



\*\*\*Unable to locate subtitle\*\*\*

# AWS Snowball Edge 개발자 안내서



# AWS Snowball Edge 개발자 안내서: \*\*\*Unable to locate subtitle\*\*\*

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

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

# Table of Contents

Snowball Edge란 무엇인가요? .....	1
Snowball Edge 기능 .....	1
관련 서비스 .....	2
Snowball Edge 서비스에 액세스 .....	3
AWS Snowball Edge 디바이스 액세스 .....	3
Snowball Edge 요금 .....	3
AWS Snowball Edge 모니터링 .....	4
최초 AWS Snowball Edge 사용자를 위한 리소스 .....	4
디바이스 하드웨어 정보 .....	4
디바이스 구성 .....	5
디바이스 사양 .....	6
지원되는 네트워크 하드웨어 .....	10
Snowball Edge를 사용하기 위한 사전 조건 .....	13
에 가입 AWS 계정 .....	14
관리자 액세스 권한이 있는 사용자 생성 .....	14
환경 정보 .....	15
특수 문자 사용 .....	16
를 사용한 Amazon S3 암호화 AWS KMS .....	17
서버 측 암호화를 통한 Amazon S3 암호화 .....	21
가져오기 및 내보내기 작업에 Snowball Edge에서 Amazon S3 어댑터를 사용하기 위한 사전 조건 .....	22
Snowball Edge에서 Amazon S3 호환 스토리지를 사용하기 위한 사전 조건 .....	22
Snowball Edge에서 컴퓨팅 인스턴스를 사용하기 위한 사전 조건 .....	23
Snowball Edge 작동 방식 .....	25
가져오기 작업 작동 방식 .....	27
내보내기 작업 작동 방식 .....	27
로컬 컴퓨팅 및 스토리지 작동 방식 .....	28
클러스터링된 로컬 컴퓨팅 및 스토리지 작업 작동 방식 .....	29
Snowball Edge 동영상 및 블로그 .....	29
Snowball Edge 디바이스의 장기 요금 .....	31
장기 요금제 기간 동안의 디바이스 교체 .....	31
배송 고려 사항 .....	33
리전별 배송 제한 .....	33
시작 .....	35

Snowball Edge 디바이스 주문 작업 생성 .....	36
작업 유형 선택 .....	36
컴퓨팅 및 스토리지 옵션 선택 .....	37
기능 및 옵션 선택 .....	41
보안, 배송 및 알림 기본 설정 선택 .....	42
작업 요약 검토 및 작업 생성 .....	45
작업 취소 .....	46
Snowball Edge 주문 작업 복제 .....	46
Snowball Edge 받기 .....	47
로컬 네트워크에 연결 .....	49
Snowball Edge에 액세스하기 위한 자격 증명 가져오기 .....	50
Snowball Edge 잠금 해제 .....	51
Snowball Edge 잠금 해제 문제 해결 .....	54
로컬 사용자 설정 .....	54
Snowball Edge 디바이스 재부팅 .....	56
Snowball Edge 전원 끄기 .....	61
디바이스 반환 .....	65
반품 배송 .....	66
배송 업체 .....	66
가져오기 상태 모니터링 .....	75
작업 완료 보고서 및 로그 가져오기 .....	75
대용량 데이터 마이그레이션 .....	79
대량 전송 계획 .....	79
1단계: 클라우드로 이동하는 대상 이해 .....	80
2단계: 대상 전송 속도 계산 .....	80
3단계: 필요한 Snowball Edge 수 결정 .....	80
4단계: 작업 생성 .....	81
5단계: 데이터를 전송 세그먼트로 분리 .....	81
대규모 전송 보정 .....	82
대규모 데이터 마이그레이션 계획 수립 .....	83
1단계: 마이그레이션 세부 정보 선택 .....	83
2단계: 배송, 보안 및 알림 기본 설정 선택 .....	88
3단계: 계획 검토 및 생성 .....	89
대규모 데이터 마이그레이션 계획 사용 .....	89
권장 작업 주문 일정 .....	89
주문된 작업 목록 .....	92

모니터링 대시보드 .....	92
AWS OpsHub 를 사용하여 디바이스 관리 .....	93
다운로드 AWS OpsHub .....	94
디바이스 잠금 해제 .....	94
로컬에서 디바이스 잠금 해제 .....	95
원격으로 디바이스 잠금 해제 .....	98
의 서명 확인 AWS OpsHub .....	101
AWS 서비스 관리 .....	105
Amazon EC2 호환 인스턴스 시작하기 .....	106
Amazon EC2 호환 인스턴스 중지 .....	108
Amazon EC2 호환 인스턴스 시작 .....	109
키 페어 작업 .....	110
Amazon EC2 호환 인스턴스 종료 .....	110
EBS 볼륨 관리 .....	111
이미지를 Amazon EC2 호환 AMI로 디바이스로 가져오기 .....	113
스냅샷 삭제 .....	117
AMI 등록 해제 .....	117
클러스터 관리 .....	118
를 사용하여 Snowball Edge에서 Amazon S3 호환 스토리지 설정 AWS OpsHub .....	119
S3 스토리지 관리 .....	125
NFS 인터페이스 관리 .....	129
디바이스 재부팅 .....	136
를 사용하여 프로필 관리 AWS OpsHub .....	139
디바이스 종료 .....	139
디바이스 별칭 편집 .....	142
OpsHub를 사용한 공개 키 인증서 관리 .....	142
OpsHub를 사용하여 퍼블릭 키 인증서 다운로드 .....	143
OpsHub를 사용하여 퍼블릭 키 인증서 갱신 .....	143
디바이스 업데이트 가져오기 .....	144
업데이트 AWS OpsHub .....	145
를 사용하여 관리 작업 자동화 AWS OpsHub .....	145
태스크 생성 및 시작 .....	146
태스크 세부 정보 보기 .....	149
태스크 삭제 .....	149
디바이스의 NTP 타임 서버 설정 .....	150
Snowball Edge 클라이언트 구성 및 사용 .....	152

Snowball Edge 클라이언트 다운로드 및 설치 .....	152
Snowball Edge 클라이언트의 프로파일 구성 .....	154
Snowball Edge 클라이언트 버전 찾기 .....	157
자격 증명 얻기 .....	158
Snowball Edge에서 서비스 시작 .....	159
Snowball Edge에서 서비스 중지 .....	159
로그 보기 및 다운로드 .....	160
디바이스 상태 보기 .....	162
서비스 상태 보기 .....	163
기능 상태 보기 .....	168
타임 서버 설정 .....	169
시간 소스 확인 .....	169
타임 서버 업데이트 .....	170
NFC 태그 검증 .....	171
MTU 크기 업데이트 .....	171
S3 어댑터를 사용하여 파일 전송 .....	173
다운로드 및 설치 AWS CLI .....	174
Linux 운영 체제 AWS CLI 에 설치 .....	174
Windows 운영 체제 AWS CLI 에 설치 .....	174
Snowball Edge 디바이스에서 AWS CLI 및 API 작업 사용 .....	175
AWS Snowball Edge에 대한 Amazon S3 API 인터페이스를 사용한 권한 부여 .....	175
로컬 Amazon S3 자격 증명 가져오기 및 사용 .....	176
S3 어댑터를 AWS CLI 엔드포인트로 구성 .....	176
S3 어댑터에 지원되지 않는 Amazon S3 기능 .....	177
작은 파일 배치 처리 .....	178
데이터 전송에 지원되는 AWS CLI 명령 .....	180
Amazon S3에 지원되는 AWS CLI 명령 .....	181
데이터 전송을 위해 지원되는 Amazon S3 REST API 작업 .....	184
NFS 인터페이스 관리 .....	186
Snowball Edge에 대한 NFS 구성 .....	187
NFS 인터페이스에 대한 Snowball Edge 구성 .....	187
Snowball Edge에서 NFS 서비스 시작 .....	188
클라이언트 컴퓨터에 NFS 엔드포인트 탑재 .....	190
NFS 인터페이스 중지 .....	191
Amazon EC2 호환 컴퓨팅 인스턴스 사용 .....	192
Amazon EC2-compatible 인스턴스 설명 .....	193

EC2 인스턴스 요금 .....	193
Snowball Edge에서 AMIs 사용 .....	193
주문 시 AMI 추가 .....	195
에서 AMI 추가 AWS Marketplace .....	196
디바이스 수신 후 AMI 추가 .....	200
Snowball Edge에 Microsoft Windows AMI 추가 .....	200
Snowball Edge로 VM 이미지 가져오기 .....	202
Snowball Edge용 최신 Amazon Linux 2 AMI 내보내기 .....	203
Snowball Edge 디바이스로 VM 이미지 가져오기 .....	203
1단계: VM 이미지를 준비하여 Snowball Edge 디바이스에 업로드 .....	204
2단계: Snowball Edge에 필요한 권한 설정 .....	205
3단계: Snowball Edge에서 VM 이미지를 스냅샷으로 가져오기 .....	212
4단계: Snowball Edge에서 스냅샷을 AMI로 등록 .....	213
5단계: Snowball Edge의 AMI에서 인스턴스 시작 .....	214
Snowball Edge에 대한 추가 AMI 작업 .....	216
AWS CLI 및 API 작업 사용 .....	219
컴퓨팅 인스턴스에 대한 네트워크 구성 .....	220
DNI 및 VNI 사전 조건 .....	221
가상 네트워크 인터페이스(VNI) 설정 .....	222
DNI 설정 .....	223
SSH를 사용하여 컴퓨팅 인스턴스에 연결 .....	226
동일한 디바이스의 컴퓨팅 인스턴스에서 버킷으로 데이터 전송 .....	227
인스턴스 자동 시작 .....	228
EC2-compatible 시작 구성 생성 .....	228
EC2-compatible 시작 구성 업데이트 .....	228
EC2-compatible 시작 구성 삭제 .....	229
EC2-compatible 시작 구성 나열 .....	229
Snowball Edge에서 가상 네트워크 인터페이스 생성 .....	229
가상 네트워크 인터페이스 설명 .....	230
가상 네트워크 인터페이스 업데이트 .....	231
가상 네트워크 인터페이스 삭제 .....	232
Amazon EC2-compatible 엔드포인트 사용 .....	233
EC2-compatible 엔드포인트를 AWS CLI 엔드포인트로 지정 .....	233
지원되는 EC2-compatible AWS CLI 명령 .....	234
지원되는 Amazon EC2-compatible API 작업 .....	249
EC2-compatible 인스턴스 자동 시작 .....	251

EC2-compatible 인스턴스와 함께 Snow용 IMDS 사용 .....	253
Snowball Edge의 IMDS 버전 .....	254
IMDSv1 및 IMDSv2를 사용하여 인스턴스 메타데이터를 검색하는 예시 .....	257
EC2-compatible 인스턴스에 블록 스토리지 사용 .....	262
보안 그룹에서 네트워크 트래픽 제어 .....	263
지원되는 EC2-compatible 인스턴스 메타데이터 및 사용자 데이터 .....	264
컴퓨팅 인스턴스 사용자 데이터 .....	265
EC2-compatible 인스턴스 중지 .....	266
EC2-compatible 인스턴스 AWS IoT Greengrass 에서 사용 .....	267
EC2-compatible 설정 AWS IoT Greengrass .....	267
Snowball Edge AWS IoT Greengrass 의 EC2-compatible 인스턴스에 설치 .....	268
사용 AWS Lambda .....	271
Lambda 사용 시작하기 .....	271
에 대한 사전 조건 AWS IoT Greengrass .....	271
Lambda에 대한 사전 조건 .....	272
Lambda 함수를 Snowball Edge 디바이스에 배포하기 .....	273
Snowball Edge에서 Amazon S3 호환 스토리지 사용 .....	274
Snowball Edge에서 Amazon S3 호환 스토리지 주문 .....	277
Snowball Edge에서 Amazon S3 호환 스토리지 설정 및 시작 .....	277
사전 조건 .....	277
로컬 환경 설정 .....	278
Snowball Edge 서비스에서 Amazon S3 호환 스토리지 시작 .....	279
엔드포인트에 대한 정보 보기 .....	281
S3 버킷 작업 .....	283
사용 AWS CLI .....	283
Java SDK 사용 .....	284
버킷 ARN 형식 .....	285
버킷 위치 형식 .....	285
버킷 액세스 결정 .....	286
버킷 또는 리전 버킷 목록 검색 .....	286
버킷 가져오기 .....	288
S3 버킷 생성 .....	288
버킷 삭제 .....	289
를 사용하여 객체 수명 주기 구성 생성 및 관리 AWS CLI .....	290
Snowball Edge 버킷에 수명 주기 구성 추가 .....	290
객체 복사 .....	292



객체 나열 .....	293
객체 가져오기 .....	295
객체 삭제 .....	297
Snowball Edge의 Amazon S3 호환 스토리지에 지원되는 REST API 작업 .....	298
Snow 디바이스 클러스터와 함께 Snowball Edge에서 Amazon S3 호환 스토리지 사용 .....	300
Snowball Edge 클러스터 쿼럼 .....	302
사용할 수 없는 클러스터 노드 다시 연결 .....	303
클러스터에서 노드 교체 .....	304
Snowball Edge에서 Amazon S3 호환 스토리지 이벤트 알림 구성 .....	316
로컬 SMTP 알림 구성 .....	318
로컬 알림을 위한 Snowball Edge 구성 .....	319
Snowball Edge에서 Amazon EKS Anywhere 사용 .....	320
Amazon EKS Anywhere on Snow용 AWS Snowball Edge 디바이스 주문 전에 완료해야 할 작업 .....	322
Ubuntu EKS Distro AMI 생성 .....	322
Harbor AMI 구축 .....	323
Amazon EKS Anywhere on Snow와 함께 사용할 AWS Snowball Edge 디바이스 주문 .....	323
Snowball Edge 디바이스에서 Amazon EKS Anywhere를 구성하고 실행 .....	324
초기 설정 .....	324
Amazon EKS Anywhere를 자동으로 구성 및 실행 .....	325
Amazon EKS Anywhere를 수동으로 구성 및 실행 .....	325
연결 해제된 작업을 위해 AWS Snow에서 Amazon EKS Anywhere 구성 .....	334
Snowball Edge 디바이스에서 Harbor 레지스트리를 구성합니다. ....	335
Snowball Edge의 Amazon EKS Anywhere 관리자 인스턴스에서 Harbor 레지스트리 사용 ....	335
클러스터 생성 및 유지 관리 .....	335
Snowball Edge에서 클러스터를 생성하는 모범 사례 .....	335
Snowball Edge에서 클러스터 업그레이드 .....	335
Snowball Edge에서 클러스터 리소스 정리 .....	336
로컬로 IAM 사용 .....	337
AWS CLI 및 API 작업 사용 .....	338
지원되는 IAM AWS CLI 명령 .....	338
Snowball Edge에서 지원되는 IAM API 작업 .....	340
Snowball Edge에서 지원되는 IAM 정책 버전 및 문법 .....	341
Snowball Edge의 IAM 정책 예제 .....	342
IAM API를 통해 Snowball Edge에서 샘플 사용자에게 GetUserInfo 호출 허용 .....	342
Snowball Edge에서 Amazon S3 API에 대한 전체 액세스 허용 .....	343

Snowball Edge의 Amazon S3 버킷에 대한 읽기 및 쓰기 액세스 허용 .....	343
.....	344
Snowball Edge의 Amazon S3 버킷에 대한 목록, 가져오기 및 액세스 허용 .....	344
Snowball Edge에서 Amazon EC2 API에 대한 전체 액세스 허용 .....	344
Snowball Edge에서 Amazon EC2-compatible 인스턴스를 시작하고 중지할 수 있는 액세스 허용 .....	345
DescribeLaunchTemplates에 대한 호출을 거부하지만 Snowball Edge에서 DescribeImages에 대한 모든 호출 허용 .....	345
Snowball Edge의 API 호출 정책 .....	346
Snowball Edge의 TrustPolicy 예제 .....	347
사용 AWS STS .....	348
Snowball Edge에서 AWS CLI 및 API 작업 사용 .....	348
Snowball Edge에서 지원되는 AWS STS AWS CLI 명령 .....	349
Snowball Edge에서 역할을 수입하는 명령 예제 .....	349
Snowball Edge에서 지원되는 AWS STS API 작업 .....	349
퍼블릭 키 인증서 관리 .....	351
인증서 나열 .....	351
인증서 나열 .....	352
인증서 삭제 .....	352
AWS 서비스에 대한 포트 요구 사항 .....	354
Snowball Edge 디바이스 관리를 사용하여 디바이스 관리 .....	356
Snowball Edge 주문 시 Snowball Edge 디바이스 관리 상태 선택 .....	357
Snowball Edge에서 Snowball Edge 디바이스 관리 활성화 .....	358
Snowball Edge의 IAM 역할에 Snowball Edge 디바이스 관리에 대한 권한 추가 .....	359
Snowball Edge Device Management CLI 명령 .....	360
Snowball Edge 디바이스 관리 작업 생성 .....	361
Snowball Edge 디바이스 관리 작업 상태 확인 .....	362
Snowball Edge Device Management를 사용하여 디바이스 정보 확인 .....	363
Snowball Edge Device Management를 사용하여 EC2-compatible 인스턴스의 상태 확인 .....	364
Snowball Edge Device Management 작업 메타데이터 보기 .....	365
Snowball Edge 디바이스 관리 작업 취소 .....	367
Snowball Edge 디바이스 관리 명령 및 구문 나열 .....	368
원격 관리에 사용할 수 있는 Snowball Edge 나열 .....	368
디바이스 간 작업 상태 나열 .....	369
디바이스에서 사용 가능한 리소스 나열 .....	370
디바이스 또는 태스크 태그 나열 .....	371

상태별로 태스크 나열 .....	372
태스크 또는 디바이스에 태그 적용 .....	373
태스크 또는 디바이스에서 태그 제거 .....	374
Snowball Edge 디바이스 업데이트 .....	375
소프트웨어 업데이트를 위한 사전 조건 .....	376
업데이트 다운로드 .....	376
업데이트 설치 .....	380
SSL 인증서 업데이트 .....	386
Amazon Linux 2 AMI 업데이트 .....	387
작업 이해 .....	388
작업 세부 정보 .....	388
작업 상태 .....	391
Snowball Edge 클러스터 작업의 상태 .....	393
가져오기 작업 .....	394
내보내기 작업 .....	395
내보내기 작업에 Amazon S3 객체 키 사용 .....	396
내보내기 작업에 대한 모범 사례 .....	404
로컬 컴퓨팅 및 스토리지 작업에 대한 정보 .....	404
로컬 스토리지 작업에 대한 정보 .....	405
디바이스 클러스터의 로컬 스토리지에 대한 정보 .....	405
모범 사례 .....	406
보안 권장 사항 .....	406
리소스 관리 모범 사례 .....	407
데이터 전송 성능 권장 사항 .....	407
데이터 전송 속도 개선 .....	408
보안 .....	410
데이터 보호 .....	410
클라우드에서 데이터 보호 .....	411
디바이스에서 데이터 보호 .....	415
ID 및 액세스 관리 .....	418
콘솔 및 작업에 대한 액세스 제어 .....	419
로깅 및 모니터링 .....	458
규정 준수 확인 .....	458
복원성 .....	459
인프라 보안 .....	459
데이터 유효성 검사 .....	461

로컬 파일 인벤토리 .....	461
Snowball Edge에서 데이터 검증 오류의 일반적인 원인 .....	461
Amazon S3로 가져온 후 수동으로 데이터 검증 .....	462
알림 .....	464
Snow가 Amazon SNS를 사용하는 방법 .....	464
작업 상태 변경에 대한 SNS 주제 암호화 .....	464
고객 관리형 KMS 키 정책 설정 .....	465
SNS 알림 예제 .....	466
를 사용하여 로깅 AWS CloudTrail .....	479
AWS Snowball Edge CloudTrail의 정보 .....	479
로그 파일 항목 이해 .....	480
할당량 .....	482
의 리전 가용성 AWS Snowball Edge .....	482
AWS Snowball Edge 작업에 대한 제한 사항 .....	483
의 속도 제한 AWS Snowball Edge .....	484
Amazon Snow S3 어댑터 연결 한도 .....	484
온프레미스 데이터 전송 제한 .....	484
컴퓨팅 인스턴스 할당량 .....	484
Snowball Edge의 컴퓨팅 리소스 스토리지 할당량 .....	485
Snowball Edge의 공유 컴퓨팅 리소스 제한 사항 .....	487
배송 제한 사항 .....	488
반환된 디바이스를 가져오기 작업을 위한 처리 제한 .....	488
문제 해결 .....	490
디바이스 식별 .....	491
부팅 문제 해결 .....	492
부팅 중 LCD 디스플레이 관련 문제 해결 .....	492
부팅 중 E Ink 디스플레이 관련 문제 해결 .....	494
연결 문제 해결 .....	495
unlock-device 명령 문제 해결 .....	495
매니페스트 파일 문제 해결 .....	496
자격 증명 문제 해결 .....	496
AWS CLI 자격 증명을 찾을 수 없는 문제 해결 .....	496
문제 해결 오류 메시지: 시크릿 액세스 키와 서명을 확인하세요. ....	497
데이터 전송 문제 해결 .....	497
가져오기 작업 문제 해결 .....	498
내보내기 작업 문제 해결 .....	498

NFS 인터페이스 문제 해결 .....	499
S3 인터페이스의 액세스 거부 오류 .....	500
S3 인터페이스에서 S3 금지 오류 발생 .....	503
AWS CLI 문제 해결 .....	507
AWS CLI 오류 메시지 문제 해결: "Profile cannot Be Null" .....	508
를 사용하여 데이터를 전송할 때 null 포인터 오류 문제 해결 AWS CLI .....	508
컴퓨팅 인스턴스 문제 해결 .....	508
가상 네트워크 인터페이스의 IP 주소가 0.0.0.0인 경우 .....	509
대규모 컴퓨팅 인스턴스를 시작할 때 Snowball Edge 디바이스가 응답 중지 .....	509
Snowball Edge의 인스턴스에 루트 볼륨이 하나 있음 .....	509
보호되지 않는 프라이빗 키 파일 오류 .....	509
문서 기록 .....	511
.....	dxviii

# Snowball Edge란 무엇인가요?

Snowball Edge는 일부 AWS 기능을 위한 온보드 스토리지 및 컴퓨팅 성능을 갖춘 디바이스입니다. Snowball Edge는 로컬에서 데이터를 처리하고, 엣지 컴퓨팅 워크로드를 실행하고, 데이터를 AWS 클라우드 또는 AWS Cloud에서 전송할 수 있습니다.

각 Snowball 디바이스는 인터넷보다 더 빠르게 데이터를 전송할 수 있습니다. 이 전송은 리전 운송업체를 통해 디바이스의 데이터를 운송하는 방식으로 이루어집니다. 어플라이언스는 견고하며 E Ink 배송 라벨이 부착되어 있습니다.

Snowball Edge 디바이스에는 스토리지 최적화 210TB와 컴퓨팅 최적화라는 두 가지 디바이스 구성 옵션이 있습니다. 이 설명서상의 Snowball Edge 디바이스는 해당 디바이스의 모든 선택적 변형을 의미합니다. 특정 정보가 하나 이상의 선택적 디바이스 구성에만 적용되는 경우 특별히 호출됩니다. 자세한 내용은 [Snowball Edge 디바이스 구성](#) 단원을 참조하십시오.

## 주제

- [Snowball Edge 기능](#)
- [Snowball Edge와 관련된 서비스](#)
- [Snowball Edge 서비스에 액세스](#)
- [Snowball Edge 요금](#)
- [AWS Snowball Edge 모니터링](#)
- [최초 AWS Snowball Edge 사용자를 위한 리소스](#)
- [AWS Snowball Edge 디바이스 하드웨어 정보](#)
- [Snowball Edge를 사용하기 위한 사전 조건](#)

## Snowball Edge 기능

Snowball Edge 디바이스에는 다음과 같은 기능이 있습니다.

- 디바이스를 위한 대용량 스토리지 또는 컴퓨팅 기능. 작업을 생성할 때 선택하는 옵션에 따라 달라집니다.
- 전송 속도가 최대 100Gbit/초인 네트워크 어댑터.
- 저장 데이터와 물리적으로 전송 중인 데이터를 보호하기 위해 암호화가 실행됩니다.
- 로컬 환경과 Amazon S3 간에 데이터를 가져오거나 내보낼 수 있으며 인터넷을 사용하지 않고도 하나 이상의 디바이스로 데이터를 물리적으로 전송할 수 있습니다.

