생성할 때 태그를 포함할 수 있습니다. 자세한 내용은 CreateFileSystem 및 AWS CLI를 사용하여 파일 시스템 생성을 참조하세요.	2019년 2월 19일
새로운 기능: EFS Infrequent Access 스토리지 클래스 및 EFS 수명 주기 관리	Amazon EFS Infrequent Access는 자주 액세스하지 않는 파일에 대해 비용 최적화된 스토리지 클래스입니다. EFS 수명 주기 관리는 Standard 스토리지에서 Infrequent Access 스토리지로 파일을 자동 이전합니다. 자세한 내용은 EFS 스토리지 클래스 섹션을 참조하세요.	2019년 2월 13일
추가 AWS 리전 지원 추가	이제 유럽(런던)의 모든 사용자가 Amazon EFS를 사용할 수 있습니다 AWS 리전.	2019년 1월 23일

AWS Backup Amazon EFS와의 서비스 통합	Amazon EFS 파일 시스템은 클라우드 및 온프레미스의 서비스 전반에 걸쳐 데이터를 백업하기 위한 완전 관리형 중앙 집중식 자동 백업 AWS 서비스 AWS Backup인를 사용하여 백업할 수 있습니다. 자세한 내용은 AWS Backup 및 Amazon EFS를 참조하세요.	2019년 1월 16일
온프레미스 스토리지 시스템에 대한 전송 게이트웨이 연결 지원이 추가되었습니다.	Amazon EFS 파일 시스템은 이제 온프레미스 스토리지 시스템에 대한 전송 게이트웨이 연결을 통해 액세스할 수 있습니다. 자세한 내용은 다른 계정 또는 VPC에서 탑재 및 연습: 다른 VPC를 사용해 파일 시스템 탑재 단원을 참조하세요.	2018년 12월 6일
EFS 파일 동기화는 이제 새 AWS DataSync 서비스의 일부입니다.	AWS DataSync 는 온프레미스 스토리지 시스템과 AWS 스토리지 서비스 간의 대량 데이터 동기화를 간소화하는 관리형 데이터 전송 서비스입니다. 자세한 내용은 를 사용하여 온프레미스 파일 시스템에서 Amazon EFS로 파일 전송 AWS DataSync을 참조하세요.	2018년 11월 26일
VPN 및 리전 간 VPC 피어링 연결 지원이 추가됨	Amazon EFS에서 이제 VPN 연결 및 리전 간 VPC 피어링 연결을 통해 액세스 가능합니다. 자세한 내용은 를 사용하여 온프레미스 파일 시스템에서 Amazon EFS로 파일 전송 AWS DataSync을 참조하세요.	2018년 10월 23일

VPN 및 리전 간 VPC 피어링 연결 지원이 추가됨	Amazon EFS 파일 시스템은 이제 VPN 연결 및 리전 간 VPC 피어링 연결을 통해 액세스 가능합니다. 자세한 내용은 다른 계정 또는 VPC에서 탑재 및 Amazon EFS에서 Direct Connect 및 VPN을 사용하는 방식 단원을 참조하세요.	2018년 10월 23일
추가 AWS 리전 지원 추가	이제 아시아 태평양(싱가포르) AWS 리전에서 모든 사용자가 Amazon EFS를 사용할 수 있습니다.	2018년 7월 13일
프로비저닝 처리량 모드 도입	이제 새로운 프로비저닝 처리량 모드를 사용하여 새로운 파일 시스템 또는 기존 파일 시스템에 처리량을 프로비저닝할 수 있습니다. 자세한 내용은 처리량 모드 섹션을 참조하세요.	2018년 7월 12일
추가 AWS 리전 지원 추가	이제 아시아 태평양(도쿄) AWS 리전에서 Amazon EFS를 사용할 수 있습니다.	2018년 7월 11일

다음 표는 2018년 7월 이전에 Amazon Elastic File System 사용 설명서에서 변경된 중요 사항을 기술한 것입니다.