- Snowball Edge 디바이스는 견고한 박스형 제품입니다. 디바이스를 배송할 준비가 되면 내장 E Ink 디스플레이가 변경되어 배송 라벨이 표시됩니다.
- Snowball Edge 디바이스에는 네트워크 연결을 관리하고 서비스 상태 정보를 가져오는 데 사용할 수 있는 온보드 LCD 디스플레이가 함께 제공됩니다.
- 로컬 스토리지 및 컴퓨팅 작업에 Snowball Edge 디바이스를 클러스터링하여 3~16개 디바이스에서 데이터 내구성을 확보하고 필요에 따라 스토리지를 로컬로 확장 또는 축소할 수 있습니다.
- Kubernetes 워크로드용 Snowball Edge 디바이스에서 Amazon EKS Anywhere를 사용할 수 있습니다.
- Snowball Edge 디바이스에는 Amazon S3 및 Amazon EC2 호환 엔드포인트가 있어 프로그래밍 방식의 사용 사례가 지원됩니다.
- Snowball Edge 디바이스는 Amazon Machine Image(AMI)를 사용하여 디바이스에서 컴퓨팅 인스턴스를 실행하는 데 사용할 수 있는 새로운 sbe1, sbe-c 및 sbe-g 인스턴스 유형을 지원합니다.
- Snowball Edge는 데이터 마이그레이션을 위해 다음과 같은 데이터 전송 프로토콜을 지원합니다.
  - NFSv3
  - NFSv4
  - NFSv4.1
  - HTTP 또는 HTTPS를 통한 Amazon S3( AWS CLI 버전 1.16.14 이하와 호환되는 API를 통해)

## Snowball Edge와 관련된 서비스

다음과 같은 관련 AWS 서비스와 함께 AWS Snowball Edge 디바이스를 사용할 수 있습니다.

- Amazon S3 어댑터 - Amazon S3 API 작업의 하위 집합을 지원하는 Snowball Edge용 Amazon S3 API를 AWS 사용하여 로 또는 밖으로 프로그래밍 방식으로 데이터를 전송하는 데 사용됩니다. 이 역할에서 데이터는 사용자를 대신하여 AWS 에 의해 Snow 디바이스로 전송되고 디바이스는 사용자에게 배송되거나(내보내기 작업의 경우) 빈 Snow 디바이스를 사용자에게 AWS 배송하고 사용자는 온프레미스 소스에서 디바이스로 데이터를 전송한 후 다시 배송합니다 AWS (가져오기 작업의 경우)"
- Snowball Edge의 Amazon S3 호환 스토리지 - Amazon EC2, Amazon EKS Anywhere on Snow 등과 같은 컴퓨팅 서비스의 데이터 요구 사항을 지원하는 데 사용됩니다. 이 기능은 Snowball Edge 디바이스에서 사용할 수 있으며, 확장된 Amazon S3 API 세트와 3~16개 노드의 유연한 클러스터 설정을 통한 복원력 향상, 로컬 버킷 관리, 로컬 알림과 같은 기능을 제공합니다.
- Amazon EC2 - Amazon EC2 API 작업의 하위 집합을 지원하는 Amazon EC2 호환 엔드포인트를 사용하여 Snowball Edge 디바이스에서 컴퓨팅 인스턴스를 실행합니다. AWS에서 Amazon EC2를 사

용하는 것에 대한 자세한 내용은 [Getting started with Amazon EC2 Linux instances](#) 섹션을 참조하세요.

- Amazon EKS Anywhere on Snow - Snowball Edge 디바이스에서 Kubernetes 클러스터를 생성하고 운영합니다. [AWS Snow에서 Amazon EKS Anywhere 사용](#)(를) 참조하세요.
- AWS Lambda 기반 AWS IoT Greengrass- AWS Snowball Edge 디바이스에서 수행된 Amazon S3 compatible storage on Snowball Edge 스토리지 작업을 기반으로 Lambda 함수를 호출합니다. Lambda 사용에 대한 자세한 내용은 [와 AWS Lambda 함께 사용 AWS Snowball Edge](#) 및 [AWS Lambda 개발자 안내서](#) 섹션을 참조하세요.
- Amazon Elastic Block Store(Amazon EBS)는 EC2 호환 인스턴스에 사용할 수 있는 블록 수준 스토리지 볼륨을 제공합니다. 자세한 내용은 [Amazon Elastic Block Store\(Amazon EBS\)](#) 섹션을 참조하세요.
- AWS Identity and Access Management (IAM) -이 서비스를 사용하여 AWS 리소스에 대한 액세스를 안전하게 제어합니다. 자세한 내용은 [IAM이란 무엇입니까?](#) 섹션을 참조하세요.
- AWS Security Token Service (AWS STS) - IAM 사용자 또는 인증하는 사용자(페더레이션 사용자)에 대해 권한이 제한된 임시 자격 증명을 요청합니다. 자세한 정보는 [IAM의 임시 보안 자격 증명](#) 섹션을 참조하세요.
- Amazon EC2 Systems Manager - 이 서비스를 사용하여 AWS에서 인프라를 보고 제어할 수 있습니다. 자세한 내용은 [AWS Systems Manager란 무엇인가요?](#) 섹션을 참조하세요.

## Snowball Edge 서비스에 액세스

[AWS Snow 패밀리 관리 콘솔](#) 또는 작업 관리 API를 사용하여 작업을 생성하고 관리할 수 있습니다. [AWS Snow 패밀리 관리 콘솔](#) 사용에 대한 자세한 내용은 [Snowball Edge 시작하기](#) 단원을 참조하십시오. 작업 관리 API에 대한 자세한 내용은 [Snowball Edge용 작업 관리 API 참조](#)를 참조하세요.

## AWS Snowball Edge 디바이스 액세스

Snowball Edge 디바이스를 온사이트에 설치한 후 LCD 화면을 사용하여 IP 주소를 구성한 다음 Snowball Edge 클라이언트 또는 AWS OpsHub를 사용하여 디바이스를 잠금 해제할 수 있습니다. 그런 다음 데이터 전송 또는 엣지 컴퓨팅 작업을 수행할 수 있습니다. 자세한 내용은 [Snowball Edge 수신](#)을 참조하십시오.

## Snowball Edge 요금

서비스 및 디바이스의 요금 및 비용에 대한 내용은 [AWS Snowball Edge Pricing](#) 섹션을 참조하세요.



# AWS Snowball Edge 모니터링

AWS 는 Snow 디바이스를 모니터링하고 Snow 디바이스가에 연결될 때 지표 및 사용 정보를 수집할 수 있습니다 AWS 리전. Snow 디바이스가에 연결되지 않은 경우 AWS 리전 AWS 는 Snow 디바이스를 모니터링하지 않습니다.

에서 복구할 수 없는 문제를 AWS 감지하고 물리적 장비를 교체해야 하는 경우 AWS 에서 알립니다. 그런 다음 사이트로 배송할 대체 작업을 배치할 수 있습니다. Snow 디바이스 모니터링은 Snow 디바이스 서비스 요금의 일부로 포함되므로 이에 대한 추가 요금은 없습니다.

## 최초 AWS Snowball Edge 사용자를 위한 리소스

AWS Snowball Edge 서비스를 처음 사용하는 경우 다음 섹션을 순서대로 읽는 것이 좋습니다.

1. 디바이스 유형 및 옵션에 대한 자세한 내용은 [AWS Snowball Edge 디바이스 하드웨어 정보](#) 섹션을 참조하세요.
2. 작업 유형에 더 알아보려면 [Snowball Edge 작업 이해](#) 섹션을 참조하세요.
3. AWS Snowball Edge 디바이스 사용 방법에 대한 end-to-end 개요는 섹션을 참조하세요 [AWS Snowball Edge 작동 방식](#).
4. 시작할 준비가 되면 [Snowball Edge 시작하기](#) 섹션을 참조하세요.
5. 컴퓨팅 인스턴스를 사용하는 작업에 대한 자세한 내용은 [Snowball Edge에서 Amazon EC2-compatible 컴퓨팅 인스턴스 사용](#) 섹션을 참조하세요.

## AWS Snowball Edge 디바이스 하드웨어 정보

모든 Snowball Edge 디바이스는 크기 및 무게와 같은 물리적 특성을 공유하지만 용도에 맞는 다양한 유형의 하드웨어를 포함합니다. 데이터 전송을 위해 설계된 디바이스는 더 많은 스토리지로 구성되고 컴퓨팅을 위해 설계된 디바이스는 더 많은 가상 CPU 및 메모리로 구성됩니다. 이 섹션에서는 Snowball Edge 디바이스의 물리적 특성과 컴퓨팅 및 스토리지 사양에 대한 정보를 제공합니다.

### 주제

- [Snowball Edge 디바이스 구성](#)
- [AWS Snowball Edge 디바이스 사양](#)
- [Snowball Edge에서 지원하는 네트워크 하드웨어](#)

## Snowball Edge 디바이스 구성

Snowball Edge 디바이스에는 다음과 같은 디바이스 구성 옵션이 있습니다.

- Snowball Edge 스토리지 최적화 210TB - 이 Snowball Edge 디바이스 옵션은 210TB의 사용 가능한 스토리지 용량을 갖추고 있습니다.
- Snow Edge 컴퓨팅 최적화 - 이 Snowball Edge 디바이스(AMD EPYC Gen2 포함)는 최대 104개의 vCPU, 416GB의 메모리, 28TB의 컴퓨팅 인스턴스 전용 NVMe SSD를 포함하여 가장 많은 컴퓨팅 기능을 갖추고 있습니다.

### Note

이러한 디바이스에서 Snowball Edge의 Amazon S3 호환 스토리지를 사용하는 경우 사용 가능한 스토리지는 달라집니다. [Amazon S3 compatible storage on Snowball Edge의 스토리지 용량은 Snowball Edge의](#) Snowball Edge에서 Amazon S3 compatible storage 사용을 참조하세요.

이 세 가지 옵션의 컴퓨팅 기능에 대한 자세한 내용은 [Snowball Edge에서 Amazon EC2-compatible 컴퓨팅 인스턴스 사용](#) 섹션을 참조하세요. 작업 생성 및 디스크 용량 차이(테라바이트)는 [여기](#)에 설명되어 있습니다.

### Note

Snowball Edge 디바이스에는 디바이스의 모든 옵션 변형이 포함됩니다. 정보가 하나 이상의 특정 선택적 구성에 적용되는 경우 이를 명시적으로 언급합니다.

다음 테이블에는 다양한 디바이스 옵션 간의 차이점이 요약되어 있습니다. 하드웨어 사양에 대한 자세한 내용은 [AWS Snowball Edge 디바이스 사양](#) 섹션을 참조하세요.

	Snowball Edge 스토리지 최적화 210TB	Snowball Edge 컴퓨팅 최적화 (AMD EPYC Gen2 및 NVMe 포함)
CPU	AMD Rome, 64코어, 2GHz	AMD Rome, 64코어, 2GHz

	Snowball Edge 스토리지 최적화 210TB	Snowball Edge 컴퓨팅 최적화 (AMD EPYC Gen2 및 NVME 포함)
vCPU	104	104
사용 가능한 메모리	416GB	416GB
보안 카드	예	예
SSD	210TB NVMe	28TB NVMe
사용 가능한 HDD	해당 사항 없음	해당 사항 없음
네트워크 인터페이스	<ul style="list-style-type: none"> <li>• 2개 10Gbit – RJ45(하나 사용 가능)</li> <li>• 25Gbit SFP28 1개</li> <li>• 100Gbit QSFP28 1개</li> </ul>	<ul style="list-style-type: none"> <li>• 2개 10Gbit – RJ45(하나 사용 가능)</li> <li>• 25Gbit SFP28 1개</li> <li>• 1x 100Gbit – QSFP28</li> </ul>
물리적 보안 기능	<ul style="list-style-type: none"> <li>• 숨겨진 마그네틱 나사</li> <li>• 침입 탐지 스위치</li> <li>• NFC 태그</li> <li>• 변조 방지 인서트</li> <li>• 변조 감지용 Android 앱</li> <li>• 컨포멀 코팅</li> </ul>	<ul style="list-style-type: none"> <li>• 숨겨진 마그네틱 나사</li> <li>• 침입 탐지 스위치</li> <li>• NFC 태그</li> <li>• 변조 방지 인서트</li> <li>• 변조 감지용 Android 앱</li> <li>• 컨포멀 코팅</li> </ul>

## AWS Snowball Edge 디바이스 사양

이 섹션에서는 AWS Snowball Edge 디바이스 유형 및 하드웨어에 대한 사양을 찾을 수 있습니다.

### 주제

- [Snowball Edge 스토리지 최적화 210TB 사양](#)
- [Snowball Edge 컴퓨팅 최적화 디바이스 사양](#)

## Snowball Edge 스토리지 최적화 210TB 사양

Snowball Edge 스토리지 최적화 210TB 디바이스에 대한 하드웨어 사양은 다음 테이블에서 확인하세요.

Item	Snowball Edge 스토리지 최적화 210TB 사양
컴퓨팅 및 메모리 사양	
CPU	vCPU 104개
RAM	416GB
스토리지 사양	
NVME 스토리지 용량	210TB의 가용 용량(객체 및 NFS 데이터 전송용)
SSD 스토리지 용량	없음
전원 공급 장치 사양	
Power	미국 AWS 리전 내 : NEMA 5-15p 100-220볼트. 모든 AWS 리전에서 전원 케이블이 포함되어 있습니다.
전력 소비	평균 사용 사례의 경우 304W(전원 공급은 정격 1,200W)
전압	100 - 240V AC
빈도	47/63Hz
데이터 및 네트워크 연결	2개 10Gbit - RJ45(하나 사용 가능) 25Gbit SFP28 1개 100Gbit QSFP28 1개
케이블	각 AWS Snowball Edge 디바이스에는 국가별 전원 케이블이 제공됩니다. 다른 케이블 또는 광 커넥터는 제공되지 않습니다. 자세한 내용은 <a href="#">Snowball Edge에서 지원하는 네트워크 하드웨어</a> 섹션을 참조하세요.

Item	Snowball Edge 스토리지 최적화 210TB 사양
열 요구 사항	AWS Snowball Edge 디바이스는 사무실 운영용으로 설계되었으며 데이터 센터 운영에 적합합니다.
데시벨 출력	평균적으로 AWS Snowball Edge 디바이스는 68데시벨의 사운드를 생성하며, 일반적으로 vacuum 클리너 또는 거실 음악보다 조용합니다.
치수 및 무게 사양	
가중치	22.54kg(49.7파운드)
Height	394mm(15.5인치)
너비	265mm(10.6인치)
길이	718mm(28.3인치)
환경 사양	
진동	비작동 사용 시 ASTM D4169 트럭 레벨 I 0.73 GRMS에 상응
충격	작동 사용 시 70G(MIL-S-901)에 상응 비작동 사용 시 50G(ISTA-3A)에 상응
Altitude	작동 사용 시 0-3,000미터(0-10,000피트)에 상응 비작동 사용 시 0-12,000미터에 상응
온도 범위	0-30°C(작동)

### Snowball Edge 컴퓨팅 최적화 디바이스 사양

Item	Snowball Edge 컴퓨팅 최적화 사양
컴퓨팅 및 메모리 사양	
CPU	vCPU 104개

Item	Snowball Edge 컴퓨팅 최적화 사양
RAM	512GB RAM(최대 416GB RAM - 고객 사용 가능)
스토리지 사양	
SSD 스토리지 용량	28TB NVMe SSD
전원 공급 장치 사양	
Power	미국 AWS 리전 내 : NEMA 5-15p 100-220볼트. 모든 AWS 리전에서 전원 케이블이 포함되어 있습니다.
전력 소비	평균 사용 사례의 경우 304W(전원 공급은 정격 1,200W)
전압	100 - 240V AC
빈도	47/63Hz
데이터 및 네트워크 연결	2개 10Gbit – RJ45(하나 사용 가능)  25Gbit SFP28 1개  100Gbit QSFP28 1개
케이블	각 AWS Snowball Edge 디바이스에는 국가별 전원 케이블이 제공됩니다. 다른 케이블 또는 광 커넥터는 제공되지 않습니다. 자세한 내용은 <a href="#">Snowball Edge에서 지원하는 네트워크 하드웨어</a> 섹션을 참조하세요.
열 요구 사항	AWS Snowball Edge 디바이스는 사무실 운영용으로 설계되었으며 데이터 센터 운영에 적합합니다.
데시벨 출력	평균적으로 AWS Snowball Edge 디바이스는 68데시벨의 사운드를 생성하며, 일반적으로 vacuum 콜리너 또는 거실 음악보다 조용합니다.
치수 및 무게 사양	
가중치	22.54kg(49.7파운드)

Item	Snowball Edge 컴퓨팅 최적화 사양
Height	394mm(15.5인치)
너비	265mm(10.6인치)
길이	718mm(28.3인치)
환경 사양	
진동	비작동 사용 시 ASTM D4169 트럭 레벨 I 0.73 GRMS에 상응
충격	작동 사용 시 70G(MIL-S-901)에 상응 비작동 사용 시 50G(ISTA-3A)에 상응
Altitude	작동 사용 시 0-3,000미터(0-10,000피트)에 상응 비작동 사용 시 0-12,000미터에 상응
온도 범위	0-45°C(작동)

## Snowball Edge에서 지원하는 네트워크 하드웨어

AWS Snowball Edge 디바이스를 사용하려면 자체 네트워크 케이블이 필요합니다. RJ45 케이블에 관련된 권장 사항은 없습니다. Mellanox 및 Finisar의 SFP+ 및 QSFP+ 케이블과 모듈은 디바이스와 호환되는 것으로 확인되었습니다.

AWS Snowball Edge 디바이스의 후면 패널을 열면 다음 스크린샷에 표시된 포트와 유사한 네트워크 포트가 표시됩니다.



AWS Snowball Edge 디바이스의 네트워크 인터페이스는 한 번에 하나만 사용할 수 있습니다. 따라서 포트 중 하나를 사용하여 다음 네트워크 하드웨어를 지원하세요.

## SFP

이 포트는 SFP28 및 SFP+ 송수신장치 모듈 그리고 DAC(direct-attach copper) 케이블과 호환되는 10G/25G SFP28 인터페이스를 제공합니다. 사용자는 자신의 송수신장치 또는 DAC 케이블을 제공해야 합니다.

- 10G 작업의 경우 SFP+ 옵션을 사용할 수 있습니다. 그러한 예는 다음과 같습니다.
  - 10Gbase-LR(단일 모드 광섬유) 송수신장치
  - 10Gbase-SR(다중 모드 광섬유) 송수신장치
  - SFP+ DAC 케이블
- 25G 작업의 경우 SFP28 옵션을 사용할 수 있습니다. 그러한 예는 다음과 같습니다.
  - 25Gbase-LR(단일 모드 광섬유) 송수신장치
  - 25Gbase-SR(다중 모드 광섬유) 송수신장치
  - SFP28 DAC 케이블

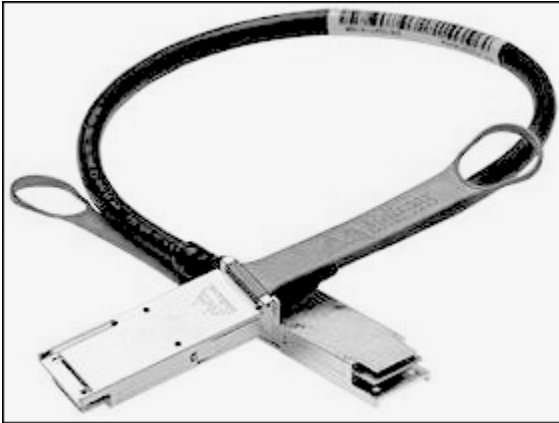


## QSFP

이 포트는 스토리지 최적화 디바이스에 40G QSFP+ 인터페이스, 컴퓨팅 최적화 디바이스에 40/50/100G QSFP+ 인터페이스를 제공합니다. 두 가지 모두 QSFP+ 송수신장치 모듈 및 DAC 케이블과 호환됩니다. 사용자는 자신의 송수신장치 또는 DAC 케이블을 제공해야 합니다. 예는 다음과 같습니다.



- 40Gbase-LR4(단일 모드 광섬유) 송수신장치
- 40Gbase-SR4(다중 모드 광섬유) 송수신장치
- QSFP+ DAC

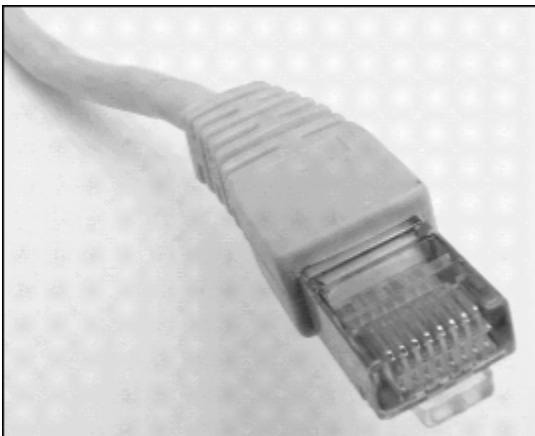


## RJ45

이 포트는 1Gbase-TX/10Gbase-TX 작업을 제공하며, RJ45 커넥터로 종단처리된 UTP 케이블을 통해 연결됩니다. Snowball Edge 디바이스에는 2개의 RJ45 포트가 있습니다. 사용할 포트 하나를 선택합니다.

1G 작업은 깜박이는 주황색 등에 의해 표시됩니다. 1G 작업은 Snowball Edge 디바이스에 대한 대규모 데이터 전송에는 권장되지 않습니다. 데이터 전송에 걸리는 시간이 대폭 늘어나기 때문입니다.

10G 작업은 깜박이는 녹색 등에 의해 표시되며, 최대 작동 거리가 55미터(180피트)인 Cat6A UTP 케이블을 필요로 합니다.



## Snowball Edge를 사용하기 위한 사전 조건

Snowball Edge를 시작하기 전에 AWS 계정이 없는 경우 계정에 가입해야 합니다. 또한 Snowball Edge와 함께 사용하도록 데이터 및 컴퓨팅 인스턴스를 구성하는 방법을 배우는 것이 좋습니다.

AWS Snowball Edge는 리전별 서비스입니다. 따라서 작업을 계획하기 전에는 해당 서비스가 AWS 리전에서 사용 가능한지 확인하세요. 위치 및 Amazon S3 버킷은 디바이스 주문 기능에 영향을 미치므로 동일 AWS 리전 또는 동일 국가 내에 있어야 합니다.

로컬 엣지 컴퓨팅 및 스토리지 작업을 위해 컴퓨팅 최적화 디바이스와 함께 Snowball Edge에서 Amazon S3 호환 스토리지를 사용하려면 주문 시 디바이스에서 S3 용량을 프로비저닝해야 합니다. Snowball Edge의 Amazon S3 호환 스토리지는 로컬 버킷 관리를 지원하므로 디바이스 또는 디바이스를 받은 후 디바이스 또는 클러스터에서 S3 버킷을 생성할 수 있습니다.

주문 프로세스의 일부로 AWS Identity and Access Management (IAM) 역할과 AWS Key Management Service (AWS KMS) 키를 생성합니다. KMS 키는 작업에 대한 잠금 해제 코드를 암호화하는 데 사용됩니다. IAM 역할 및 KMS 키 생성에 대한 자세한 내용은 [Snowball Edge 디바이스 주문 작업 생성을 참조하세요](#).

### Note

아시아 태평양(뭍바이) AWS 리전 서비스는 Amazon on Internet Services Private Limited(AISPL)에서 제공합니다. 아시아 태평양(뭍바이)에서 Amazon Web Services에 가입하는 방법에 대한 자세한 내용은 AISPL 가입을 AWS 리전참조하세요. <https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/manage-account-payment-aispl.html#aisplsignup>

### 주제

- [에 가입 AWS 계정](#)
- [관리자 액세스 권한이 있는 사용자 생성](#)
- [환경 정보](#)
- [특수 문자가 포함된 파일 이름 사용](#)
- [를 사용한 Amazon S3 암호화 AWS KMS](#)
- [서버 측 암호화를 통한 Amazon S3 암호화](#)
- [가져오기 및 내보내기 작업에 Snowball Edge에서 Amazon S3 어댑터를 사용하기 위한 사전 조건](#)

- [Snowball Edge에서 Amazon S3 호환 스토리지를 사용하기 위한 사전 조건](#)
- [Snowball Edge에서 컴퓨팅 인스턴스를 사용하기 위한 사전 조건](#)

## 에 가입 AWS 계정

이 없는 경우 다음 단계를 AWS 계정완료하여 생성합니다.

에 가입하려면 AWS 계정

1. <https://portal.aws.amazon.com/billing/signup>을 엽니다.
2. 온라인 지시 사항을 따릅니다.

등록 절차 중 전화 또는 텍스트 메시지를 받고 전화 키패드로 확인 코드를 입력하는 과정이 있습니다.

에 가입하면 AWS 계정AWS 계정 루트 사용자들이 생성됩니다. 루트 사용자에게는 계정의 모든 AWS 서비스 및 리소스에 액세스할 권한이 있습니다. 보안 모범 사례는 사용자에게 관리 액세스 권한을 할당하고, 루트 사용자만 사용하여 [루트 사용자 액세스 권한이 필요한 작업](#)을 수행하는 것입니다.

AWS 는 가입 프로세스가 완료된 후 확인 이메일을 보냅니다. 언제든지 <https://aws.amazon.com/>으로 이동하고 내 계정을 선택하여 현재 계정 활동을 보고 계정을 관리할 수 있습니다.

## 관리자 액세스 권한이 있는 사용자 생성

에 가입한 후 일상적인 작업에 루트 사용자를 사용하지 않도록 관리 사용자를 AWS 계정보호 AWS IAM Identity Center, AWS 계정 루트 사용자활성화 및 생성합니다.

보안 AWS 계정 루트 사용자

1. 루트 사용자를 선택하고 AWS 계정 이메일 주소를 입력하여 계정 소유자 [AWS Management Console](#)로 로그인합니다. 다음 페이지에서 비밀번호를 입력합니다.

루트 사용자를 사용하여 로그인하는 데 도움이 필요하면 AWS Sign-In User Guide의 [루트 사용자 로 로그인](#)을 참조하세요.

2. 루트 사용자의 다중 인증(MFA)을 활성화합니다.

지침은 IAM 사용 설명서의 [AWS 계정 루트 사용자\(콘솔\)에 대한 가상 MFA 디바이스 활성화를 참조하세요.](#)

## 관리자 액세스 권한이 있는 사용자 생성

1. IAM Identity Center를 활성화합니다.

지침은 AWS IAM Identity Center 사용 설명서의 [AWS IAM Identity Center 설정](#)을 참조하세요.

2. IAM Identity Center에서 사용자에게 관리자 액세스 권한을 부여합니다.

를 자격 증명 소스 IAM Identity Center 디렉터리로 사용하는 방법에 대한 자습서는 사용 AWS IAM Identity Center 설명서의 [기본값으로 사용자 액세스 구성을 IAM Identity Center 디렉터리](#) 참조하세요.

## 관리 액세스 권한이 있는 사용자로 로그인

- IAM Identity Center 사용자로 로그인하려면 IAM Identity Center 사용자를 생성할 때 이메일 주소로 전송된 로그인 URL을 사용합니다.

IAM Identity Center 사용자를 사용하여 로그인하는 데 도움이 필요하면 AWS Sign-In 사용 설명서의 [AWS 액세스 포털에 로그인](#)을 참조하세요.

## 추가 사용자에게 액세스 권한 할당

1. IAM Identity Center에서 최소 권한 적용 모범 사례를 따르는 권한 세트를 생성합니다.

지침은 AWS IAM Identity Center 사용 설명서의 [Create a permission set](#)를 참조하세요.

2. 사용자를 그룹에 할당하고, 그룹에 Single Sign-On 액세스 권한을 할당합니다.

지침은 AWS IAM Identity Center 사용 설명서의 [Add groups](#)를 참조하세요.

## 환경 정보

데이터 세트와 로컬 환경 설정 방법을 이해하면 데이터 전송을 완료하는 데 도움이 됩니다. 주문하기 전에 다음 사항을 고려하세요.

### 어떤 데이터를 전송하고 있나요?

많은 수의 작은 파일을 전송하는 것은 제대로 작동하지 않습니다 AWS Snowball Edge. 이는 Snowball Edge가 각 개별 객체를 암호화하기 때문입니다. 소용량 파일에는 크기가 1MB 미만인 파일이 포함됩니다. AWS Snowball Edge 디바이스로 전송하기 전에 압축하는 것이 좋습니다. 각 디렉터리 내 파일이나 디렉터리를 500,000개 이하로 유지하는 것도 좋습니다.

## 전송 중에 데이터에 액세스할 수 있습니까?

정적 데이터 세트를 보유하는 것이 중요합니다. 즉 전송 중에 데이터에 액세스하는 사용자나 시스템이 없어야 합니다. 그렇지 않으면 체크섬 불일치로 인해 파일 전송이 실패할 수 있습니다. 파일은 전송되지 않으며 Failed로 표시됩니다.

데이터 손상을 방지하려면 데이터를 전송하는 동안 AWS Snowball Edge 디바이스의 연결을 해제하거나 네트워크 설정을 변경하면 안 됩니다. 파일은 디바이스에 기록되는 동안 정적 상태에 있어야 합니다. 파일이 디바이스에 쓰여지는 동안 수정된 파일은 읽기/쓰기 충돌을 일으킬 수 있습니다.

## 네트워크가 AWS Snowball Edge 데이터 전송을 지원하나요?

Snowball Edge는 RJ45, SFP+ 또는 QSFP+ 네트워킹 어댑터를 지원합니다. 스위치가 기가비트 스위치인지 확인하세요. 스위치 브랜드에 따라 기가비트 또는 10/100/1000으로 표시될 수 있습니다. Snowball Edge 디바이스는 메가비트 스위치 또는 10/100 스위치를 지원하지 않습니다.

## 특수 문자가 포함된 파일 이름 사용

객체 이름에 특수 문자가 포함된 경우 오류가 발생할 수 있다는 점에 유의해야 합니다. Amazon S3에서는 특수 문자를 허용하지만 다음 문자는 사용하지 않는 것이 좋습니다.

- 백슬래시('\')
- 왼쪽 중괄호('{')
- 오른쪽 중괄호('}')
- 왼쪽 대괄호('[')
- 오른쪽 대괄호(']')
- '보다 작음' 기호('<')
- '보다 큼' 기호('>')
- 인쇄되지 않는 ASCII 문자(128~255 사이의 10진수)
- 캐럿('^')
- 백분율 문자('%')
- 억음 악센트 기호('')
- 인용 부호
- 물결표('~')
- '파운드' 문자('#')

- 세로 막대/파이프('|')

파일에 객체 이름에 이러한 문자가 하나 이상 있는 경우 객체를 AWS Snowball Edge 디바이스에 복사하기 전에 객체 이름을 바꿉니다. 파일 이름에 공백이 있는 Windows 사용자는 개별 객체를 복사하거나 재귀 명령을 실행할 때 주의해야 합니다. 명령에서 이름에 공백이 포함된 객체의 이름은 인용 부호로 묶습니다. 이러한 파일의 예시는 다음과 같습니다.

운영 체제	파일 이름: test file.txt
Windows	"C:\Users\ <username>\desktop\test file.txt"</username>
iOS	/Users/<username>/test\ file.txt
Linux	/home/<username>/test\ file.txt

**Note**

객체 메타데이터 중 객체 이름과 크기만이 유일하게 전송됩니다.

## 를 사용한 Amazon S3 암호화 AWS KMS

기본 AWS 관리형 또는 고객 관리형 암호화 키를 사용하여 데이터를 가져오거나 내보낼 때 데이터를 보호할 수 있습니다.

### AWS KMS 관리형 키와 함께 Amazon S3 기본 버킷 암호화 사용

를 사용하여 AWS 관리형 암호화를 활성화하려면 AWS KMS

1. <https://console.aws.amazon.com/s3/>에서 S3 콘솔을 엽니다.
2. 암호화할 Amazon S3 버킷을 선택합니다.
3. 오른쪽에 나타나는 마법사에서 속성을 선택합니다.
4. 기본 암호화 상자에서 비활성화됨(이 옵션은 회색으로 표시됨)을 선택하여 기본 암호화를 활성화합니다.
5. 암호화 방법으로 AWS-KMS를 선택한 다음 사용할 KMS 키를 선택합니다. 이 키는 버킷에 있는 객체를 암호화하는 데 사용됩니다.

## 6. 저장을 선택합니다.

Snowball Edge 작업을 생성한 후 그리고 데이터를 가져오기 전에 기존 IAM 역할 정책에 명령문을 추가합니다. 이 역할은 주문 프로세스 중에 생성됩니다. 작업 유형에 따라 기본 역할은 Snowball-import-s3-only-role 또는 Snowball-export-s3-only-role과 유사한 이름을 갖습니다.

이러한 문의 예시는 다음과 같습니다.

### 데이터 가져오기

AWS KMS 관리형 키를 사용한 서버 측 암호화(SSE-KMS)를 사용하여 가져오기 작업과 연결된 Amazon S3 버킷을 암호화하는 경우 IAM 역할에 다음 명령문도 추가해야 합니다.

#### Example Snowball IAM 역할 가져오기 예시

```
{
  "Effect": "Allow",
  "Action": [
    "kms: GenerateDataKey",
    "kms: Decrypt"
  ],
  "Resource": "arn:aws:kms:us-west-2:123456789012:key/abc123a1-abcd-1234-efgh-111111111111"
}
```

### 데이터 내보내기

AWS KMS 관리형 키와 함께 서버 측 암호화를 사용하여 내보내기 작업과 연결된 Amazon S3 버킷을 암호화하는 경우 IAM 역할에 다음 명령문도 추가해야 합니다.

#### Example Snowball IAM 역할 내보내기

```
{
  "Effect": "Allow",
  "Action": [
    "kms:Decrypt"
  ],
  "Resource": "arn:aws:kms:us-west-2:123456789012:key/abc123a1-abcd-1234-efgh-111111111111"
}
```

## AWS KMS 고객 키와 함께 S3 기본 버킷 암호화 사용

KMS 키를 통한 기본 Amazon S3 버킷 암호화를 사용하여 가져오고 내보내는 데이터를 보호할 수 있습니다.

### 데이터 가져오기

를 사용하여 고객 관리형 암호화를 활성화하려면 AWS KMS

1. 에 로그인 AWS Management Console 하고 <https://console.aws.amazon.com/kms> AWS Key Management Service (AWS KMS) 콘솔을 엽니다.
2. 를 변경하려면 페이지 오른쪽 상단에 있는 리전 선택기를 AWS 리전사용합니다.
3. 왼쪽 탐색 창에서 고객 관리형 키를 선택한 다음 사용하려는 버킷과 연결된 KMS 키를 선택합니다.
4. 키 정책이 아직 확장되지 않은 경우 키 정책을 확장합니다.
5. 키 사용자 섹션에서 추가를 선택하고 IAM 역할을 검색합니다. IAM 역할을 선택하고 추가를 선택합니다.
6. 또는 정책 보기로 전환을 선택하여 키 정책 문서를 표시하고 키 정책에 문을 추가할 수 있습니다. 정책의 예시는 다음과 같습니다.

### Example AWS KMS 고객 관리형 키에 대한 정책

```
{
  "Sid": "Allow use of the key",
  "Effect": "Allow",
  "Principal": {
    "AWS": [
      "arn:aws:iam::111122223333:role/snowball-import-s3-only-role"
    ]
  },
  "Action": [
    "kms:Decrypt",
    "kms:GenerateDataKey"
  ],
  "Resource": "*"
}
```

이 정책을 AWS KMS 고객 관리형 키에 추가한 후에는 Snowball 작업과 연결된 IAM 역할도 업데이트해야 합니다. 기본적으로 해당 역할은 snowball-import-s3-only-role입니다.



## Example Snowball IAM 역할 가져오기

```
{
  "Effect": "Allow",
  "Action": [
    "kms: GenerateDataKey",
    "kms: Decrypt"
  ],
  "Resource": "arn:aws:kms:us-west-2:123456789012:key/abc123a1-abcd-1234-efgh-111111111111"
}
```

자세한 내용은 [에 대한 자격 증명 기반 정책\(IAM 정책\) 사용 AWS Snowball Edge](#) 단원을 참조하십시오.

사용 중인 KMS 키는 다음과 같습니다.

```
"Resource": "arn:aws:kms:region:AccountID:key/*"
```

## 데이터 내보내기

### Example AWS KMS 고객 관리형 키에 대한 정책

```
{
  "Sid": "Allow use of the key",
  "Effect": "Allow",
  "Principal": {
    "AWS": [
      "arn:aws:iam::111122223333:role/snowball-import-s3-only-role"
    ]
  },
  "Action": [
    "kms:Decrypt",
    "kms:GenerateDataKey"
  ],
  "Resource": "*"
}
```

이 정책을 AWS KMS 고객 관리형 키에 추가한 후에는 Snowball 작업과 연결된 IAM 역할도 업데이트해야 합니다. 기본적으로 해당 역할은 다음과 같습니다.

snowball-export-s3-only-role

## Example Snowball IAM 역할 내보내기

```
{
  "Effect": "Allow",
  "Action": [
    "kms: GenerateDataKey",
    "kms: Decrypt"
  ],
  "Resource": "arn:aws:kms:us-west-2:123456789012:key/abc123a1-abcd-1234-efgh-111111111111"
}
```

이 정책을 AWS KMS 고객 관리형 키에 추가한 후에는 Snowball 작업과 연결된 IAM 역할도 업데이트해야 합니다. 기본적으로 해당 역할은 snowball-export-s3-only-role입니다.

## 서버 측 암호화를 통한 Amazon S3 암호화

AWS Snowball Edge 는 Amazon S3 관리형 암호화 키(SSE-S3)를 사용한 서버 측 암호화를 지원합니다. 서버 측 암호화는 저장 데이터를 보호하기 위한 것이고, SSE-S3에서는 Amazon S3의 저장 데이터를 보호하기 위해 강력한 멀티 팩터 암호화를 제공합니다. 자세한 내용은 Amazon Simple Storage Service 사용 설명서의 [Amazon S3 관리형 암호화 키를 통한 서버 측 암호화\(SSE-S3\)를 사용하여 데이터 보호](#) 섹션을 참조하세요.

### Note

현재 AWS Snowball Edge 는 고객 제공 키(SSE-C)를 사용한 서버 측 암호화를 지원하지 않습니다. 그러나 해당 SSE 유형을 사용하여 가져온 데이터를 보호하고 싶을 수 있으며, 내보내려는 데이터에 해당 SSE 유형을 이미 사용하고 있을 수 있습니다. 이 경우에는 다음 사항에 유의해야 합니다.

- 가져오기 - SSE-C를 사용하여 S3으로 가져온 객체를 암호화하려면, SSE-KMS 또는 SSE-S3 암호화를 버킷 정책의 일부로 설정한 다른 버킷으로 해당 객체를 복사합니다.
- 내보내기 - SSE-C를 사용하여 암호화된 객체를 내보내려면, 먼저 서버 측 암호화를 제공하지 않거나 버킷 정책에 SSE-KMS 또는 SSE-S3가 지정되어 있는 다른 버킷으로 해당 객체를 복사합니다.

## 가져오기 및 내보내기 작업에 Snowball Edge에서 Amazon S3 어댑터를 사용하기 위한 사전 조건

디바이스를 사용하여 온프레미스 데이터 소스에서 클라우드로 또는 클라우드에서 온프레미스 데이터 스토리지로 데이터를 이동할 때 Snowball Edge에서 S3 어댑터를 사용할 수 있습니다. 자세한 내용은 [Snowball Edge와의 데이터 마이그레이션을 위해 Amazon S3 어댑터를 사용하여 파일 전송 단원을 참조](#)하십시오.

작업과 연결된 Amazon S3 버킷은 Amazon S3 Standard 스토리지 클래스를 사용해야 합니다. 첫 번째 작업을 생성하기 전에 다음에 유념하세요.

Amazon S3로 데이터를 가져오는 작업의 경우 다음 단계를 따릅니다.

- 전송할 파일 및 폴더의 이름이 Amazon S3의 [객체 키 명명 지침](#)에 따라 지정되었는지 확인합니다. 이름이 이러한 지침을 충족하지 않는 파일 또는 폴더는 Amazon S3로 가져오지 않습니다.
- Amazon S3로 가져오려는 데이터를 계획합니다. 자세한 내용은 [Snowball Edge를 사용하여 대규모 전송 계획](#) 단원을 참조하십시오.

Amazon S3에서 데이터를 내보내기 전에 다음 단계를 따릅니다.

- 작업을 생성할 때 내보낼 데이터를 이해합니다. 자세한 내용은 [Snowball Edge 디바이스로 데이터를 내보낼 때 Amazon S3 객체 키 사용](#) 단원을 참조하십시오.
- 파일 이름에 콜론(:)이 있는 파일의 경우 이러한 파일을 가져올 내보내기 작업을 생성하기 전에 Amazon S3에서 파일 이름을 변경합니다. 파일 이름에 콜론이 있는 파일은 Microsoft Windows Server로의 내보내기가 실패합니다.

## Snowball Edge에서 Amazon S3 호환 스토리지를 사용하기 위한 사전 조건

엣지 로케이션의 디바이스에 데이터를 저장하고 로컬 컴퓨팅 작업에 데이터를 사용하는 경우 Snowball Edge에서 Amazon S3 호환 스토리지를 사용합니다. 디바이스가 반환되면 로컬 컴퓨팅 작업에 사용되는 데이터는 Amazon S3로 가져오지 않습니다.

Amazon S3 호환 스토리지가 포함된 로컬 컴퓨팅 및 스토리지용 Snow 디바이스를 주문할 때는 다음 사항을 염두에 두세요.

- 디바이스를 주문할 때 Amazon S3 스토리지 용량을 프로비저닝합니다. 따라서 디바이스를 주문하기 전에 필요한 스토리지를 고려해야 합니다.

- Snowball Edge 디바이스를 주문하는 대신 디바이스를 받은 후 디바이스에서 Amazon S3 버킷을 생성할 수 있습니다.
- Snowball Edge에서 Amazon S3 호환 스토리지를 사용하려면 AWS CLI (v2.11.15 이상), Snowball Edge 클라이언트 또는의 최신 버전을 다운로드 AWS OpsHub 하여 컴퓨터에 설치해야 합니다.
- 디바이스를 받은 후이 설명서의 Snowball Edge에서 Amazon S3 호환 스토리지 [사용에 따라 Snowball Edge에서 Amazon S3 호환 스토리지](#)를 구성, 시작 및 사용합니다.

## Snowball Edge에서 컴퓨팅 인스턴스를 사용하기 위한 사전 조건

sbe1, sbe-c 및 인스턴스 유형을 AWS Snowball Edge 사용하여 호스팅되는 Amazon EC2-compatible 컴퓨팅 sbe-g 인스턴스를 실행할 수 있습니다.

- sbe1 인스턴스 유형은 Snowball Edge Edge 스토리지 최적화 옵션이 있는 디바이스에서 작동합니다.
- sbe-c 인스턴스 유형은 Snowball Edge Edge 컴퓨팅 최적화 옵션이 있는 디바이스에서 작동합니다.


Snowball Edge Edge 디바이스 옵션에서 지원되는 모든 컴퓨팅 인스턴스 유형은 AWS Snowball Edge 디바이스에 고유합니다. 이에 상응하는 클라우드 기반 인스턴스와 마찬가지로 이들 인스턴스 역시 시작하려면 Amazon Machine Image(AMI)가 필요합니다. Snowball Edge Edge 작업을 생성하기 전에 인스턴스의 AMI를 선택합니다.

Snowball Edge Edge에서 컴퓨팅 인스턴스를 사용하려면 Snowball Edge 디바이스를 주문하는 작업을 생성하고 AMIs를 지정합니다. AWS Snowball Edge 관리 콘솔, AWS Command Line Interface (AWS CLI) 또는 AWS SDKs. 일반적으로, 인스턴스를 사용하기 위해 작업을 생성하기 전에 수행해야 하는 정리 작업 사전 조건이 일부 존재합니다.

컴퓨팅 인스턴스를 사용하는 작업의 경우 작업에 AMI를 추가하려면 먼저 AWS 계정에 AMI가 있어야 하며 지원되는 이미지 유형이어야 합니다. 현재 지원되는 AMI는 다음 운영 체제를 기반으로 합니다.

- [Amazon Linux 2](#)
- [CentOS 7 \(x86\\_64\) - HVM 업데이트 포함](#)
- Ubuntu 16.04 LTS - Xenial (HVM)
- [Ubuntu 20.04 LTS - Focal](#)
- [Ubuntu 22.04 LTS - Jammy](#)
- [Microsoft Windows Server 2012 R2](#)

- [Microsoft Windows Server 2016](#)
- [Microsoft Windows Server 2019](#)

 Note

Ubuntu 16.04 LTS - Xenial(HVM) 이미지는에서 더 이상 지원되지 AWS Marketplace 않지만 Amazon EC2 VM Import/Export를 통해 Snowball Edge 디바이스에서 사용할 수 있으며 AMIs.

이들 이미지는 [AWS Marketplace](#)에서 얻을 수 있습니다.

SSH를 사용하여 Snowball Edge에서 실행 중인 인스턴스에 연결하는 경우 자체 키 페어를 사용하거나 Snowball Edge에서 키 페어를 생성할 수 있습니다. AWS OpsHub 를 사용하여 디바이스에서 키 페어를 생성하려면 [섹션을 참조하세요에서 EC2-compatible 인스턴스의 키 페어 작업 AWS OpsHub](#). AWS CLI 를 사용하여 디바이스에서 키 페어를 생성하려면 create-key-pair의 [섹션을 참조하세요Snowball Edge에서 지원되는 EC2-compatible AWS CLI 명령 목록](#). 키 페어 및 Amazon Linux 2에 대한 자세한 내용은 Amazon EC2 사용 설명서에서 [Amazon EC2 키 페어 및 Linux 인스턴스](#) 섹션을 참조하세요.

디바이스에서 컴퓨팅 인스턴스를 사용하는 것에 대한 자세한 내용은 [Snowball Edge에서 Amazon EC2-compatible 컴퓨팅 인스턴스 사용](#) 섹션을 참조하세요.

# AWS Snowball Edge 작동 방식

AWS Snowball Edge 디바이스에서 소유 AWS하며 사용 중인 온프레미스 위치에 상주합니다.

AWS Snowball Edge 디바이스에 사용할 수 있는 작업 유형은 세 가지입니다. 작업 유형은 저마다 사용 사례가 다르지만, 디바이스 주문, 수령 및 반환 방법의 작업 흐름은 모든 작업 유형에서 동일합니다. 작업 유형에 관계없이 모든 작업은 작업 완료 후 NIST(National Institute of Standards and Technology) 800-88 표준의 데이터 삭제 단계를 거칩니다.

## 공유 워크플로우

1. 작업 생성 - 각 작업은 AWS Snow 패밀리 관리 콘솔에서 생성되거나 작업 관리 API를 통해 프로그래밍 방식으로 생성됩니다. 콘솔이나 API를 통해 작업 상태를 추적할 수 있습니다.
2. 작업을 위해 디바이스 준비 작업을 위해 AWS Snowball Edge 디바이스를 준비합니다. 현재 작업 상태는 Snowball 준비 중입니다. 이 준비 프로세스는 디바이스 주문 작업이 생성된 시점으로부터 최대 4주가 걸릴 수 있습니다. 원활한 전환을 위해 이 타임라인을 프로젝트 계획에 반영해야 합니다.
3. 리전의 운송업체가 디바이스를 고객에게 배송 - 운송업체가 해당 위치에서 제품을 수거해 가며, 현재 작업 상태는 배송 중입니다. 콘솔 또는 작업 관리 API에서 추적 번호와 추적 웹 사이트에 대한 링크를 찾을 수 있습니다. 해당 리전의 운송업체에 대한 자세한 내용은 [Snowball Edge의 배송 고려 사항](#) 섹션을 참조하세요.
4. 디바이스 받기 - 며칠 후 리전의 통신 사업자가 작업을 생성할 때 제공한 주소로 AWS Snowball Edge 디바이스를 전송하고 작업 상태가 사용자에게 배달됨으로 변경됩니다. 디바이스 자체가 배송용 컨테이너이므로 디바이스가 도착했을 때 상자에 든 상태로 배송되지 않은 것을 알 수 있습니다.
5. 보안 인증 정보 받기 및 또는 Snow Edge 클라이언트 다운로드 - 보안 인증 정보, 작업 매니페스트, 매니페스트 잠금 해제 코드를 받고 Snowball Edge 클라이언트를 다운로드하는 등 데이터 전송을 준비합니다.
  - Snowball Edge 클라이언트는 디바이스에서 온프레미스 데이터 대상 주소로 전송되는 데이터의 흐름을 관리하는 데 사용하는 도구입니다.