변경	설명	변경 날짜
추가 AWS 리전 지원 추가	이제 아시아 태평양(서울) AWS 에서 모든 사용자가 Amazon EFS를 사용할 수 있습니다.	2018년 5월 30일
CloudWatch 지표 수식 지원 추가	지표 수식을 사용하면 여러 CloudWatch 지표를 쿼리하고 수식 표현식을 활용함으로써 이러한 지표를 기반으로 새로운 시계열을 만들 수 있습니다. 자세한 내용은	2018년 4월 4일

변경	설명	변경 날짜
	CloudWatch 지표와 함께 지표 수식 사용 단원을 참조하십시오.	
amazon-efs-utils 오픈 소스 도구 세트와 전송 중 암호화 추가	amazon-efs-utils 도구는 탑재 같은 Amazon EFS 사용을 간소화시키는 오픈 소스 실행 파일 세트입니다. 추가 요금 없이 amazon-efs-utils 를 사용할 수 있으며, GitHub에서 도구를 다운로드할 수 있습니다. 자세한 내용은 Amazon EFS 클라이언트 수동 설치 단원을 참조하십시오. 이 릴리스에서는 Amazon EFS가 전송 계층 보안(TLS) 터널링을 통해 전송 중 암호화를 지원합니다. 자세한 내용은 Amazon EFS의 데이터 암호화 단원을 참조하십시오.	2018년 4월 4일
에 따라 파일 시스템 제한이 업데이트됨 AWS 리전	Amazon EFS가 모든 AWS 리전의 전체 계정에서 파일 시스템 수의 제한을 증가시켰습니다. 자세한 내용은 변경할 수 없는 Amazon EFS 리소스 할당량 단원을 참조하십시오.	2018년 3월 15일
추가 AWS 리전 지원 추가	이제 미국 서부(캘리포니아 북부)의 모든 사용자가 Amazon EFS를 사용할 수 있습니다 AWS 리전.	2018년 3월 14일
유휴 시(저장된) 데이터 암호화	이제 Amazon EFS에서 저장된 데이터 암호화를 지원합니다. 자세한 내용은 Amazon EFS의 데이터 암호화 단원을 참조하십시오.	2017년 8월 14일
추가 리전 지원 추가	이제 유럽(프랑크푸르트) 리전에서 모든 사용자가 Amazon EFS를 사용할 수 있습니다.	2017년 7월 20일
도메인 이름 시스템(DNS)을 사용하는 파일 시스템 이름	Amazon EFS에서는 이제 파일 시스템에 대해 DNS 이름을 지원합니다. 파일 시스템의 DNS 이름은 연결 중인 Amazon EC2 인스턴스의 가용 영역에서 탑재 대상의 IP 주소를 자동으로 확인합니다. 자세한 내용은 DNS 이름을 사용하여 Amazon EC2에 탑재 단원을 참조하십시오.	2016년 12월 20일

변경	설명	변경 날짜
파일 시스템에 대한 태그 지원 증가	Amazon EFS에서는 현재 파일 시스템당 태그를 50개 지원합니다. Amazon EFS의 태그에 대한 자세한 내용은 EFS 리소스에 태그 지정 섹션을 참조하세요.	2016년 8월 29일
정식 출시	Amazon EFS가 이제 미국 동부(버지니아 북부), 미국 서부(오레곤) 및 유럽(아일랜드) 리전에서 모든 사용자에게 정식으로 출시되었습니다.	2016년 6월 28일
파일 시스템 제한 증가	각 AWS 리전 에 대한 계정별로 만들 수 있는 Amazon EFS 파일 시스템의 수가 5개에서 10개로 늘었습니다.	2015년 8월 21일
업데이트된 시작하기 연습	연습 시작하기가 업데이트되어 시작하기 과정이 간소화되었습니다.	2015년 8월 17일
새 안내서	이 설명서는 Amazon Elastic File System 사용 설명서의 최초 릴리스입니다.	2015년 5월 26일

기계 번역으로 제공되는 번역입니다. 제공된 번역과 원본 영어의 내용이 상충하는 경우에는 영어 버전이 우선합니다.