[AWS Snowball Edge 리소스](#) 페이지에서 Snowball Edge 클라이언트를 다운로드 및 설치합니다.

[AWS Snowball Edge 리소스](#) 페이지에서 Snowball Edge 클라이언트를 다운로드하고 소유 중인 강력한 워크스테이션에 설치해야 합니다.

- 디바이스에 대한 액세스를 인증하는 데 사용되는 매니페스트는 암호화되어 있으며, 잠금 해제 코드로만 암호화를 해제할 수 있습니다. 디바이스가 해당 위치의 온프레미스에 있을 때 콘솔이나 작업 관리 API에서 매니페스트를 가져올 수 있습니다.

- 잠금 해제 코드는 매니페스트의 암호를 해제하는 데 사용되는 29자 코드입니다. 콘솔이나 작업 관리 API에서 잠금 해제 코드를 가져올 수 있습니다. 디바이스가 사용자의 시설에 있는 동안 무단 액세스를 방지하기 위해 잠금 해제 코드를 적어서 매니페스트와 분리된 장소에 보관하는 것이 좋습니다.
6. 하드웨어 배치 - 디바이스를 데이터 센터로 옮기고 케이스상의 지침에 따라 개봉합니다. 디바이스를 전원 및 로컬 네트워크에 연결합니다.
  7. 디바이스 전원 켜기 - 이제 LCD 디스플레이 위에 있는 전원 버튼을 눌러 디바이스를 켭니다. 몇 분 정도 기다리면 준비 완료 화면이 나타납니다.
  8. 디바이스의 IP 주소 가져오기 - LCD 디스플레이에는 연결 탭이 있습니다. 이 탭을 탭하고 AWS Snowball Edge 디바이스의 IP 주소를 가져옵니다.
  9. Snowball Edge 클라이언트를 사용하여 디바이스 잠금 해제 - Snowball Edge 클라이언트를 사용하여 AWS Snowball Edge 디바이스 잠금을 해제하는 경우 디바이스의 IP 주소, 매니페스트 경로 및 잠금 해제 코드를 입력합니다. Snowball Edge 클라이언트는 매니페스트의 암호화를 해제하여 디바이스에 대한 액세스를 인증하는 데 매니페스트를 사용합니다.
  10. 디바이스 사용 - 디바이스가 실행됩니다. 이를 사용하여 Amazon S3 어댑터 또는 NFS(Network File System) 탑재 지점으로 데이터를 전송하거나 Snowball Edge에서 Amazon S3 호환 스토리지를 사용하는 로컬 컴퓨팅 및 스토리지로 데이터를 전송할 수 있습니다.
  11. 디바이스 반품 준비 - 온프레미스 위치에서 디바이스 작업을 완료한 후 LCD 디스플레이 위의 전원 버튼을 누릅니다. 디바이스의 전원이 꺼지는 데 20초 정도 걸립니다. 디바이스의 플러그를 뽑고 전원 케이블을 디바이스 상단의 케이블 누크에 넣은 다음, 디바이스의 도어 세 개를 모두 닫습니다. 이제 디바이스는 반환 준비가 되었습니다.
  12. 리전의 통신사가 디바이스를 로 반환 AWS - 통신사에 AWS Snowball Edge 디바이스가 있으면 작업의 상태가 전송 중 AWS이 됩니다.

#### Note

내보내기 및 클러스터 작업의 경우 추가 단계가 있습니다. 자세한 내용은 [Snowball Edge 내보내기 작업 작동 방식](#) 및 [Snowball Edge 클러스터링된 로컬 컴퓨팅 및 스토리지 작업의 작동 방식](#) 섹션을 참조하세요.

## 주제

- [Snowball Edge 가져오기 작업 작동 방식](#)
- [Snowball Edge 내보내기 작업 작동 방식](#)
- [Snowball Edge 로컬 컴퓨팅 및 스토리지 작업의 작동 방식](#)

- [Snowball Edge 동영상 및 블로그](#)

## Snowball Edge 가져오기 작업 작동 방식

각 가져오기 작업은 하나의 Snowball 어플라이언스를 사용합니다. AWS Snow 패밀리 관리 콘솔 또는 작업 관리 API에서 Snowball Edge 디바이스를 주문하는 작업을 생성하면 Snowball이 배송됩니다. 며칠 뒤 배송된 Snowball Edge 디바이스를 네트워크에 연결하고, Amazon S3로 가져오려는 데이터를 디바이스로 전송합니다. 데이터 전송이 완료되면 Snowball을 로 다시 배송 AWS하고 데이터를 Amazon S3로 가져옵니다.

### Important

S3 Object Lock을 켜고 기본 보관 설정을 활성화한 경우 가져오기 프로세스가 Snow 디바이스에서 Amazon S3의 버킷에 쓸 수 없습니다. Amazon S3 Object Lock이 활성화되면 이를 비활성화하거나 버킷의 버전 관리를 일시 중지할 수 없습니다. 버킷에 기본 보존 설정이 활성화된 S3 객체 잠금이 있는 경우 Snowball Edge를 반환하기 전에 S3 객체 잠금의 보존 설정을 비활성화합니다. 에서 디바이스에서 데이터를 가져온 후 버킷에서 재연결 설정을 다시 AWS활성화합니다. 자세한 내용은 [S3 객체의 보관 기간 설정 또는 수정](#)을 참조하세요.

또한 버킷의 IAM 정책이 버킷에 쓰기를 금지하는 경우에도 가져오기 프로세스가 Amazon S3의 버킷에 쓸 수 없습니다. 자세한 내용은 [Amazon S3의 자격 증명 및 액세스 관리](#)를 참조하세요.

## Snowball Edge 내보내기 작업 작동 방식

각 내보내기 작업은 원하는 수의 AWS Snowball Edge 디바이스를 사용할 수 있습니다. 목록에 단일 디바이스에 담을 수 있는 것보다 많은 데이터가 포함되어 있는 경우 여러 디바이스가 제공됩니다. 각 작업 파트에는 디바이스가 정확히 하나씩 연결되어 있습니다. 작업 파트가 모두 생성되면 첫 번째 작업 파트가 Snowball 준비 중 상태가 됩니다.

### Note

작업을 여러 파트로 분할하는 목록 작업은 Amazon S3의 기능이며, Amazon S3 작업과 동일하게 요금이 청구됩니다.



이후 데이터를 디바이스로 내보내기 시작합니다. 데이터를 내보내는 데 필요한 시간은 데이터 세트의 특성에 따라 달라집니다. 예를 들어 크기가 작은 파일(10MB 미만)을 많이 내보내는 경우 훨씬 더 오래 걸립니다. 내보내기가 완료되면 AWS는 리전의 운송업체가 디바이스를 픽업할 수 있도록 준비합니다. 디바이스가 도착하면 AWS Snowball Edge 디바이스를 네트워크에 연결하고 디바이스의 데이터를 네트워크의 스토리지로 전송합니다.

데이터 전송이 완료되면 디바이스를 로 다시 배송합니다 AWS. 내보내기 작업용 디바이스를 수령 하면 해당 디바이스를 완전히 삭제 처리합니다. 이 삭제는 NIST(National Institute of Standards and Technology) 800-88 표준에 따른 것입니다. 이 단계가 완료되면 해당 작업 파트가 완료됩니다.

- 키리스팅

S3 버킷의 객체를 내보내기 전에 버킷을 스캔합니다. 스캔 후 버킷을 변경하면 누락되거나 변경된 객체를 스캔하기 때문에 작업이 지연될 수 있습니다.

- S3 Glacier Flexible Retrieval

는 S3 Glacier 스토리지 클래스에 있는 객체를 내보낼 AWS Snowball Edge 수 없다는 점에 유의해야 합니다. 버킷에 있는 객체를 성공적으로 AWS Snowball Edge 이 내보내려면 먼저 이러한 객체를 복원해야 합니다.

## Snowball Edge 로컬 컴퓨팅 및 스토리지 작업의 작동 방식

Amazon EKS Anywhere on Snow에서 AWS EC2-compatible 컴퓨팅 인스턴스 또는 Kubernetes 컨테이너를 실행하여 AWS Snowball Edge 디바이스의 로컬 컴퓨팅 및 스토리지 기능을 사용할 수 있습니다. 컴퓨팅 기능을 위해 데이터 스토리지는 Snowball Edge의 Amazon S3 호환 스토리지에서 제공됩니다.

Snowball Edge 디바이스에 Amazon S3 버킷을 생성하여, 로컬 데이터 액세스, 로컬 데이터 처리, 데이터 레지던시가 필요한 애플리케이션에서 온프레미스로 객체를 쉽게 저장하고 검색할 수 있습니다. Snowball Edge의 Amazon S3 호환 스토리지는 Amazon S3 APIs를 SNOW사용하고 여러 Snowball Edge 디바이스에 데이터를 내구성 있고 중복적으로 저장하도록 설계된 새로운 스토리지 클래스인 를 제공합니다. 버킷 수명 주기 정책, 암호화, 태그 지정을 포함하여 Amazon S3에서와 같이 Snowball Edge 버킷에서 동일한 API 및 기능을 사용할 수 있습니다. 디바이스 또는 디바이스가 로 반환되면 Snowball Edge의 Amazon S3 호환 스토리지에 생성되거나 저장된 AWS모든 데이터가 삭제됩니다. 자세한 내용은 [로컬 컴퓨팅 및 스토리지 전용 작업](#) 섹션을 참조하세요.

자세한 내용은 [Snowball Edge 디바이스를 사용하여 로컬 컴퓨팅 및 스토리지 기능을 제공하는 방법에 대한 정보](#) 단원을 참조하십시오.

## Snowball Edge 클러스터링된 로컬 컴퓨팅 및 스토리지 작업의 작동 방식

클러스터 작업은 로컬 스토리지 및 컴퓨팅 전용 특수 작업입니다. 향상된 데이터 내구성과 스토리지 용량이 필요한 워크로드에 적합합니다. 자세한 내용은 [Snowball Edge 디바이스 클러스터에 로컬 스토리지를 제공하는 작업에 대한 정보](#) 단원을 참조하십시오.

### Note

독립형 로컬 스토리지 및 컴퓨팅 작업과 마찬가지로 별도의 가져오기 작업의 일부로 추가 디바이스를 주문하지 않고는 클러스터에 저장된 데이터를 Amazon S3로 가져올 수 없습니다. 이 디바이스를 주문하면 클러스터에서 디바이스로 데이터를 전송하고, 가져오기 작업용 디바이스를 반환할 때 데이터를 가져올 수 있습니다.

클러스터에는 노드라고 하는 3~16개의 AWS Snowball Edge 디바이스가 있습니다. 해당 리전의 운송 업체로부터 노드를 받으면 모든 노드를 전원과 네트워크에 연결하여 IP 주소를 획득합니다. 이러한 IP 주소가 있으면 하나의 잠금 해제 명령과 노드 중 하나의 IP 주소를 사용하여 클러스터의 모든 노드를 한 번에 잠금 해제합니다. 자세한 내용은 [Snowball Edge 클라이언트 구성 및 사용](#) 단원을 참조하십시오.

또는 Snowball Edge의 Amazon S3 호환 스토리지와 다른 노드 간에 분산된 데이터를 사용하여 잠금 해제된 클러스터에 데이터를 쓸 수 있습니다.

클러스터 작업을 마치면 모든 노드를 다시 배송합니다 AWS. 클러스터 노드를 수신하면 Snowball을 완전히 삭제합니다. 이 삭제는 NIST(National Institute of Standards and Technology) 800-88 표준에 따른 것입니다.

## Snowball Edge 동영상 및 블로그

- [AWS Snowball Edge 디바이스에서 snow-transfer-tool을 사용하여 혼합 파일 크기 마이그레이션](#)
- [AWS Snowball Edge 데이터 마이그레이션](#)
- [AWS OpsHub for Snow Family](#)
- [Novetta는 IoT와 기계 학습을 재해 복구에 대응하는 Edge에 배포](#)
- [DMS 및를 사용하여 대규모 데이터베이스 마이그레이션 활성화 AWS Snowball Edge](#)
- [를 사용한 데이터 마이그레이션 모범 사례 AWS Snowball Edge](#)
- [AWS Snowball Edge resources](#)

- [AWS Snowball Edge 컴퓨팅 최적화 디바이스의 Amazon S3 호환 스토리지 이제 정식 출시](#)
- [Snowball Edge 디바이스의 Amazon S3 호환 스토리지 시작하기 AWS](#)

# Snowball Edge 디바이스의 장기 요금

Snowball Edge 디바이스를 주문할 때 사용 사례에 가장 적합한 가격 옵션을 선택할 수 있습니다. 요금은 두 가지 방법으로 제공됩니다. 하나는 디바이스를 사용하는 날짜에 따른 온디맨드 지불 또는 디바이스 유형에 따른 월별, 1년 또는 3년 단위의 장기 선불입니다. 디바이스 사용이 중단되지 않도록 이전 기간이 종료될 때 새 선불 기간이 시작되도록 1년 또는 3년 기간의 장기 요금제 옵션을 자동 갱신하도록 선택할 수 있습니다. 월별 장기 요금제 옵션은 디바이스를 소지하고 있는 동안 자동으로 갱신됩니다. 디바이스 주문에 대한 자세한 내용은 이 설명서의 [Snowball Edge 디바이스 주문 작업 생성을 참조하세요](#).

장기 요금제를 사용하면 예산 편의성 외에도 운영 요구 사항이 변경되는 요금 책정 기간에 Snowball Edge 디바이스를 교체할 수 있습니다. 예를 들어 새 디바이스에 새 AMI 또는 Amazon S3의 새 데이터가 포함되도록 디바이스를 교체하거나 장애가 발생한 디바이스를 교체하도록 요청할 수 있습니다. [장기 요금제 기간 동안의 Snowball Edge 디바이스 교체](#)(를) 참조하세요.

## Note

하드웨어 또는 Snow 서비스에 기인하는 소프트웨어 문제가 아닌 다른 이유로 1년 또는 3년 약정 요금제 미만의 AWS Snowball Edge 디바이스를 교체하거나 교체하도록 요청하는 경우 Device Cycling 요금이 부과됩니다. 이 디바이스 사이클링 요금은 구성에 대한 월별 요금 (Snowball Edge Compute Optimized) 또는 온디맨드 작업 요금으로 결정됩니다.

장기 요금제에 대한 자세한 내용은 [Optimizing cost with long-term pricing options for AWS Snowball Edge](#) 섹션을 참조하세요. 에 대한 AWS Snowball Edge 요금은 [AWS Snowball Edge 요금](#)을 AWS 리전 참조하세요.

## 장기 요금제 기간 동안의 Snowball Edge 디바이스 교체

장기 요금제 기간 동안 Snowball Edge 디바이스를 교체하려면 새 디바이스를 주문하고 현재 디바이스를 즉시 반품해야 합니다.

1. 교체 Snowball Edge 디바이스를 위한 새 작업을 생성합니다. 교체 디바이스는 작업 유형이 같고 사용 중인 디바이스와 동일한 컴퓨팅 및 스토리지 옵션을 갖추고 있어야 합니다. 이 설명서의 [Snowball Edge 디바이스 주문 작업 생성을 참조하세요](#).

2. 가지고 있는 디바이스를 즉시 반납합니다. [Snowball Edge 전원 끄기](#) 및 단원을 참조하십시오. [Snowball Edge 디바이스 반환](#). AWS 는 디바이스 교체 물류를 관리하며, 이 스왑에 대해 디바이스 사이클링 요금이 부과됩니다.

## Snowball Edge의 배송 고려 사항

Snowball Edge 디바이스 주문 작업을 생성할 때 배송 주소를 제공하고 배송 속도를 선택합니다. 이 배송 기간은 작업을 생성한 날로부터 디바이스를 수신하는 데 걸리는 시간을 나타내지 않습니다. 대신 디바이스가 AWS 와 배송 주소 간에 전송 중인 시간을 나타냅니다.

디바이스가 배송되기 전에 작업에 맞게 디바이스를 프로비저닝하고 준비하는 데 최대 4주가 걸릴 수 있습니다. 원활한 전환을 위해 이 타임라인을 프로젝트 계획에 반영해야 합니다. AWS 가 디바이스 배송을 준비하는 동안 통째로 작업 상태를 모니터링할 수 있습니다 AWS Snow 패밀리 관리 콘솔. 자세한 내용은 [Snowball Edge 작업의 상태](#) 단원을 참조하십시오.

### Note

선택한 배송 속도가 디바이스를 AWS 사용자에게 보내고 사용자가 디바이스를 반환할 때 적용됩니다 AWS.

Snowball Edge 디바이스는 디바이스가 주문된 AWS 리전 내에서 데이터를 가져오거나 내보내는 데만 사용할 수 있습니다.

Snowball Edge 디바이스 주문 작업을 생성할 때 배송 속도를 선택하고 배송 주소를 입력하는 방법에 대한 자세한 내용은 [섹션을 참조하세요](#) [보안, 배송 및 알림 기본 설정 선택](#). Snowball Edge 디바이스에 반환하는 방법에 대한 자세한 내용은 [섹션을 참조하세요](#) [Snowball Edge 디바이스 반환](#).

배송 요금에 대한 자세한 내용은 [AWS Snowball Edge 요금](#)을 참조하십시오.

## Snowball Edge에 대한 리전 기반 배송 제한

Snowball Edge 디바이스 주문 작업을 생성하기 전에 AWS 리전 Amazon S3 데이터와 동일한에서 콘솔에 로그인해야 합니다.는 아시아 태평양(인도)에서 아시아 태평양(호주)으로 AWS 리전등 동일한 내의 국가 간에 Snowball Edge를 배송하지 AWS 않습니다.


유럽 연합(EU) 회원국에서는 국가 간 배송에 대한 예외가 있습니다. 유럽 AWS 리전에서의 데이터 전송의 경우 나열된 EU 회원국으로만 디바이스를 배송합니다.

오스트리아, 벨기에, 불가리아, 크로아티아, 사이프러스, 체코 공화국, 덴마크, 에스토니아, 핀란드, 프랑스, 독일, 그리스, 헝가리, 이탈리아, 아일랜드, 라트비아, 리투아니아, 룩셈부르크, 몰타, 네덜란드, 폴란드, 포르투갈, 루마니아, 슬로바키아, 슬로베니아, 스페인, 스웨덴.

Snowball Edge는 디바이스가 주문된 AWS 리전에만 반환할 수 있습니다.

동일한 국가 내 국내 배송은 허용됩니다. 예시:

- 영국 리전에서 데이터 전송의 경우, 당사는 디바이스를 영국 국내로 배송합니다.
- 아시아 태평양(뭄바이)에서 데이터를 전송할 경우 인도 내에서만 디바이스를 배송합니다.

 Note

AWS 는 Snowball Edge를 사서함으로 배송하지 않습니다.

# Snowball Edge 시작하기

AWS Snowball Edge 디바이스를 사용하면 인터넷에 연결하는 것이 옵션이 아닐 수 있는 위치에서 AWS 클라우드 로컬 및 비용 효율적으로 스토리지 및 컴퓨팅 성능에 액세스할 수 있습니다. 또한 온프레미스 데이터 센터와 Amazon Simple Storage Service(S3) 간에 수백 테라바이트 또는 페타바이트의 데이터를 전송할 수 있습니다.

아래에서는 AWS Snow 패밀리 관리 콘솔에서 첫 번째 AWS Snowball Edge 디바이스 작업을 생성하여 완료하기 위한 일반 지침을 확인할 수 있습니다. 콘솔에는 가장 일반적인 워크플로우가 작업 유형별로 분리되어 있습니다. AWS Snowball Edge 디바이스의 특정 구성 요소에 대한 자세한 내용은 이 문서에 나옵니다. 서비스에 대한 개요는 [AWS Snowball Edge 작동 방식](#) 섹션을 참조하세요.

Snowball Edge가 배송되기 전에 작업에 맞게 프로비저닝하고 준비하는 데 최대 4주가 걸릴 수 있습니다. 원활한 전환을 위해 이 타임라인을 프로젝트 계획에 반영해야 합니다.

시작하기 전에 AWS Identity and Access Management (IAM)에서 AWS 계정 및 관리자 사용자를 생성해야 합니다. 자세한 내용은 [Snowball Edge를 사용하기 위한 사전 조건](#) 단원을 참조하세요.

## 주제

- [Snowball Edge 디바이스 주문 작업 생성](#)
- [Snowball Edge 주문 작업 취소](#)
- [에서 Snowball Edge 주문 작업 복제 AWS Snow 패밀리 관리 콘솔](#)
- [Snowball Edge 받기](#)
- [로컬 네트워크에 Snowball Edge 연결](#)
- [Snowball Edge에 액세스하기 위한 자격 증명 가져오기](#)
- [Snowball Edge 잠금 해제](#)
- [Snowball Edge에서 로컬 사용자 설정](#)
- [Snowball Edge 디바이스 재부팅](#)
- [Snowball Edge 전원 끄기](#)
- [Snowball Edge 디바이스 반환](#)
- [Snowball Edge 반품 배송](#)
- [Snowball Edge에서 가져오기 상태 모니터링](#)
- [데이터 전송 작업 완료 보고서 및 로그 가져오기](#)



# Snowball Edge 디바이스 주문 작업 생성

Snowball Edge 디바이스를 주문하려면에서 Snowball Edge 디바이스를 주문하는 작업을 생성합니다 AWS Snow 패밀리 관리 콘솔. 작업은 AWS 사용하여 고객이 Snowball Edge 디바이스를 사용하는 수명 주기를 설명하는 용어입니다. 작업은 디바이스를 주문할 때 시작되고, 디바이스를 AWS 준비하여 사용자에게 배송하고 사용자가 사용하는 경우 계속되며, 디바이스를 반환한 후 디바이스를 AWS 수신하고 처리한 후 완료됩니다. 작업은 내보내기, 가져오기, 로컬 컴퓨팅 및 스토리지 유형별로 분류됩니다. 자세한 내용은 [AWS Snowball Edge 작업 이해](#)를 참조하세요.

디바이스를 주문하는 작업을 생성한 후 AWS Snow 패밀리 관리 콘솔을 사용하여 작업 상태를 보고 디바이스를 배송할 AWS 준비를 하고 디바이스가 반환된 후 주문한 디바이스의 진행 상황을 모니터링할 수 있습니다. 자세한 내용은 [작업 상태](#)를 참조하세요. 에서 디바이스를 반환하고 처리한 후를 통해 작업 완료 보고서 및 로그에 액세스할 AWS 수 있습니다 AWS Snow 패밀리 관리 콘솔. 자세한 내용은 [콘솔에서 작업 완료 보고서 및 로그 가져오기](#)를 참조하세요.

또한 작업 관리 API를 사용하여 작업을 생성하고 관리할 수도 있습니다. 자세한 내용은 [AWS Snowball Edge API 참조](#)를 참조하십시오.

## 주제

- [작업 유형 선택](#)
- [컴퓨팅 및 스토리지 옵션 선택](#)
- [기능 및 옵션 선택](#)
- [보안, 배송 및 알림 기본 설정 선택](#)
- [작업 요약 검토 및 작업 생성](#)

## 작업 유형 선택

작업 생성의 첫 번째 단계는 필요한 작업 유형을 결정하고 AWS Snow 패밀리 관리 콘솔을 사용하여 작업 계획을 시작하는 단계입니다.

### 작업 유형 선택

1. 에 로그인 AWS Management Console하고를 엽니다 [AWS Snow 패밀리 관리 콘솔](#). 이에서 작업을 처음 생성하는 경우 AWS Snowball Edge 페이지 AWS 리전가 표시됩니다. 그렇지 않으면 기존 작업 목록이 표시됩니다.
2. 디바이스를 주문하는 첫 번째 작업인 경우 AWS Snowball Edge 디바이스 주문을 선택합니다. 500TB가 넘는 데이터를 마이그레이션하는 작업이 여러 개 있을 것으로 예상되면 500TB를 초과하

는 라지 데이터 마이그레이션 계획 생성을 선택합니다. 그렇지 않으면 왼쪽 탐색 표시줄에서 작업 생성을 선택합니다. 다음 단계를 선택하여 작업 계획 페이지를 엽니다.

3. 작업 이름 섹션의 작업 이름 상자에 작업 이름을 입력합니다.
4. 필요에 따라 다음 작업 유형 중 하나를 선택합니다.
  - Amazon S3로 가져오기 - 빈 Snowball Edge 디바이스를 AWS 배송하도록 하려면 이 옵션을 선택합니다. 디바이스를 로컬 네트워크에 연결하고 Snowball Edge 클라이언트를 실행합니다. NFS 공유 또는 S3 어댑터를 사용하여 디바이스에 데이터를 복사 AWS하고 다시 배송하면 데이터가 업로드됩니다 AWS.
  - Amazon S3에서 내보내기 - Amazon S3 버킷에서 디바이스로 데이터를 내보내려면 이 옵션을 선택합니다. AWS 는 데이터를 디바이스에 로드하여 사용자에게 전송합니다. 디바이스를 로컬 네트워크에 연결하고 Snowball Edge 클라이언트를 실행합니다. 디바이스에서 서버로 데이터를 복사합니다. 완료되면 디바이스를 배송하고 디바이스에서 AWS데이터가 지워집니다.
  - 로컬 컴퓨팅 및 스토리지 전용 - 데이터를 전송하지 않고 디바이스에서 컴퓨팅 및 스토리지 워크로드를 수행할 수 있습니다.

### Choose a job type

**Import into Amazon S3** [Info](#)

AWS will ship an empty device to you for storage and compute workloads. You'll transfer your data onto it, and ship it back. After AWS gets it, your data will be moved.

**Export from Amazon S3** [Info](#)

Choose what data you want to export from your S3 buckets for storage and compute workloads. AWS will load that data onto a device and ship it to you. When you're done ship the device back for erasing.

**Local compute and storage only** [Info](#)

Perform local compute and storage workloads without transferring data. You can order multiple devices in a cluster for increased durability and storage capacity. Includes rugged and rack-mountable devices.

5. 다음을 선택하여 계속 진행합니다.

## 컴퓨팅 및 스토리지 옵션 선택

Snowball Edge 디바이스의 하드웨어 사양, 여기에 포함할 Amazon EC2-compatible 인스턴스, 데이터 저장 방법 및 요금을 선택합니다.

## 디바이스의 컴퓨팅 및 스토리지 옵션 선택

1. Snow 디바이스 섹션에서 주문할 Snowball Edge 디바이스를 선택합니다.

### Note

주문 AWS 리전 하는와 선택한 작업 유형에 따라 일부 Snowball Edge를 사용하지 못할 수 있습니다.

2. 요금 옵션 선택 섹션의 요금 옵션 선택 메뉴에서 이 작업에 적용할 요금 유형을 선택합니다. 1년 또는 3년 약정 선결제 요금을 선택한 경우 자동 갱신에서 현재 기간이 종료될 때 요금을 자동으로 갱신하려면 켜짐을 선택하고 현재 기간이 종료될 때 요금이 자동으로 갱신되지 않도록 하려면 꺼짐을 선택합니다. Snowball Edge 디바이스의 장기 요금 옵션에 대한 자세한 내용은 이 설명서의 [Snowball Edge 디바이스의 장기 요금](#)을 참조하세요. 의 디바이스 요금은 [AWS Snowball Edge 요금](#)을 AWS 리전참조하세요.
3. 스토리지 유형 선택 섹션에서 필요에 따라 선택하세요.
  - S3 어댑터: S3 어댑터를 사용하여 Amazon S3 REST API 작업을 사용하여 프로그래밍 방식으로 Snowball Edge와 데이터를 주고받을 수 있습니다.
  - Amazon S3 호환 스토리지: Amazon S3 호환 스토리지를 사용하여 S3와 호환되고 내구성이 뛰어나며 확장 가능한 객체 스토리지를 단일 Snowball Edge 디바이스 또는 다중 디바이스 클러스터에 배포할 수 있습니다.
  - NFS 기반 데이터 전송: NFS(Network File System) 기반 데이터 전송을 사용하여 컴퓨터에서 Snowball Edge의 Amazon S3 버킷으로 파일을 끌어서 놓습니다.

### Warning

NFS 기반 데이터 전송은 S3 어댑터를 지원하지 않습니다. NFS 기반 데이터 전송을 진행하는 경우 NFS 공유를 탑재하여 객체를 전송해야 합니다. 를 사용하여 객체 AWS CLI 를 전송하면 실패합니다.

자세한 내용은 AWS Snowball Edge 개발자 안내서의 [오프라인 데이터 전송에 NFS 사용을](#) 참조하세요.

**Note**

사용할 수 있는 스토리지 유형 옵션은 선택한 작업 유형과 Snow 디바이스에 따라 다릅니다.

4. 스토리지 유형으로 S3 어댑터를 선택하거나 블록 스토리지를 지원하는 디바이스를 선택한 경우에는 다음을 수행하여 디바이스에 포함할 S3 버킷을 하나 이상 선택합니다.

- S3 버킷 선택 섹션에서 다음 중 하나 이상을 수행하여 하나 이상의 S3 버킷을 선택합니다.
  1. S3 버킷 이름 목록에서 사용할 S3 버킷을 선택합니다.
  2. 항목 검색 필드에 버킷 이름 전체 또는 일부를 입력하여 항목의 사용 가능한 버킷 목록을 필터링한 다음 버킷을 선택합니다.
  3. 새 S3 버킷을 만들려면 새 S3 버킷 생성을 선택합니다. 새 버킷 이름이 버킷 이름 목록에 표시됩니다. 이를 선택합니다.

하나 이상의 S3 버킷을 포함할 수 있습니다. 이러한 버킷은 디바이스에 로컬 S3 버킷으로 표시됩니다.

**Select your S3 buckets** [Info](#)

The S3 buckets you select will appear as directories on your device. Data stored in these buckets on the device will not be transferred to S3 on return.

[Create a new S3 bucket](#)

🔍

<input type="checkbox"/>	S3 bucket name	Date created
<input type="checkbox"/>	my-gobally-unique-bucket-name	3/15/2023, 5:20:20 PM EDT
<input type="checkbox"/>	do-not-delete-gatedgarden-audit-669419309129	3/11/2023, 5:13:13 PM EST

5. Amazon S3 호환 스토리지를 스토리지 유형으로 선택한 경우, S3 스토리지 용량 섹션에서 다음을 수행합니다.
- a. 단일 디바이스 또는 디바이스 클러스터에서 Snowball Edge의 Amazon S3 호환 스토리지를 사용하려면 선택합니다. 이 설명서의 [AWS Snowball Edge 클러스터 사용을 참조하세요](#).
  - b. Snowball Edge의 Amazon S3 호환 스토리지에 사용할 디바이스 스토리지 양을 선택합니다.

**Note**

Snowball Edge에서 Amazon S3 호환 스토리지를 사용하는 경우 디바이스를 받은 후 Amazon S3 버킷을 관리하고 생성할 수 있으므로 주문하는 동안 버킷을 선택할 필요가 없습니다. 이 설명서의 [Snowball Edge의 Amazon S3 호환 스토리지](#)를 참조하세요.

**S3 storage capacity**

Select device type

Single device

Cluster

Select storage amount

2.5 TB Single device ▼

Single device


Block storage: 41 TB

6. 스토리지 유형으로 NFS 기반 데이터 전송을 선택한 경우, S3 버킷 선택 섹션에서 다음 중 하나 이상을 수행하여 하나 이상의 S3 버킷을 선택합니다.
  - a. S3 버킷 이름 목록에서 사용할 S3 버킷을 선택합니다.
  - b. 항목 검색 필드에 버킷 이름 전체 또는 일부를 입력하여 항목의 사용 가능한 버킷 목록을 필터링한 다음 버킷을 선택합니다.
  - c. 새 S3 버킷을 만들려면 새 S3 버킷 생성을 선택합니다. 새 버킷 이름이 버킷 이름 목록에 표시됩니다. 이를 선택합니다.
  - d. NFS 데이터 전송에 사용할 S3 버킷을 선택한 후 AMI용 블록 스토리지로 사용할 S3 버킷도 선택합니다. [S3](#) 버킷 선택 단계를 참조하세요.

하나 이상의 S3 버킷을 포함할 수 있습니다. 이러한 버킷은 디바이스에 로컬 S3 버킷으로 표시됩니다.

### Choose your NFS storage

These S3 buckets will appear on directories on your device. You can transfer data onto these buckets using NFS.

 Only data stored in these directories will be ingested to your S3 buckets in the cloud.

The NFS storage limit is 80 TB

[Create a new S3 bucket](#)

<input type="checkbox"/>	S3 bucket name	Date created
<input type="checkbox"/>	this-unique-bucket-name	6/14/2023, 12:20:08 PM EDT

7. EC2 호환 인스턴스를 사용한 컴퓨팅 - 선택 사항 섹션에서 디바이스에 포함할 계정의 Amazon EC2 호환 AMI를 선택합니다. 검색 필드에 버킷 이름 전체 또는 일부를 입력하여 항목의 사용 가능한 AMI 목록을 필터링한 다음 AMI를 선택합니다.

보안 셸(SSH)을 위한 AMI 구성에 대한 자세한 내용은 [Snowball Edge 및 SSH를 위한 AMI 구성을](#) 참조하세요.

자세한 내용은 이 설명서의 [디바이스 주문 시 AMI 추가](#)를 참조하십시오.

이 기능은 추가 요금이 부과됩니다. 자세한 내용은 [AWS Snowball Edge 요금](#)을 참조하세요.

8. 다음 버튼을 선택합니다.

## 기능 및 옵션 선택

Amazon EKS Anywhere for AWS Snow, AWS IoT Greengrass 인스턴스 및 원격 디바이스 관리 기능을 포함하여 Snowball Edge 디바이스 작업에 포함할 기능 및 옵션을 선택합니다.

### 기능 및 옵션 선택

1. Amazon EKS Anywhere on AWS Snow 섹션에서 AWS Snow에 Amazon EKS Anywhere를 포함하려면 Snow에 Amazon EKS Anywhere 포함을 선택한 후 다음을 수행합니다.

#### Note

Amazon EKS Anywhere에서 지원하는 사용 가능한 최신 Kubernetes 버전으로 Kubernetes 클러스터를 생성하는 것이 좋습니다. 자세한 내용은 [Amazon ECS Anywhere](#)

[버전 관리](#)를 참조하세요. 애플리케이션에 특정 버전의 Kubernetes가 필요한 경우 Amazon EKS에서 표준 또는 확장 지원으로 제공하는 모든 버전의 Kubernetes를 사용합니다. 배포 수명 주기를 계획할 때 Kubernetes 버전의 릴리스 및 지원 날짜를 고려합니다. 이렇게 하면 사용하려는 Kubernetes 버전에 대한 지원이 잠재적으로 손실되는 것을 방지할 수 있습니다. 자세한 내용은 [Amazon EKS Kubernetes 출시 일정](#)을 참조하세요.

- a. 자체 AMI 구축 섹션에서 Amazon EKS Anywhere용으로 구축한 AMI를 선택합니다. [Amazon EKS Anywhere on Snow용 AWS Snowball Edge 디바이스 주문 전에 완료해야 할 작업을\(를\)](#) 참조하세요.
  - b. 여러 Snowball Edge 디바이스에서 Amazon EKS Anywhere 클러스터를 운영하려면 고가용성 섹션에서 주문에 포함할 디바이스 수를 선택합니다.
2. AWS IoT Greengrass Snow의 섹션에서 IoT 워크로드에 대해 검증된 AMI를 포함하려면 Snow 디바이스에 AWS IoT Greengrass 검증된 AMI 설치를 선택합니다.
  3. AWS OpsHub 또는 Snowball Edge Client에서 Snowball Edge 디바이스의 원격 관리를 활성화하려면 AWS OpsHub 또는 Snowball Edge Client를 사용하여 원격으로 Snow 디바이스 관리를 선택합니다.
  4. 다음 버튼을 선택합니다.

## 보안, 배송 및 알림 기본 설정 선택

### 주제

- [Snowball Edge에 대한 보안 기본 설정 선택](#)
- [Snowball Edge를 수신하고 반환하기 위한 배송 기본 설정을 선택합니다.](#)
- [Snowball Edge 작업에 대한 알림 기본 설정 선택](#)

### Snowball Edge에 대한 보안 기본 설정 선택

보안을 설정하면 AWS Snowball Edge 작업에 대한 권한 및 암호화 설정이 추가되어 전송 중에 데이터를 보호할 수 있습니다.

#### 작업 보안 설정

1. 암호화 섹션에서 사용하려는 KMS 키를 선택합니다.

- 기본 AWS Key Management Service (AWS KMS) 키를 사용하려면 AWS/importexport(기본 값)를 선택합니다. 이 키는 다른 키가 정의되지 않은 경우 가져오기 및 내보내기 작업을 보호하는 기본 키입니다.
  - 자체 AWS KMS 키를 제공하려면 키 ARN 입력을 선택하고 키 ARN 상자에 Amazon 리소스 이름(ARN)을 입력한 다음 이 KMS 키 사용을 선택합니다. 키 ARN이 목록에 추가됩니다.
2. 서비스 액세스 유형 선택 섹션에서 다음 중 하나를 수행합니다.
- Snow 콘솔을 선택하면 서비스 연결 역할을 생성하고 사용하여 사용자를 대신하여 AWS 리소스에 액세스합니다.는 AWS Snowball Edge에 사용자를 대신하여 Amazon S3 및 Amazon Simple Notification Service(Amazon SNS)를 사용할 수 있는 권한을 부여합니다. 역할은 Snow 서비스에 AWS 보안 토큰 서비스(AWS STS) AssumeRole 신뢰를 부여합니다.
  - 사용할 기존 서비스 역할 추가를 선택하여 원하는 역할 ARN을 지정하거나 기본 역할을 사용할 수 있습니다.
3. 다음을 선택합니다.

Snowball Edge를 수신하고 반환하기 위한 배송 기본 설정을 선택합니다.

Snowball Edge 디바이스를 수령하고 반환하려면 디바이스를 앞뒤로 배송해야 하므로 정확한 배송 정보를 제공하는 것이 중요합니다.

#### 배송 세부 정보 제공

1. 배송 주소 섹션에서 기존 주소를 선택하거나 새 주소를 추가합니다.
  - 최근 주소 사용을 선택하면 파일에 있는 주소가 표시됩니다. 목록에서 원하는 주소를 신중하게 선택합니다.
  - 새 주소 추가를 선택한 경우 요청된 주소 정보를 입력합니다. 는 새 배송 정보를 AWS Snow 패밀리 관리 콘솔 저장합니다.

#### Note

입력한 해당 주소의 국가는 디바이스의 목적지 국가와 일치해야 하며 해당 국가에서 유효해야 합니다.



2. 배송 속도 섹션에서 작업의 배송 속도를 선택합니다. 이 배송 기간은 작업을 생성한 날로부터 디바이스를 수신하는 데 걸리는 시간을 나타내지 않습니다. 대신 AWS 와 배송 주소 간에 디바이스가 전송 중인 시간을 나타냅니다.

디바이스가 배송되기 전에 작업에 맞게 디바이스를 프로비저닝하고 준비하는 데 최대 4주가 걸릴 수 있습니다. 원활한 전환을 위해 이 타임라인을 프로젝트 계획에 반영해야 합니다.

선택할 수 있는 배송 속도는 다음과 같습니다.

- 1일 배송(영업일 기준 1일)
- 2일 배송(영업일 기준 2일)

## Snowball Edge 작업에 대한 알림 기본 설정 선택

알림은 AWS Snowball Edge 작업의 최신 상태를 업데이트합니다. 작업 상태가 변경되면 Amazon Simple Notification Service(SNS)에서 이메일을 받습니다.

### 알림 설정

- 알림 설정 섹션에서 다음 중 하나를 수행합니다.
  - 기존 SNS 주제를 사용하려면 기존 SNS 주제 사용을 선택하고 목록에서 Amazon 리소스 이름(ARN) 주제를 선택합니다.
  - 새 SNS 주제를 생성하려면 새 SNS 주제 생성을 선택합니다. 주제의 이름을 입력하고 이메일 주소를 제공합니다.

#### Note

미국 서부(캘리포니아 북부) 및 미국 서부(오리건) 리전에서 생성된 Snow 디바이스 주문 작업은 미국 동부(버지니아 북부) 리전을 통해 라우팅됩니다. 따라서 Amazon SNS와 같은 서비스 직접 호출은 미국 동부(버지니아 북부)를 통해서도 이루어집니다. 최상의 경험을 위해 미국 동부(버지니아 북부) 리전에서 새 SNS 주제를 생성하는 것이 좋습니다.

알림은 작업의 다음 상태 중 하나와 관련이 있습니다.

- 작업 생성됨
- 디바이스 준비 중

- 배송 준비 중
- 배송 중
- 사용자에게 배달됨
- 로 전송 중 AWS
- 분류 시설에 있음
- 의 AWS
- 가져오기
- 완료됨
- 취소됨

작업 상태 변경 알림 및 암호화된 SNS 주제에 대한 자세한 내용은 이 설명서의 [Snowball Edge에 대한 알림을 참조하세요](#).

다음을 선택합니다.

## 작업 요약 검토 및 작업 생성

AWS Snowball Edge 작업에 필요한 모든 정보를 제공한 후 작업을 검토하고 생성합니다. 작업을 생성한 후 AWS 는 Snowball Edge를 배송 준비하기 시작합니다.

작업에는 특정 국가의 수출 통제법이 적용되며 수출 허가가 필요할 수 있습니다. 미국 수출 및 재수출 법도 적용됩니다. 해당 국가 및 미국의 법률 및 규정을 위반하는 행위는 금지됩니다.

1. 작업을 생성하기 전에 작업 요약 페이지에서 모든 섹션을 검토합니다. 변경하려면 해당 섹션에서 편집을 선택하고 관련 정보를 편집합니다.
2. 검토 및 편집이 끝나면 작업 생성을 선택합니다.

### Note

Snowball Edge 디바이스를 주문하는 작업을 생성한 후 작업 생성 상태에서 요금을 부과하지 않고 취소할 수 있습니다. 자세한 내용은 [AWS Snow 패밀리 관리 콘솔을 통한 작업 취소](#)를 참조하세요.

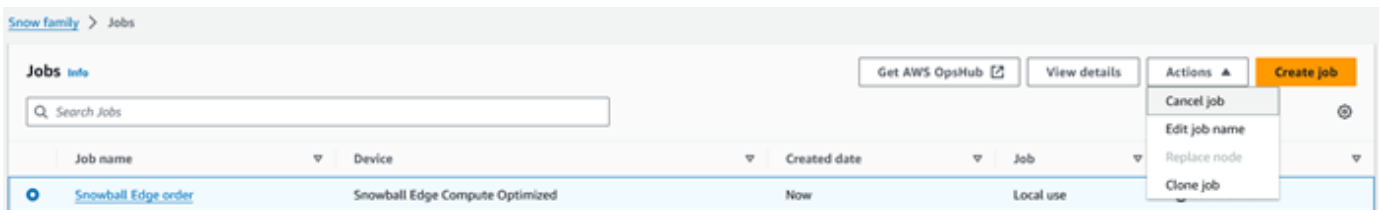
작업이 생성된 후 작업 상태 섹션에서 작업 상태를 확인할 수 있습니다. 작업 상태에 대한 자세한 내용은 [작업 상태](#) 섹션을 참조하십시오.

## Snowball Edge 주문 작업 취소

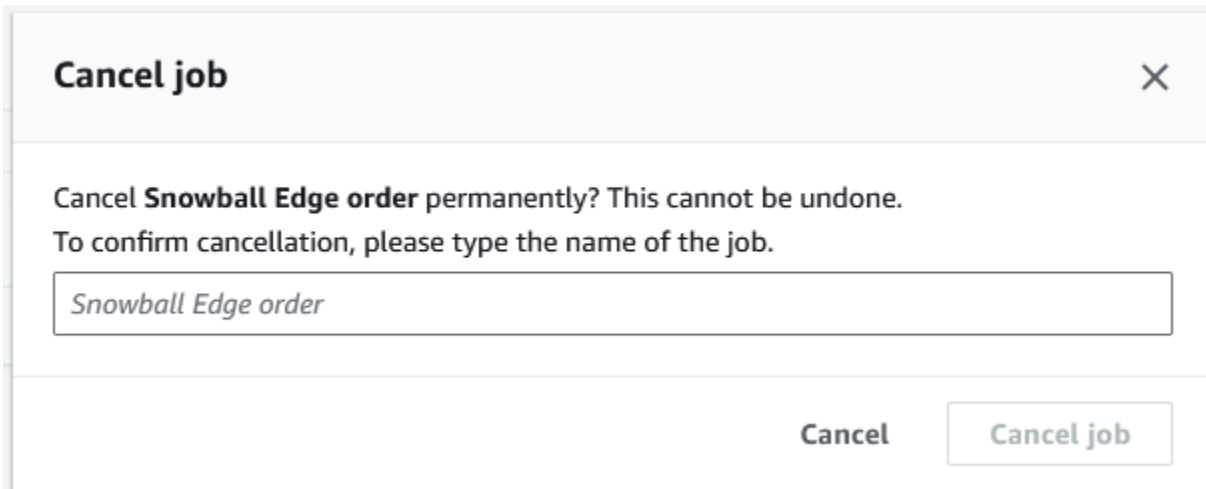
Snowball Edge 디바이스 주문 작업을 생성한 후를 통해 작업을 취소할 수 있습니다 AWS Snow 패밀리 관리 콘솔. 작업을 취소하면 주문한 디바이스가 수신되지 않습니다. 작업 생성된 작업 상태의 작업만 취소할 수 있습니다. 작업이 이 상태를 지나 진행된 후에는 작업을 취소할 수 없습니다. 자세한 내용은 [작업 상태](#)를 참조하세요.

작업 생성 상태인 동안 작업을 취소하면 Snowball Edge 디바이스에 대한 요금이 부과되지 않습니다. 디바이스가 준비되고 배송된 후에만 결제가 시작됩니다.

1. [AWS Snow 패밀리 관리 콘솔](#)에 로그인합니다.
2. 취소할 작업을 선택합니다.
3. 작업을 선택합니다. 표시되는 메뉴에서 작업 취소를 선택합니다.



4. 작업 취소 창이 나타납니다. 작업 취소를 확인하려면 **job name**을 입력하고 작업 취소를 선택합니다. 작업 목록에서 취소됨이 상태 옆에 나타납니다.



## 에서 Snowball Edge 주문 작업 복제 AWS Snow 패밀리 관리 콘솔

가져오기 작업 또는 로컬 컴퓨팅 및 스토리지 작업을 처음 생성할 때 둘 이상의 AWS Snowball Edge 디바이스가 필요함을 발견할 수 있습니다. 가져오기 작업과 로컬 컴퓨팅 및 스토리지 작업은 디바이스

한 대와 연결되므로 디바이스가 여러 대 필요하다는 것은 작업을 두 개 이상 생성해야 함을 뜻합니다. 추가 작업을 생성할 때 콘솔에서 작업 생성 마법사를 다시 실행하거나 기존 작업을 복제할 수 있습니다.

### Note

작업 복제는 콘솔에서 추가 작업을 더 쉽게 생성할 수 있도록 해주는 바로 가기입니다. 작업 관리 API를 사용하여 작업을 생성할 경우 작업 생성 명령을 다시 실행할 수 있습니다.

작업 복제는 자동으로 수정되는 이름을 제외하고 작업을 정확하게 다시 생성하는 것을 의미합니다. 복제는 간단한 프로세스입니다.

콘솔에서 작업을 복제하려면

1. 의 테이블에서 작업을 AWS Snow 패밀리 관리 콘솔선택합니다.
2. [Actions]에서 [Clone job]을 선택합니다.

[Create job] 마법사가 마지막 페이지인 [Step 6: Review]로 열립니다.

3. 정보를 검토하고 해당되는 [Edit] 버튼을 선택하여 원하는 사항을 변경합니다.
4. 복제한 작업을 생성하려면 [Create job]을 선택합니다.

복제된 작업의 이름은 [**Job Name**-clone-*number*] 형식으로 지정됩니다. 작업 이름에 자동으로 추가되는 이 숫자는 이 작업을 처음 복제한 후 복제한 횟수를 나타냅니다. 예를 들어 [AprilFinanceReports-clone]은 [AprilFinanceReports] 작업의 첫 번째 복제된 작업을 나타내고, [DataCenterMigration-clone-42]는 [DataCenterMigration] 작업의 42번째 복제를 나타냅니다.

## Snowball Edge 받기

AWS Snowball Edge 디바이스를 받으면 상자에 들어 있지 않을 수 있습니다. 디바이스는 그 자체가 물리적으로 견고한 배송 컨테이너입니다. 디바이스가 처음 도착하면 손상되거나 명백히 변경된 부분이 있는지 검사합니다. 디바이스에서 의심이 가는 부분이 있으면 내부 네트워크에 연결하지 마세요. 그 대신에 [AWS Support](#)에 연락하여 문제점을 알리면 새 디바이스가 배송됩니다.

**⚠ Important**

AWS Snowball Edge 디바이스는의 속성입니다 AWS. AWS Snowball Edge 디바이스를 조작하는 것은 AWS 허용 가능한 사용 정책을 위반하는 것입니다. 액세스 정책에 대한 자세한 내용은 [AWS 이용목적제한방침](#)을 참조하세요.

디바이스의 외관은 다음과 같습니다.



디바이스를 내부 네트워크에 연결할 준비가 되었다면 다음 섹션을 참조하세요.

다음: [로컬 네트워크에 Snowball Edge 연결](#)

## 로컬 네트워크에 Snowball Edge 연결

다음 절차에 따라 AWS Snowball Edge 디바이스를 로컬 네트워크에 연결합니다. 디바이스를 인터넷에 연결할 필요는 없습니다. 디바이스에는 세 개의 도어(전면, 후면, 상단)가 있습니다.

디바이스를 네트워크에 연결하려면

1. 전면 및 후면 도어를 디바이스 도어 슬롯 안으로 밀어 엽니다. 그러면 디바이스 전면에 내장된 LCD 디스플레이의 터치 스크린과 후면의 전원 및 네트워크 포트에 접근할 수 있습니다.

### Note

Snowball Edge 디바이스를 사용하는 동안에는 전면 및 후면 도어를 닫지 마세요. 도어가 열려 있으면 공기가 디바이스를 식힐 수 있습니다. 디바이스를 사용하는 동안 도어를 닫으면 과열을 방지하기 위해 디바이스가 종료될 수 있습니다.

2. 상단 도어를 열고 케이블 누크에서 제공된 전원 케이블을 제거하고 디바이스를 전원에 연결합니다.
3. RJ45, SFP+ 또는 QSFP+ 네트워크 케이블 중 하나를 선택하고 디바이스를 네트워크에 연결합니다. 네트워크 포트는 디바이스의 후면에 있습니다.
4. LCD 디스플레이 위의 전원 버튼을 눌러 AWS Snowball Edge 디바이스의 전원을 켭니다.
5. 디바이스가 준비되면 디바이스가 시작 준비를 하는 동안 LCD 디스플레이에 짧은 비디오가 재생됩니다. 약 10분 후 디바이스는 잠금 해제할 준비가 됩니다.
6. (선택 사항) [CONNECTION]을 선택하여 LCD 디스플레이를 통해 기본 네트워크 설정을 변경합니다.

다음 절차를 사용하여 IP 주소를 제공한 다른 정적 주소로 변경할 수 있습니다.

부팅 문제를 해결하려면 [Snowball Edge의 부팅 문제 해결](#) 섹션을 참조하세요.

AWS Snowball Edge 디바이스의 IP 주소를 변경하려면

1. LCD 디스플레이에서 연결을 터치합니다.

AWS Snowball Edge 디바이스에 대한 현재 네트워크 설정이 표시된 화면이 나타납니다. 드롭다운 상자 아래의 IP 주소는 AWS Snowball Edge 디바이스가 요청한 DHCP 주소를 반영하도록 자동으로 업데이트됩니다.

2. (선택 사항) IP 주소를 정적 IP 주소로 변경합니다. 그대로 유지할 수도 있습니다.

현재 디바이스가 네트워크에 연결되어 있습니다.

#### Important

데이터 손상을 방지하려면 사용 중에 AWS Snowball Edge 디바이스의 연결을 끊거나 연결 설정을 변경하지 마십시오.

다음: [Snowball Edge에 액세스하기 위한 자격 증명 가져오기](#)

## Snowball Edge에 액세스하기 위한 자격 증명 가져오기

각 작업에는 Snowball Edge에 대한 액세스를 인증하기 위해 AWS Snow 패밀리 관리 콘솔 또는 작업 관리 API에서 가져와야 하는 자격 증명 세트가 있습니다. 이러한 자격 증명은 암호화된 매니페스트 파일 및 연결된 잠금 해제 코드입니다. 매니페스트 파일에는 작업 및 작업에 연결된 권한에 대한 중요한 정보가 포함되어 있습니다.

#### Note

보안 인증 정보는 디바이스가 전송된 후에 받게 됩니다. AWS Snow 패밀리 관리 콘솔에서 작업 상태를 확인할 수 있습니다. 자세한 내용은 [Snowball Edge 작업의 상태](#) 단원을 참조하십시오.

콘솔을 사용하여 보안 인증 정보 받기

1. 에 로그인 AWS Management Console 하고를 엽니다 [AWS Snow 패밀리 관리 콘솔](#).
2. 콘솔에서 작업 매니페스트를 다운로드할 특정 작업에 대한 테이블을 검색한 다음 해당 작업을 선택합니다.
3. 작업 상태 창을 확장하고 작업 세부 정보 보기를 선택합니다.
4. 세부 정보 창이 표시되면 보안 인증 정보를 확장하고 다음을 수행합니다.

- 디바이스를 잠금 해제하려면 29자를 모두 제공해야 하기 때문에 잠금 해제 코드(하이픈 포함)를 기록해 둡니다.
- 대화 상자에서 매니페스트 다운로드를 선택하고 지침에 따라 컴퓨터에 작업 매니페스트 파일을 다운로드합니다. 매니페스트 파일의 이름에는 [Job ID]가 포함됩니다.

### Note

잠금 해제 코드의 복사본을 해당 작업에 대한 매니페스트와 동일한 컴퓨터 내 위치에 저장하지 않는 것이 좋습니다. 자세한 내용은 [Snowball Edge 디바이스 사용 모범 사례](#) 단원을 참조하십시오.

이제 자격 증명이 있으므로 다음 단계는 AWS Snowball Edge 디바이스 잠금을 해제하는 데 사용되는 Snowball Edge 클라이언트를 다운로드하는 것입니다.

다음: [Snowball Edge 클라이언트 다운로드 및 설치](#)

## Snowball Edge 잠금 해제

이 섹션에서는 Snowball Edge 클라이언트를 사용하여 Snowball Edge 디바이스를 잠금 해제하는 방법을 설명합니다. Snowball Edge용 그래픽 사용자 인터페이스(GUI) 도구 AWS OpsHub인을 사용하여 디바이스를 잠금 해제하려면 [디바이스 잠금 해제](#).

Snowball Edge 디바이스를 사용하여 데이터를 전송하거나 엣지 컴퓨팅 작업을 수행하기 전에 디바이스의 잠금을 해제해야 합니다. 디바이스를 잠금 해제할 때 29자리 잠금 해제 코드와 매니페스트 파일의 두 가지 형식의 자격 증명을 제공하여 디바이스에 액세스할 수 있는 기능을 인증합니다. 디바이스 잠금을 해제한 후 디바이스를 추가로 구성하고, 디바이스 간에 데이터를 이동하고, Amazon EC2-compatible 인스턴스를 설정하고 사용할 수 있습니다.

디바이스를 잠금 해제하기 전에 디바이스를 전원 및 네트워크에 연결하고 전원을 켜 다음 IP 주소를 할당해야 합니다. [로컬 네트워크에 Snowball Edge 연결](#) 참조하세요. Snowball Edge 디바이스에 대한 다음 정보가 필요합니다.

- Snowball Edge 클라이언트를 다운로드 및 설치합니다. 자세한 내용은 [Snowball Edge 클라이언트 다운로드 및 설치](#) 단원을 참조하십시오.
- AWS Snow 패밀리 관리 콘솔에서 자격 증명을 받습니다. 하나 이상의 독립 실행형 디바이스의 경우 각 Snowball Edge의 잠금 해제 코드 및 매니페스트 파일입니다. Snowball Edge 디바이스 클러스터



의 경우 클러스터의 잠금 해제 코드 하나와 매니페스트 파일 하나입니다. 자격 증명 다운로드에 대한 자세한 내용은 [Snowball Edge에 액세스하기 위한 자격 증명 가져오기](#) 섹션을 참조하세요.

- 각 디바이스의 전원을 켜고 네트워크에 연결합니다. 자세한 내용은 [로컬 네트워크에 Snowball Edge 연결](#) 단원을 참조하십시오.

Snowball Edge 클라이언트를 사용하여 독립형 디바이스를 잠금 해제하려면

1. AWS Snowball Edge 디바이스의 LCD 디스플레이에 있는 연결 탭에서 AWS Snowball Edge 디바이스의 IP 주소를 찾습니다. 이 IP 주소를 기록해 둡니다.
2. `unlock-device` 명령을 사용하여 다음과 같이 Snowball Edge의 IP 주소와 자격 증명을 사용하여 Snowball Edge에 대한 액세스를 인증합니다.

```
snowballEdge unlock-device --endpoint https://ip-address-of-device --manifest-file /Path/to/manifest/file.bin --unlock-code 29-character-unlock-code
```

다음 메시지로 디바이스가 성공적으로 잠금 해제되었음을 나타냅니다.

```
Your Snowball Edge device is unlocking. You may determine the unlock state of your device using the describe-device command. Your Snowball Edge device will be available for use when it is in the UNLOCKED state.
```

명령이 `connection refused`를 반환하는 경우 [Snowball Edge 잠금 해제 문제 해결](#) 섹션을 참조하세요.

Example/ `unlock-device` 명령

이 예에서 디바이스에 대한 IP 주소는 `192.0.2.0`이고, 매니페스트 파일 이름은 `JID2EXAMPLE-0c40-49a7-9f53-916aEXAMPLE81-manifest.bin`이며, 29자로 된 잠금 해제 코드는 `12345-abcde-12345-ABCDE-12345`입니다.

```
snowballEdge unlock-device --endpoint https://192.0.2.0 --manifest-file /Downloads/JID2EXAMPLE-0c40-49a7-9f53-916aEXAMPLE81-manifest.bin / --unlock-code 12345-abcde-12345-ABCDE-12345
```

## Snowball Edge 클라이언트를 사용하여 Snowball Edge 디바이스 클러스터를 잠금 해제하려면

1. 클러스터에 있는 디바이스의 IP 주소는 각 AWS Snowball Edge 디바이스의 LDC 디스플레이의 연결 탭 아래에서 확인합니다. IP 주소를 적어둡니다.
2. `snowballEdge unlock-cluster` 명령을 사용하여 다음과 같이 클러스터에 있는 AWS Snowball Edge 디바이스 중 하나의 IP 주소, 자격 증명 및 클러스터에 있는 모든 디바이스의 IP 주소를 사용하여 디바이스 클러스터에 대한 액세스를 인증합니다.

```
snowballEdge unlock-cluster --endpoint https://ip-address-of-device --manifest-file Path/to/manifest/file.bin --unlock-code 29-character-unlock-code --device-ip-addresses ip-address-of-cluster-device-1 ip-address-of-cluster-device-2 ip-address-of-cluster-device-3
```

다음 메시지로 디바이스의 클러스터가 성공적으로 잠금 해제되었음을 나타냅니다.

```
Your Snowball Edge Cluster is unlocking. You may determine the unlock state of your cluster using the describe-cluster command. Your Snowball Edge Cluster will be available for use when your Snowball Edge devices are in the UNLOCKED state.
```

명령이 `connection refused`를 반환하는 경우 [Snowball Edge 잠금 해제 문제 해결](#) 섹션을 참조하세요.

### Example/ unlock-cluster 명령

이 예제에서는 디바이스 5개로 구성된 클러스터의 경우 클러스터에 있는 디바이스 중 하나의 IP 주소는 `192.0.2.0`이고 매니페스트 파일 이름은 `JID2EXAMPLE-0c40-49a7-9f53-916aEXAMPLE81-manifest.bin`이며 29자 잠금 해제 코드는 `12345-abcde-12345-ABCDE-12345`입니다.

```
snowballEdge unlock-cluster --endpoint https://192.0.2.0 --manifest-file /
Downloads/JID2EXAMPLE-0c40-49a7-9f53-916aEXAMPLE81-manifest.bin /

--unlock-code 12345-abcde-12345-ABCDE-12345 --device-ip-addresses 192.0.2.0
192.0.2.1 192.0.2.2 192.0.2.3 192.0.2.4
```

## Snowball Edge 잠금 해제 문제 해결

unlock-device 명령이 connection refused를 반환할 경우 명령 구문을 잘못 입력했거나 컴퓨터 또는 네트워크 구성으로 인해 명령이 Snow 디바이스에 도달하지 못할 수 있습니다. 상황을 해결하려면 다음 조치를 취하세요.

1. 명령이 올바르게 입력되었는지 확인합니다.
  - a. 디바이스의 LCD 화면을 사용하여 명령에 사용된 IP 주소가 올바른지 확인합니다.
  - b. 파일 이름을 포함하여 명령에 사용된 매니페스트 파일의 경로가 올바른지 확인합니다.
  - c. [AWS Snowball Edge Management Console](#)을 사용하여 명령에 사용된 잠금 해제 코드가 올바른지 확인합니다.
2. 사용 중인 컴퓨터가 Snow 디바이스와 동일한 네트워크 및 서브넷에 있는지 확인합니다.
3. 사용 중인 컴퓨터와 네트워크가 Snow 디바이스에 대한 액세스를 허용하도록 구성되어 있는지 확인합니다. 운영 체제의 ping 명령을 사용하여 컴퓨터가 네트워크를 통해 Snow 디바이스에 도달할 수 있는지 확인합니다. 바이러스 백신 소프트웨어의 구성, 방화벽 구성, 가상 프라이빗 네트워크(VPN) 또는 컴퓨터 및 네트워크의 기타 구성을 확인합니다.

이제 Snowball Edge 사용을 시작할 수 있습니다.

다음: [Snowball Edge에서 로컬 사용자 설정](#)

## Snowball Edge에서 로컬 사용자 설정

다음은 AWS Snowball Edge 디바이스에서 로컬 관리자를 설정하는 단계입니다.

1. 루트 사용자 자격 증명 검색

snowballEdge list-access-keys 및 snowballEdge get-secret-access-key를 사용하여 로컬 자격 증명을 가져옵니다. 자세한 내용은 [Snowball Edge에 대한 자격 증명 가져오기](#) 단원을 참조하십시오.

## 2. **aws configure**를 사용하여 루트 사용자 자격 증명 구성

AWS Access Key ID, AWS Secret Access Key 및 Default region name을 지정합니다. 리전 이름은 snow이어야 합니다. 선택적으로 Default output format을 지정합니다. 구성에 대한 자세한 내용은 AWS Command Line Interface 사용 설명서 [의 구성을 AWS CLI](#) AWS CLI참조 하세요.

## 3. 디바이스에 하나 이상의 로컬 사용자 생성

create-user 명령을 사용하여 디바이스에 사용자를 추가합니다.

```
aws iam create-user --endpoint endpointIPAddress:6078 --region snow --user-name UserName --profile ProfileID
```

비즈니스 요구 사항에 따라 사용자를 추가한 후에는 AWS 루트 보안 인증 정보를 안전한 위치에 저장하고 계정 및 서비스 관리 작업에만 사용할 수 있습니다. IAM 사용자 생성에 대한 자세한 내용은 IAM 사용 설명서의 [AWS 계정에서 IAM 사용자 생성](#) 섹션을 참조하세요.

## 4. 사용자의 액세스 키 생성

### Warning

이 시나리오에서는 프로그래밍 방식 액세스 권한과 장기 보안 인증이 있는 IAM 사용자가 필요하며 이는 보안 위험을 내포합니다. 이 위험을 줄이려면 이러한 사용자에게 작업을 수행하는 데 필요한 권한만 제공하고 더 이상 필요하지 않을 경우 이러한 사용자를 제거하는 것이 좋습니다. 필요한 경우 액세스 키를 업데이트할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 [액세스 키 업데이트](#)를 참조하세요.

create-access-key 명령을 사용하여 사용자의 액세스 키를 생성합니다.

```
aws iam create-access-key --endpoint endpointIPAddress:6078 --region snow --user-name UserName --profile ProfileID
```

액세스 키 정보를 파일에 저장하고 사용자에게 배포합니다.

## 5. 액세스 정책 생성

사용자에 따라 디바이스 기능에 대한 액세스 수준을 다르게 설정할 수 있습니다. 다음 예시에서는 s3-only-policy라는 정책 문서를 생성하여 사용자에게 연결합니다.

JSON

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

```
aws iam create-policy --endpoint endpointIPAddress:6078 --region snow --policy-name
s3-only-policy --policy-document file://s3-only-policy --profile ProfileID
```

## 6. 사용자에게 정책 연결

attach-user-policy를 사용하여 s3-only-policy를 사용자에게 연결합니다.

```
aws iam attach-user-policy --endpoint endpointIPAddress:6078 --region snow
--user-name UserName --policy-arn arn:aws:iam::AccountID:policy/POLICYNAME --
profile ProfileID
```

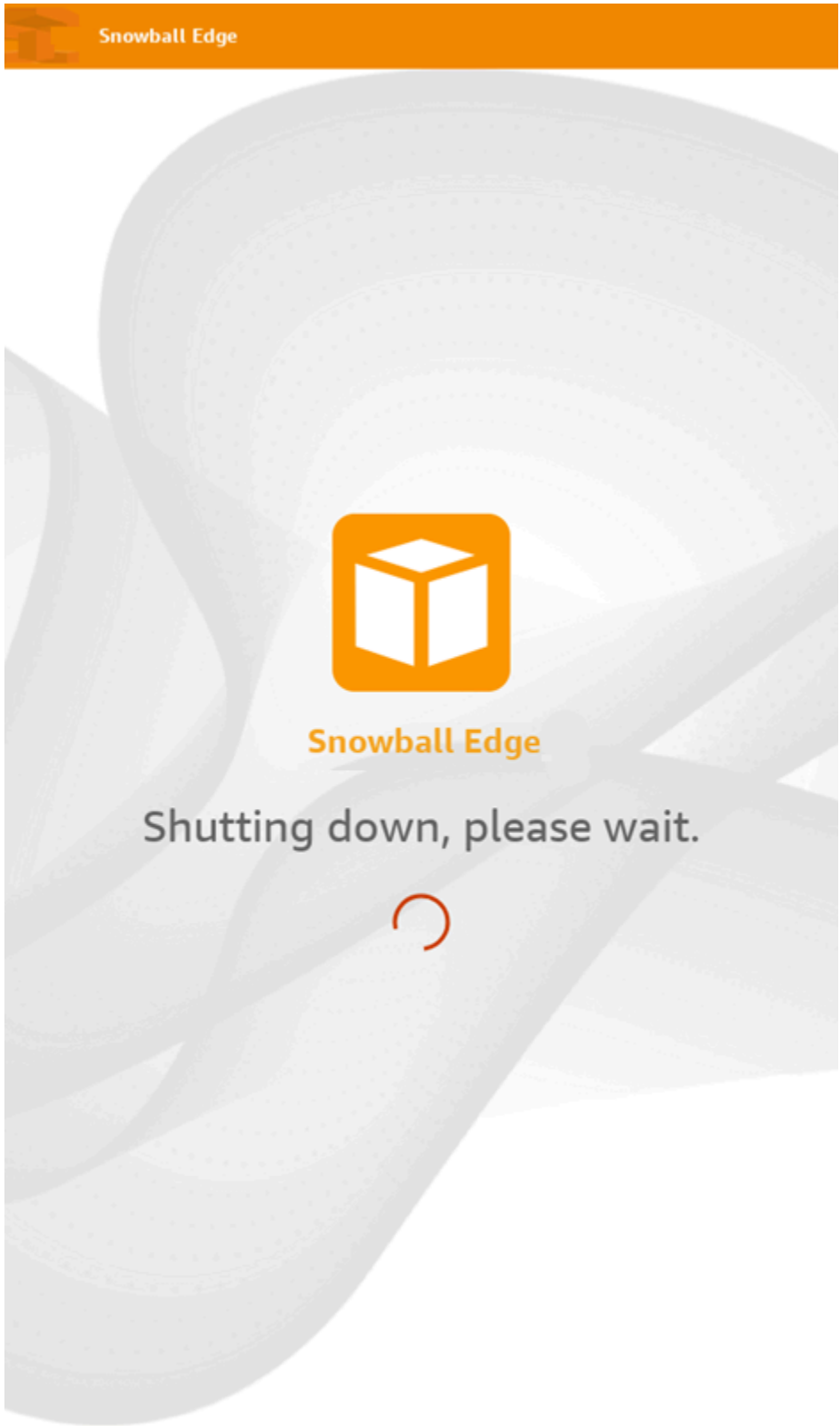
로컬로 IAM 사용에 대한 자세한 내용은 [Snowball Edge에서 로컬로 IAM 사용](#) 섹션을 참조하세요.

## Snowball Edge 디바이스 재부팅

Snowball Edge 디바이스를 재부팅하기 전에 디바이스로의 모든 데이터 전송이 중지되었는지 확인합니다.

전원 버튼을 사용하여 디바이스를 재부팅하려면:

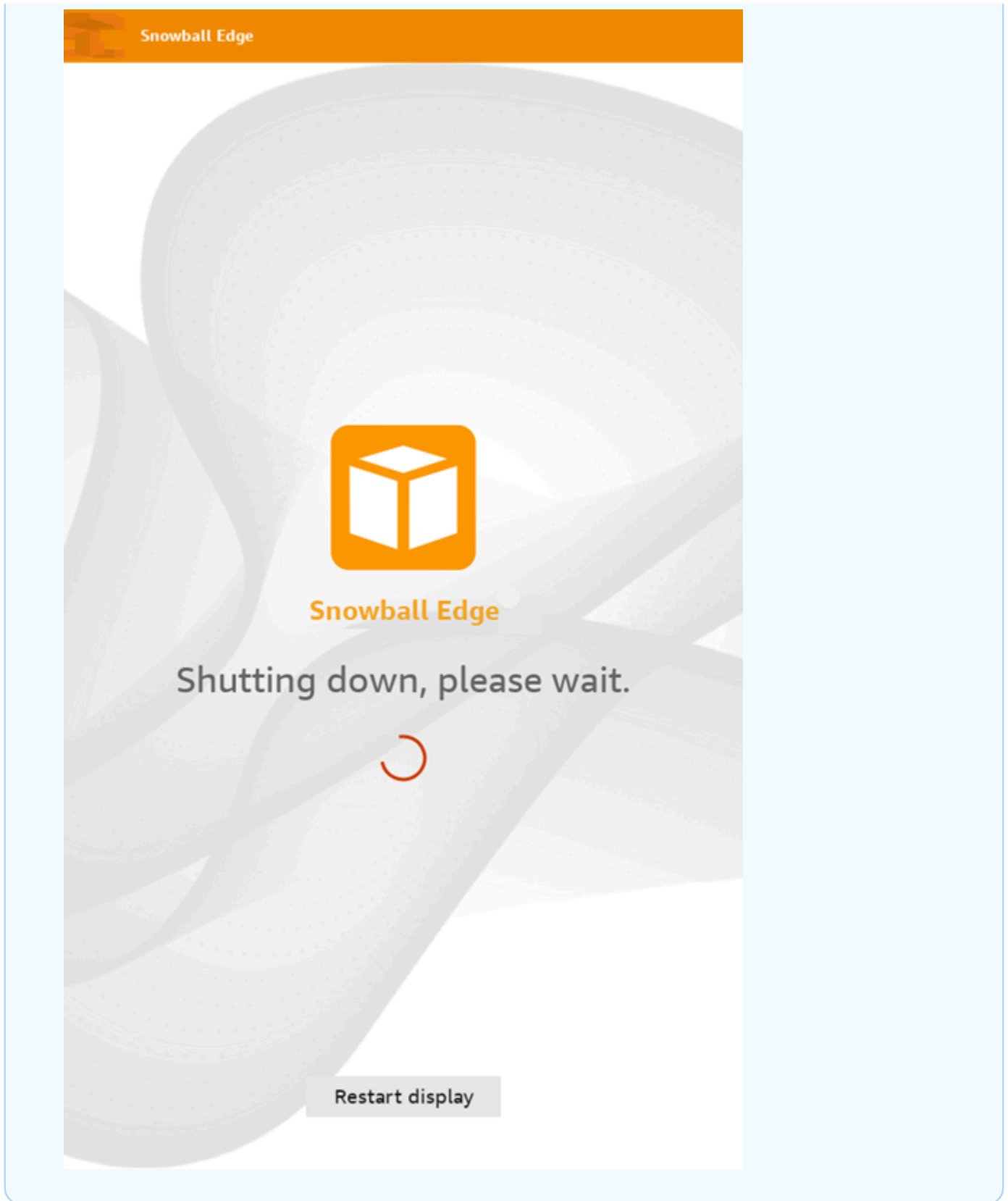
1. 디바이스와의 모든 통신이 종료된 후 LCD 화면 위에 있는 전원 버튼을 눌러서 전원을 끕니다. 디바이스를 종료하는 데 20초 정도 걸립니다. 디바이스가 종료되는 동안 LCD 화면에 디바이스가 종료되었음을 알리는 메시지가 표시됩니다.



**Note**

디바이스가 실제로 종료되지 않았는데도 LCD 화면에 종료 메시지가 표시되는 경우 화면의 디스플레이 재시작 버튼을 눌러 화면을 정상 작동 상태로 되돌립니다.





2. 전원 버튼을 누릅니다. 디바이스가 준비되면 디바이스가 시작 준비를 하는 동안 LCD 디스플레이에 짧은 비디오가 재생됩니다. 약 10분 후 디바이스는 잠금 해제할 준비가 됩니다.

3. 디바이스의 잠금을 해제합니다. [Snowball Edge 잠금 해제](#) 섹션을 참조하세요.

Snowball Edge 클라이언트를 사용하여 디바이스를 재부팅하려면:

1. 디바이스와의 모든 통신이 종료되면 `reboot-device` 명령을 사용하여 디바이스를 재부팅합니다. 디바이스가 준비되면 디바이스가 시작 준비를 하는 동안 LCD 디스플레이에 짧은 비디오가 재생됩니다. 약 10분 후 디바이스는 잠금 해제할 준비가 됩니다.

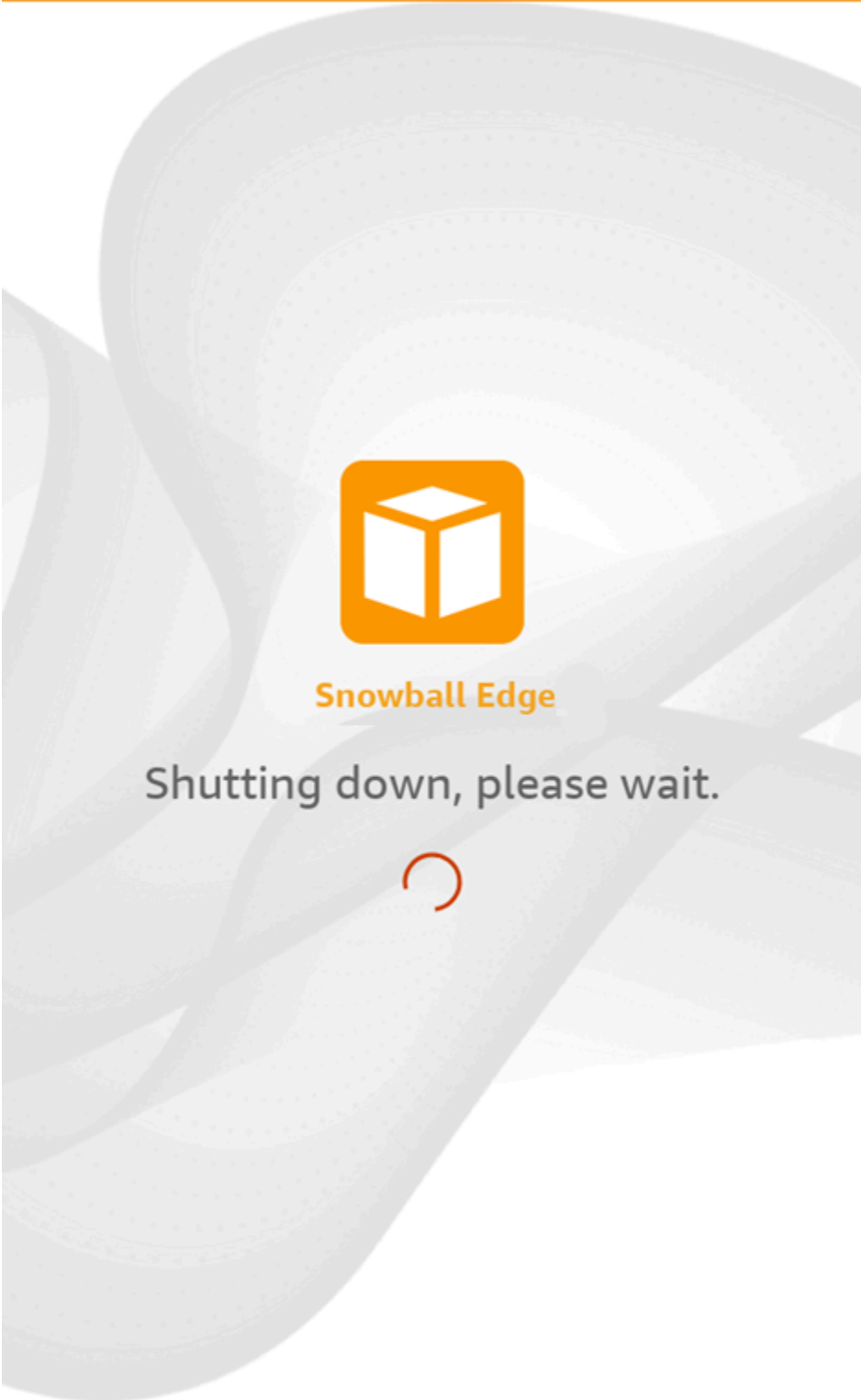
```
snowballEdge reboot-device --profile profile-name
```

2. 디바이스의 잠금을 해제합니다. [Snowball Edge 잠금 해제](#) 섹션을 참조하세요.

## Snowball Edge 전원 끄기

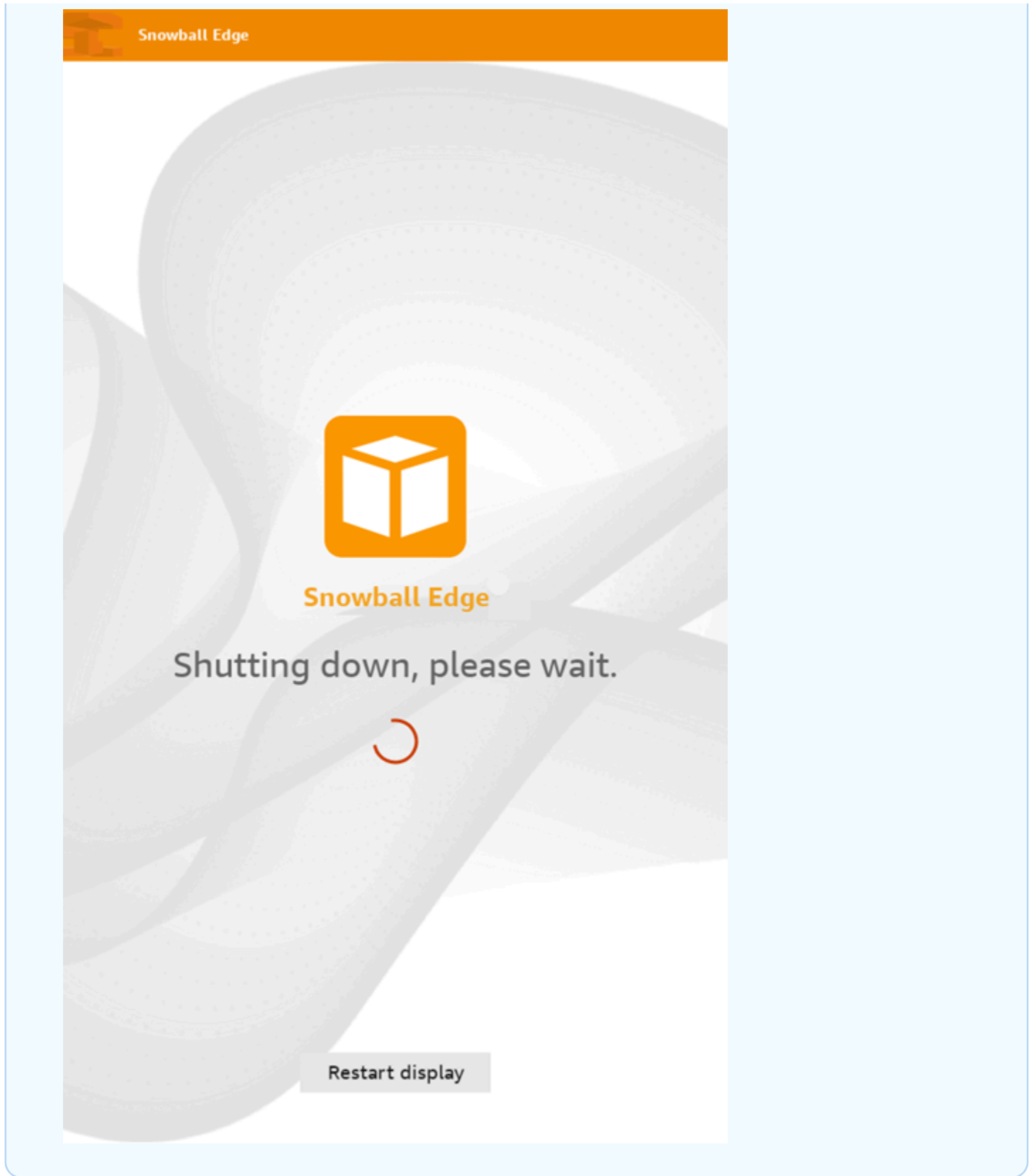
에서 AWS Snowball Edge 디바이스로 데이터 전송을 마쳤으면 디바이스를 반환할 준비를 합니다 AWS. 계속하기 전에 디바이스에 대한 모든 데이터 전송이 중단되었는지 확인합니다. NFS 인터페이스를 사용하여 데이터를 전송하는 경우에는 인터페이스를 비활성화한 후 디바이스 전원을 꺼야 합니다. 자세한 내용은 [NFS 인터페이스 관리](#)를 참조하세요.

디바이스와의 모든 통신이 종료된 후 LCD 화면 위에 있는 전원 버튼을 눌러서 전원을 끕니다. 디바이스를 종료하는 데 20초 정도 걸립니다. 디바이스가 종료되는 동안 LCD 화면에 디바이스가 종료되었음을 알리는 메시지가 표시됩니다.



**Note**

디바이스가 실제로 종료되지 않았는데도 LCD 화면에 종료 메시지가 표시되는 경우 화면의 디스플레이 재시작 버튼을 눌러 화면을 정상 작동 상태로 되돌립니다.



디바이스가 종료되면 E Ink 디스플레이에 배송 정보가 표시됩니다. E Ink 디스플레이에 반품 배송 정보가 표시되지 않으면 지원에 문의하세요.

다음: [Snowball Edge 디바이스 반환](#)

## Snowball Edge 디바이스 반환

Snowball Edge 사용을 완료하고 전원을 끈 후 배송 업체가 해당 Edge를 로 반환합니다 AWS. 운송업체는 디바이스 배송에 대한 추적 번호를 자동으로 제공합니다. 추적 번호는 AWS Snow 패밀리 관리 콘솔에 표시됩니다. 콘솔에서 작업 상태 세부 정보를 확인하여 추적 번호 및 운송업체의 추적 웹사이트에 대한 링크에 액세스할 수 있습니다. 자세한 내용은 [Snowball Edge 디바이스의 반품 배송을 참조하세요](#).

통신 사업자는 디바이스를 AWS 정렬 시설로 전달하고 디바이스는 AWS 데이터 센터로 전달됩니다. 데이터 센터에서 AWS 는 배송 중에 디바이스가 변조되지 않았는지, 디바이스가 정상인지 확인합니다. 디바이스에 Amazon S3로 가져올 데이터가 포함된 경우 AWS 에서 가져오기를 시작합니다. 그렇지 않으면 디바이스의 데이터가 안전하게 삭제됩니다. 가에서 디바이스를 AWS 처리할 때 상태 변경을 추적할 수 있습니다 AWS Snow 패밀리 관리 콘솔. 디바이스 주문 작업을 생성할 때 이 옵션을 선택하면 상태 변경에 대한 Amazon SNS 알림을 받게 됩니다. 자세한 내용은 [가져오기 상태 모니터링](#)을 참조하세요.

최종 상태 값에는 AWS Snowball Edge 디바이스가 수신된 시간 AWS, 데이터 가져오기가 시작된 시간 및 작업이 완료된 시간이 포함됩니다.

### Note

디바이스에 Amazon S3로 가져오려는 데이터가 포함되어 있고 디바이스의 데이터를 가져오지 않으려면 지원에 문의하여 Snow 작업 취소를 요청하십시오. 작업을 취소하면 데이터 전송을 건너뛰고 설정된 프로세스에 따라 디바이스를 안전하게 삭제합니다. 엄격한 관리 연속성 및 운영 절차로 인해 당사는 귀하의 데이터가 들어 있는 디바이스를 당사 시설에 보관할 수 없습니다.

반품 배송을 위해 AWS Snowball Edge 디바이스를 준비하려면

1. 디바이스의 전원을 끕니다. 자세한 내용은 [Snowball Edge 전원 끄기](#) 단원을 참조하십시오.
2. 디바이스에 연결된 모든 네트워크 케이블을 분리합니다.
3. 전원 케이블을 분리합니다. AWS Snowball Edge 디바이스 상단의 케이블 누크에 보관합니다.

4. AWS Snowball Edge 디바이스의 후면, 상단 및 전면에 있는 문을 닫습니다. 딸깍 소리가 나고 느껴질 때까지 도어를 누릅니다.

다음: [Snowball Edge 반품 배송](#)

## Snowball Edge 반품 배송

AWS Snowball Edge 디바이스는 AWS 데이터 센터에서 배송되어 데이터 센터로 전달됩니다. 디바이스의 E Ink 화면에 있는 선불 배송 정보에는 AWS Snowball Edge 디바이스를 반환할 주소가 포함됩니다. 반품 배송 속도는 디바이스를 수령했을 때의 원래 배송 속도와 일치합니다. AWS Snow 패밀리 관리 콘솔을 사용하여 상태 변경을 추적하고 해당 리전의 운송업체를 통해 패키지 진행 상황을 추적할 수 있습니다.

AWS Snowball Edge 디바이스를 반환하는 방법에 대한 자세한 내용은 섹션을 참조하세요 [Snowball Edge 운송업체](#).

### Important

에서 달리 지시하지 않는 한 AWS Snowball Edge 디바이스에 별도의 배송 라벨을 부착하지 AWS하십시오. AWS Snowball Edge 디바이스 E Ink 디스플레이에 표시된 배송 정보를 항상 사용합니다.

## Snowball Edge 운송업체

Snowball Edge 디바이스 주문 작업을 생성할 때 AWS Snowball Edge 디바이스를 배송할 주소를 제공합니다. 리전을 지원하는 운송업체가에서 사용자 AWS 로, 그리고에서 사용자로 디바이스 배송을 처리합니다 AWS. 작업이 배송 준비 중 상태에 도달하면 아웃바운드 배송 정보를 볼 수 있습니다.

배송되는 모든 AWS Snowball Edge 디바이스에 대한 추적 번호가 있습니다. [AWS Snow 패밀리 관리 콘솔](#) 작업 대시보드 또는 작업 관리 API에서 추적 번호와 추적 웹 사이트에 대한 링크를 찾을 수 있습니다.

이러한 통신 사업자는 AWS Snowball Edge 디바이스에 대해 지원됩니다.

- 인도의 운송업체는 경우 Blue Dart입니다.
- 한국, 일본, 호주, 인도네시아의 운송업체는 Kuehne + Nagel입니다.
- 중국과 홍콩의 운송업체는 S.F. Express입니다.

- 다른 모든 리전의 경우 [UPS](#)가 배송 업체입니다.

## 주제

- [EU, 미국, 영국, 남아프리카 및 캐나다의 운송업체는 Snowball Edge UPS입니다.](#)
- [영국에서의 Snowball Edge 픽업](#)
- [브라질에서의 Snowball Edge 픽업](#)
- [호주에서의 Snowball Edge 픽업](#)
- [인도에서의 Snowball Edge 픽업](#)
- [한국에서의 Snowball Edge 픽업](#)
- [홍콩에서의 Snowball Edge 픽업](#)
- [싱가포르, 일본, 인도네시아에서의 Snowball Edge 픽업](#)
- [아랍에미리트 두바이에서의 Snowball Edge 수령 및 반품](#)
- [Snowball Edge의 배송 속도](#)

EU, 미국, 영국, 남아프리카 및 캐나다의 운송업체는 Snowball Edge UPS입니다.

UPS는 주로 EU, 미국, 영국, 남아프리카 및 캐나다에서 디바이스를 픽업할 수 있습니다. 몇 가지 유용한 지침은 다음과 같습니다.

- UPS에서 직접 픽업을 예약하거나 AWS Snowball Edge 디바이스를 배송할 UPS 패키지 인계 시설로 데려갑니다 AWS.
- E Ink 디스플레이의 선불 UPS 배송 라벨에는 AWS Snowball Edge 디바이스의 반송 주소가 포함되어 있습니다.
- AWS Snowball Edge 디바이스는 AWS 정렬 시설로 전달되고 AWS 데이터 센터로 전달됩니다. UPS는 추적 번호를 제공합니다.

### Important

에서 달리 지시하지 않는 한 AWS Snowball Edge 디바이스에 별도의 배송 라벨을 부착하지 AWS하십시오. 항상 디바이스 E Ink 디스플레이에 표시된 배송 라벨을 사용하세요.

UPS는 EU 회원 국가(오스트리아, 벨기에, 불가리아, 크로아티아, 키프로스, 체코, 덴마크, 에스토니아, 핀란드, 프랑스, 독일, 그리스, 헝가리, 이탈리아, 아일랜드, 라트비아, 리투아니아, 룩셈부르크, 몰타,



네덜란드, 폴란드, 포르투갈, 루마니아, 슬로바키아, 슬로베니아, 스페인, 스웨덴)로만 Snowball Edge 디바이스를 배송합니다.

#### Note

영국과 유럽 연합 국가 간의 주문은 이제 국제 주문으로 간주되며 특별한 국제 절차를 통한 승인이 필요합니다. 영국과 EU 간에 디바이스를 배송해야 하는 경우, UPS에 픽업 또는 하역 일정을 잡기 전에 <snowball-shipping@amazon.com>으로 이메일을 보내 상업 송장을 요청하세요.

제품의 Snowball Edge에 대한 UPS 서비스는 한 국가 내에서만 국내 서비스를 제공합니다.

### 영국에서의 Snowball Edge 픽업

영국에서는 UPS의 Snowball Edge 픽업과 관련하여 다음 정보에 유의하도록 합니다.

- UPS에서 직접 픽업을 예약하여 AWS Snowball Edge 디바이스를 픽업하도록 예약하거나 AWS Snowball Edge 디바이스를 배송할 UPS 패키지 인계 시설로 데려갑니다 AWS.
- E Ink 디스플레이의 선불 UPS 배송 라벨에는 AWS Snowball Edge 디바이스를 반환할 올바른 주소가 포함되어 있습니다.
- AWS Snowball Edge 디바이스는 AWS 정렬 시설로 전달되고 AWS 데이터 센터로 전달됩니다. UPS에서는 작업에 대한 추적 번호를 자동으로 보고합니다.

#### Important

에서 별도로 지시하지 않는 한 AWS Snowball Edge 디바이스에 별도의 배송 라벨을 부착하지 AWS하십시오. 항상 디바이스의 E Ink 디스플레이에 표시된 배송 라벨을 사용하세요.

제품의 Snowball Edge에 대한 UPS 서비스는 한 국가 내에서만 국내 서비스를 제공합니다.

#### Note

2021년 1월부터 영국은 더 이상 EU에 속하지 않습니다. 영국과 다른 EU 국가 간의 주문은 국제 주문이며, 일반 배송 절차가 아닌 경우 특별 국제 절차를 통해서만 승인됩니다. 승인을 받아 EU 국가에서 LHR로 또는 영국에서 EU 국가로 디바이스를 반품하는 고객은 먼저

<snowball-shipping@amazon.com>에 반품을 요청해야 합니다. 그러면 UPS와 픽업/드롭 오프를 협의하기 전에 상업 송장을 제공할 수 있습니다.

## 브라질에서의 Snowball Edge 픽업

다음은 브라질에서 Snowball Edge 디바이스를 픽업하기 위한 UPS에 대한 몇 가지 지침입니다.

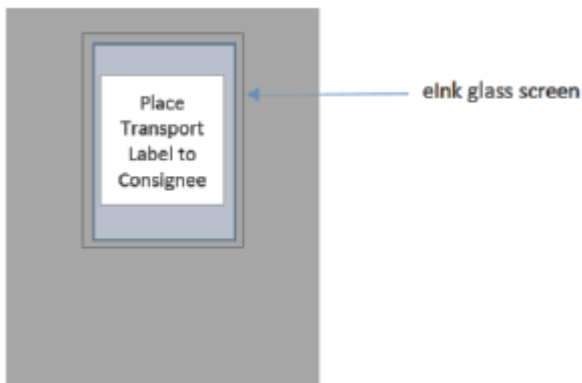
- Snowball Edge 디바이스를 반환할 준비가 되면 0800-770-9035으로 연락하여 UPS 픽업을 예약합니다.
- Snowball Edge Edge는 26개 주와 Distrito Federal을 포함하는 브라질 국내에서 사용할 수 있습니다.
- Cadastro Nacional de Pessoa Juridica(CNPJ) 세금 ID가 있는 경우에는 작업을 생성하기 전에 이 ID를 알고 있어야 합니다.
- Snowball Edge 디바이스를 반환하려면 관련 문서를 발행해야 합니다. Imposto sobre Circulação de Mercadorias e Serviços(ICMS) 등록에 따라 거주하고 있는 주에서 어떤 문서가 필요한지 세무 부서에 확인하시기 바랍니다.
  - 상파울루 내부 - 일반적으로 비 ICMS 신고서와 전자 세금 계산서(NF-e)가 필요합니다.
  - 상파울루 이외 리전 - 일반적으로 다음이 필요합니다.
    - ICMS를 제외한 세금 신고서
    - nota fiscal avulsa
    - 전자 세금 계산서(NF-e)

### Note

비 ICMS 납세자 신고서의 경우에는 보관용 1부와 발송용 3부 등 총 4부를 복사하는 것이 좋습니다.

## 호주에서의 Snowball Edge 픽업

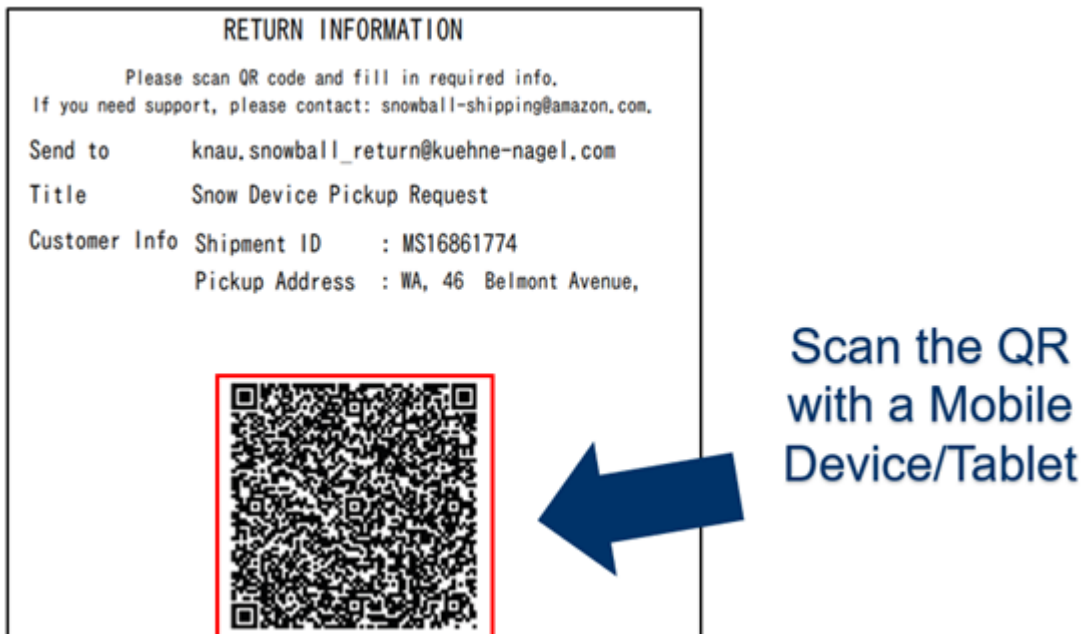
호주에서 AWS Snowball Edge 디바이스를 다시 배송하는 경우 Snow 디바이스의 E Ink 레이블 위에 반송 전송 레이블(이 지침이 들어 있는 주머니에 있음)을 AWS배치합니다.



### Note

디바이스와 함께 반품 라벨을 받지 못한 경우 디바이스 일련 번호 또는 참조 번호와 함께 [knau.snowball\\_return@kuehne-nagel.com](mailto:knau.snowball_return@kuehne-nagel.com)으로 이메일을 보내세요.

Snowball Edge의 반환을 조정하려면 모바일 디바이스를 사용하여 반환 지침의 QR 코드를 스캔합니다. 디바이스에는 이메일 메시지로 연결되는 하이퍼링크가 표시됩니다. 메시지에는 이메일 주소, 제목, 관리 번호 또는 위탁 번호와 같은 정보가 포함됩니다. 픽업 날짜, 이름, 연락처 세부 정보를 입력하거나 변경 사항이 있는 경우 새 픽업 주소를 기재합니다.



## 인도에서의 Snowball Edge 픽업

인도에서는 Blue Dart가 Snowball 디바이스를 픽업합니다. Snowball 디바이스를 반환할 준비가 되면 디바이스를 끄고 반환 배송을 위해 준비합니다. 픽업을 예약하려면 제목에 Snowball Pickup Request를 적어 [snowball-pickup@amazon.com](mailto:snowball-pickup@amazon.com) 으로 이메일을 보냅니다. 이메일 본문에 다음 정보를 포함합니다.

- 작업 ID - 반환하려는 Snowball과 연결된 작업 ID입니다 AWS.
- AWS 계정 ID - 작업을 생성한 AWS 계정의 ID입니다.
- 요청 날짜 - Snowball 디바이스를 픽업하려는 날짜입니다.
- 가장 빠른 픽업 시간(현지 시간) - 픽업을 원하는 가장 빠른 시간입니다.
- 가장 늦은 픽업 시간(현지 시간) - 픽업을 원하는 가장 늦은 시간입니다.
- 특별 지침(선택 사항) - 픽업 일정을 잡기 위한 연락처 세부 정보를 포함하여 Snowball 픽업에 대한 특별 지침입니다.

Snowball Edge 팀이 Blue Dart에서 픽업을 예약하고 확인 이메일을 보냅니다. Blue Dart는 종이 배송 라벨을 제공하고 Snowball Edge 디바이스를 픽업합니다.

### Important

- 인도에서 Snowball을 사용하는 경우 해당 주에 관련 세금 서류를 모두 제출하세요.
- AWS 인도에서 반품 배송 요청을 처리하려면 최소 72시간 전에 통지해야 합니다. 72시간 이전에 받은 반품 배송 요청에 대한 일일 요금을 환급할 AWS 수 없습니다.

## 한국에서의 Snowball Edge 픽업

한국에서 Kuehne + Nagel이 픽업을 처리합니다. 디바이스를 반송할 준비가 되면 제목 줄에 Snowball 픽업 요청이라고 입력한 후 [snowball-shipping@amazon.com](mailto:snowball-shipping@amazon.com)에 이메일을 보내 픽업을 예약할 수 있습니다. 이메일 본문에 다음 정보를 포함합니다.

- 작업 ID - 반환하려는 Snowball과 연결된 작업 ID입니다 AWS.
- 픽업 주소 - 디바이스를 픽업할 주소입니다.
- 픽업 날짜 - 디바이스를 픽업하려는 가장 빠른 날짜입니다.
- 연락처 세부 정보 - 필요한 경우 Kuehne + Nagel이 사용자와 연락하는 데 사용할 수 있는 이름, 이메일 주소 및 현지 전화 번호입니다.

곧 디바이스를 픽업하도록 제공한 주소에 대한 정보가 포함된 Snowball 팀의 후속 이메일을 수신할 것입니다. 디바이스의 전원을 껐다가 켜고 픽업 준비를 합니다. 일반적으로 1300에서 1500 사이입니다.

## 홍콩에서의 Snowball Edge 픽업

홍콩에서는 S.F. Express가 픽업을 처리합니다. 디바이스를 반환할 준비가 되면 제목 줄에 Snowball Pickup Request라고 입력한 후 snowball-shipping@amazon.com에 이메일을 보내 픽업을 예약할 수 있습니다. 이메일 본문에 다음 정보를 포함합니다.

- 작업 ID
- AWS 계정 ID
- 담당자 이름
- 연락처 전화 번호
- 연락처 이메일 주소
- 디바이스 픽업을 원하는 일자
- 가장 빠른 픽업 시간
- 가장 늦은 픽업 시간
- 픽업 주소

S.F. Express에서 픽업 날짜를 조정 한 후에는 일정을 변경할 수 없습니다.

디바이스는 S.F. Express AWS 에서 로 전달됩니다. S.F. Express 반송 추적 번호를 보면 배달 시점을 알 수 있습니다.

## 싱가포르, 일본, 인도네시아에서의 Snowball Edge 픽업

싱가포르, 일본, 인도네시아에서 디바이스를 반환할 준비가 되면 반환용 E Ink 라벨에 표시된 QR 코드를 휴대폰으로 스캔합니다. 그러면 이메일 템플릿으로 바로 이동할 수 있습니다. 픽업 날짜/시간 및 연락처 세부 정보를 입력합니다.

## RETURN

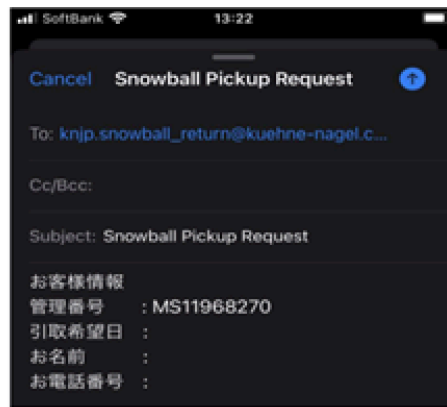
AWS Jobs ID QF6LNZGKTZPF  
 シリアル番号 2R 207138750022  
 管理番号 MS14003547



## 返送のご案内

“以下のQRコードをスキャンし情報を入力の上、  
 メールにてご連絡をお願い致します。”

送信先アドレス knjp.snowball\_return@kuehne-nagel.com  
 件名 Snow Ball Pickup Request  
 お客様情報  
 管理番号 : MS14003547  
 引取希望日 : 要記入  
 お名前 : 要記入  
 お電話番号 : 要記入



**i** Note

픽업 주소가 디바이스가 배송된 주소와 다른 경우 지정된 배송업체에 알릴 수 있도록 이메일 본문에 새 주소를 추가합니다.

**i** Note

일본의 경우 운송업체에서 120.00달러의 배송료를 부과합니다. 요금에 대한 설명은 Snowball Edge를 나타내지만 모든 Snowball Edge 배송에는 요금이 적용됩니다.

## 아랍에미리트 두바이에서의 Snowball Edge 수령 및 반품

두바이에서 AWS Snowball Edge 디바이스를 수령하거나 반환할 때 따라야 하는 몇 가지 지침은 다음과 같습니다.

### Snowball Edge 디바이스 수령

자유 구역에서 Snowball Edge 디바이스를 수령하고 UPS로부터 패키지 배송 준비가 완료되었다는 알림을 받으면 자유 구역의 게이트 패스를 신청, 획득 및 공유합니다.

자유 리전 또는 본토에 있는 경우 디바이스를 수령하면 POD(배달 증명)에 서명합니다.

## Snowball Edge 디바이스 반환

Snowball Edge 디바이스를 반환하는 경우, 600 544 743번으로 직접 또는 UPS 웹 사이트를 통해 UPS 에 픽업을 예약하여 UPS가 디바이스를 픽업하도록 합니다. 디바이스를 픽업하기 전에 E Ink 디스플레이에 반환 배송 정보가 표시되는지 확인합니다. [Snowball Edge 디바이스 반환](#)을(를) 참조하세요. 자유 구역에서 디바이스 픽업을 위해 UPS 기사가 배정되었다는 알림을 받으면 자유 구역용 게이트 패스를 신청, 획득 및 공유합니다.

E Ink 디스플레이의 선불 배송 라벨에는 Snowball Edge 디바이스를 반환할 정확한 주소가 포함되어 있습니다.

Snowball Edge 디바이스는 AWS 정렬 시설로 전달되어 AWS 데이터 센터로 전달됩니다. UPS는 작업에 대한 추적 번호를 자동으로 제공합니다.

### Important

에서 별도로 지시하지 않는 한 Snowball Edge 디바이스에 별도의 배송 라벨을 부착하지 AWS 마십시오. 항상 디바이스의 E Ink 디스플레이에 표시된 배송 라벨을 사용하세요.

제품의 Snowball Edge에 대한 UPS 서비스는 한 국가 내에서만 국내 서비스를 제공합니다.

## Snowball Edge의 배송 속도

국가마다 이용 가능한 배송 기간이 다릅니다. 이러한 배송 속도는 AWS Snowball Edge 디바이스를 배송하는 국가를 기반으로 합니다. 배송 기간은 다음과 같습니다.

- 호주, 일본, 싱가포르, 인도네시아, 대한민국 - 이들 국가 내에서 배송하는 경우 1~3일의 표준 배송 기간이 걸립니다.
- 브라질 - 브라질 내에서 배송하는 경우 영업 시간 기준 2일(영업일 기준) 이내에 배달하는 UPS Domestic Express Saver 배송을 이용할 수 있습니다. 배송 속도는 국경에서 지연되는 시간의 영향을 받을 수 있습니다.
- EU(유럽 연합) - EU 내 국가로 배송할 경우 특급 배송을 이용할 수 있습니다. 일반적으로 특급 배송된 AWS Snowball Edge 디바이스는 약 1일 후에 배송됩니다. 또한 EU 내 대부분의 국가에서는 표준 배송을 이용하며 일반적으로 편도 기준으로 1주일이 채 걸리지 않습니다.
- 홍콩 - 홍콩 내에서 배송할 경우 특급 배송을 이용할 수 있습니다.
- 인도 - 인도 내에서 배송 시 Snowball Edge 디바이스는 AWS 가 모든 관련 세금 문서를 수신한 지 7 일 이내(근무일 기준)에 발송됩니다.
- 아랍에미리트 두바이 - Courier Express Saver 배송 서비스를 이용할 수 있습니다.

- 영국 (UK) - 영국 내에서 배송하는 경우 특급 배송을 이용할 수 있습니다. 일반적으로 Snowball Edge 디바이스 특급 배송 시 약 1일 이내에 전달 완료됩니다. 또한 표준 배송을 이용하면 일반적으로 편도 기준으로 1주일 이내 걸리지 않습니다.
- 미국(US) 및 캐나다 - 미국 또는 캐나다 내에서 배송하는 경우 1일 배송 및 2일 배송 서비스를 이용할 수 있습니다.

## Snowball Edge에서 가져오기 상태 모니터링

콘솔에서 가져오기 작업의 상태를 모니터링하려면 작업이 생성된 AWS 리전의 [AWS Snow 패밀리 관리 콘솔](#)에 로그인합니다. 표에서 추적하려는 작업을 선택하거나 표 위에 있는 검색 창에서 선택한 파라미터로 검색합니다. 작업을 선택하면 작업의 실시간 상태를 나타내는 막대를 포함하여 표 내에 해당 작업에 대한 세부 정보가 나타납니다.

### Note

구성한 액세스 권한 관련 문제로 인해 Snow 디바이스에서 데이터 센터로 데이터를 가져올 수 없는 경우에는 사용자에게 이를 알리려고 시도하며, 사용자는 알림을 받은 날로부터 30일 이내에 문제를 해결할 수 있습니다. 문제가 해결되지 않으면 Snowball Edge 작업을 취소하고 디바이스에서 데이터를 삭제할 수 있습니다.

디바이스가 도착하면 AWS작업 상태가 전송 중에서 AWS로 변경됩니다. 데이터를 Amazon S3로 가져오기 시작하는 데는 평균 하루가 걸립니다. 그러면 작업 상태가 [Importing]으로 변경됩니다. Snowball Edge에서 데이터를 AWS로 가져오는 데 걸리는 시간은 Snowball Edge로 이동하는 데 걸리는 시간과 거의 동일합니다. 데이터를 가져오면 작업 상태가 완료됨 상태로 변경됩니다.

이제 사용하여 Amazon S3로 첫 번째 데이터 가져오기 작업이 AWS Snowball Edge 완료되었습니다. 콘솔에서 데이터 전송에 대한 보고서를 얻을 수 있습니다. 콘솔에서 이 보고서에 액세스하려면 테이블에서 작업을 선택하고 펼쳐서 작업의 세부 정보를 표시합니다. [Get report]를 선택하여 작업 완료 보고서를 PDF 파일로 다운로드합니다. 자세한 내용은 [데이터 전송 작업 완료 보고서 및 로그 가져오기](#) 단원을 참조하십시오.

다음: [데이터 전송 작업 완료 보고서 및 로그 가져오기](#)

## 데이터 전송 작업 완료 보고서 및 로그 가져오기

Snowball Edge를 사용하여 Amazon S3로 데이터를 가져오거나 Amazon S3에서 데이터를 내보내면 다운로드 가능한 PDF 작업 보고서를 얻을 수 있습니다. 가져오기 작업의 경우 가져오기 프로세스가 끝



날 때 이 보고서를 사용할 수 있습니다. 내보내기 작업의 경우 일반적으로 작업 부분의 AWS Snowball Edge 디바이스가 전달되는 동안 작업 보고서를 사용할 수 있습니다. 로컬 컴퓨팅 및 스토리지 전용 작업에는 작업 완료 보고서를 사용할 수 없습니다.

작업 보고서는 Amazon S3 데이터 전송 상태에 대한 인사이트를 제공합니다. 보고서에는 기록 보관을 위한 작업 또는 작업 파트에 대한 세부 정보가 포함됩니다. 또한 작업 보고서에는 디바이스와 Amazon S3 간에 전송되는 총 객체 수와 바이트 수에 대한 종합적인 개요를 제공하는 테이블이 포함되어 있습니다.

전송되는 객체의 상태에 대해 더 심층적인 이해가 필요한 경우 성공 로그 및 실패 로그라는 두 가지 관련 로그를 살펴볼 수 있습니다. 로그는 CSV(쉼표로 분리된 값) 형식으로 저장되며 각 로그의 이름에는 로그에서 설명하는 작업 또는 작업 파트의 ID가 포함됩니다.

AWS Snow 패밀리 관리 콘솔에서 보고서와 로그를 다운로드할 수 있습니다. 다음은 샘플 보고서입니다.

## Snow Family Job Completion Report



**Region:** us-gov-east-1(OSU)  
**Job ID:** JIDd6d95004-fe1a-42d3-895d-684f357ef840  
**Snow Device Serial ID:** 207117851234  
**Job type:** IMPORT  
**Device type:** Snowball Edge Storage Optimized  
**Storage type:** S3  
**Job creation date:** 2022-06-02 19:32:27.831 GMT  
**Job state:** Completed  
**Customer address:**  
 123 Any Street  
 Any Town, USA

### Transfer details:

Transfer type	Total	Success	Failed
Objects	2,635	2,635	0
Bytes	32.2 TB	32.2 TB	0 B

### Job state transition details:

The job was created on 2022-06-02 19:32:27.831 GMT  
 The snowball got allocated on 2022-06-06 19:10:43.670 GMT  
 The snowball was shipped on 2022-06-07 21:59:50.937 GMT  
 The snowball was at customer on 2022-06-08 14:04:45.856 GMT  
 The snowball was shipped to AWS on 2022-06-28 20:57:42.246 GMT  
 The snowball was at our sorting facility on 2022-06-29 14:06:20.737 GMT  
 The snowball was at AWS on 2022-06-30 23:12:45.017 GMT  
 The data transfer started on 2022-06-30 23:21:34.805 GMT  
 The data transfer was completed on +54473-09-10 22:23:46 GMT

Please review your job's status from the console.  
 For Snow job details, please see: <https://docs.aws.amazon.com/snowball/>

작업 보고서 및 로그를 가져오려면

1. 에 로그인 AWS Management Console 하고를 엽니다 [AWS Snow 패밀리 관리 콘솔](#).
2. 테이블에서 작업 또는 작업 부분을 선택하고 상태 창을 확장합니다.

작업 보고서 및 로그 가져오기에 대한 세 가지 옵션, 즉 [Get job report], [Download success log] 및 [Download failure log]가 나타납니다.

### 3. 다운로드할 로그를 선택합니다.

다음 목록에서는 보고서에 대해 가능한 값을 설명합니다.

- 완료됨 - 전송이 성공적으로 완료되었습니다. 성공 로그에서 더 자세한 정보를 확인할 수 있습니다.
- 오류와 함께 완료됨 - 데이터의 일부 또는 전부가 전송되지 않았습니다. 실패 로그에서 더 자세한 정보를 확인할 수 있습니다.

# Snowball Edge를 사용한 대규모 데이터 마이그레이션

온프레미스 위치에서 대용량 데이터를 마이그레이션하려면 신중한 계획, 오케스트레이션, 실행이 필요합니다. 그래야만 데이터를 성공적으로 AWS에 마이그레이션할 수 있습니다.

기한을 놓치고 예산을 초과하며 마이그레이션 실패가 발생할 가능성을 방지하기 위해 마이그레이션을 시작하기 전에 데이터 마이그레이션 전략을 마련하는 것이 좋습니다. AWS Snow 서비스는 Snowball Edge Large Data Migration Manager(LDMM) 기능을 통해 대규모 데이터 마이그레이션 프로젝트를 배치, 주문 및 추적하는 데 도움이 됩니다 AWS Snow 패밀리 관리 콘솔.

주제, [Snowball Edge를 사용하여 대규모 전송 계획](#) 및 [Snowball Edge를 사용하여 대규모 전송 보정](#)에서는 수동 데이터 마이그레이션 프로세스를 설명합니다. Snowball Edge LDMM 마이그레이션 계획을 사용하여 수동 단계를 간소화할 수 있습니다.

주제

- [Snowball Edge를 사용하여 대규모 전송 계획](#)
- [Snowball Edge를 사용하여 대규모 전송 보정](#)
- [Snowball Edge를 사용하여 대규모 데이터 마이그레이션 계획 생성](#)
- [Snowball Edge에서 대규모 데이터 마이그레이션 계획 사용](#)

## Snowball Edge를 사용하여 대규모 전송 계획

다음 섹션의 지침에 따라 현장에 있는 AWS Snowball Edge 디바이스와 서버 간의 대용량 데이터 전송을 계획하고 보정하는 것이 좋습니다.

주제

- [1단계: 클라우드로 이동하는 대상 이해](#)
- [2단계: 대상 전송 속도 계산](#)
- [3단계: 필요한 Snowball Edge 수 결정](#)
- [4단계: 작업 생성](#)
- [5단계: 데이터를 전송 세그먼트로 분리](#)

## 1단계: 클라우드로 이동하는 대상 이해

를 사용하여 첫 번째 작업을 생성하기 전에 전송해야 하는 데이터의 양 AWS Snow 패밀리 관리 콘솔, 현재 저장된 위치 및 전송하려는 대상을 평가해야 합니다. 규모가 페타바이트 이상인 데이터 전송의 경우 관리 정리 작업을 사용하면 Snowball Edge가 도착할 때 훨씬 더 쉬워집니다.

데이터를 AWS 클라우드로 처음 마이그레이션하는 경우 클라우드 마이그레이션 모델을 설계하는 것이 좋습니다. 클라우드 마이그레이션은 하룻밤 사이에 이루어지지 않습니다. 모든 시스템이 예상대로 작동하도록 하려면 신중한 계획 프로세스를 거쳐야 합니다.

이 단계를 완료하면 클라우드로 이동할 총 데이터 양을 알게 됩니다.

## 2단계: 대상 전송 속도 계산

각 서버에 연결된 Snowball Edge로 데이터를 얼마나 빨리 전송할 수 있는지 추정하는 것이 중요합니다. 이 예상 속도(MB/초)는 로컬 네트워크 인프라를 사용하여 데이터 소스에서 Snowball Edge 디바이스로 데이터를 전송할 수 있는 속도를 결정합니다.

### Note

대용량 데이터 전송의 경우 Amazon S3를 사용하여 데이터를 전송하는 것이 좋습니다. AWS Snow 패밀리 관리 콘솔에서 디바이스를 주문할 때 이 옵션을 선택해야 합니다.

기본 전송 속도를 결정하려면 데이터의 소규모 하위 집합을 Snowball Edge 디바이스로 전송하거나 10GB의 샘플 파일을 전송하고 처리량을 관찰합니다.

대상 전송 속도를 확인하는 동안 네트워크 속도, 전송 중인 파일의 크기, 로컬 서버에서 데이터를 읽는 속도를 변경하는 등 환경(네트워크 구성 포함)을 조정하여 처리량을 개선할 수 있다는 점에 유의해야 합니다. Amazon S3 어댑터는 조건이 허용하는 한 빨리 Snowball Edge에 데이터를 복사합니다.

## 3단계: 필요한 Snowball Edge 수 결정

클라우드로 이동하려는 총 데이터 양, 예상 전송 속도 및가 데이터를 이동할 수 있도록 허용하려는 일수를 사용하여 대규모 데이터 마이그레이션에 필요한 Snowball Edge 수를 AWS결정합니다. 디바이스 유형에 따라 Snowball Edge 디바이스에는 약 39.5TB 또는 210TB의 사용 가능한 스토리지 스페이스가 있습니다. 예를 들어 300TB의 데이터를 10일 AWS 이상으로 이동하고 전송 속도가 250MB/s인 경우 210TB의 스토리지가 있는 Snowball Edge 디바이스 2개가 필요합니다.

**Note**

Snowball Edge LDMM은 동시에 지원할 수 있는 Snowball Edge 수를 추정하는 마법사를 제공합니다. 자세한 내용은 [Snowball Edge를 사용하여 대규모 데이터 마이그레이션 계획 생성 단원](#)을 참조하십시오.

## 4단계: 작업 생성

필요한 Snowball Edge 수를 파악한 후에는 각 디바이스에 대한 가져오기 작업을 생성해야 합니다. 여러 작업 생성은 Snowball Edge LDMM에 의해 간소화됩니다. 자세한 내용은 [다음 작업 주문하기](#) 단원을 참조하십시오.

**Note**

권장 작업 주문 일정에서 바로 다음 작업을 주문하고 계획에 자동으로 추가할 수 있습니다. 자세한 내용은 [권장 작업 주문 일정](#) 단원을 참조하십시오.

## 5단계: 데이터를 전송 세그먼트로 분리

여러 개의 작업이 관련된 대용량 데이터 전송의 경우 다수의 소용량 관리 가능 데이터 집합으로 데이터를 로컬에서 분리하는 것이 좋습니다. 이렇게 하면 각 파티션을 한 번에 전송하거나 여러 파티션을 병렬로 전송할 수 있습니다. 파티션을 계획할 때 결합된 파티션의 데이터가 작업의 Snowball Edge에 맞는지 확인합니다. 예를 들어 다음과 같은 방법으로 전송을 여러 파티션으로 분리할 수 있습니다.

- 예를 들어 210TB의 스토리지를 갖춘 Snowball Edge 디바이스와 함께 사용할 수 있도록 각각 20TB의 파티션 10개를 생성할 수 있습니다.
- 대용량 파일의 경우 각 파일을 Amazon S3의 객체 크기 제한인 최대 5TB에 상당하는 개별 파티션으로 만들 수 있습니다.
- 각 파티션은 크기가 다를 수 있으며 각 개별 파티션은 동일한 종류의 데이터로 구성될 수 있습니다. 예를 들어 한 파티션에는 소용량 파일, 다른 파티션에는 압축된 아카이브, 다른 파티션에는 대용량 파일 등이 있을 수 있습니다. 이 접근법은 다양한 종류의 파일에 대한 평균 전송 속도를 확인하는 데 도움이 될 수 있습니다.

**Note**

전송되는 각 파일에 대해 메타데이터 작업이 수행됩니다. 이 오버헤드는 파일 크기와 상관없이 동일하게 유지됩니다. 따라서 작은 파일을 더 큰 번들로 압축하거나 파일을 배치 처리하거나 더 큰 개별 파일을 전송하면 더 빠른 성능을 이용할 수 있습니다.

전송을 1일 이상 실행한 후 대용량 이기종 전송의 문제를 해결하려고 할 경우 작업이 복잡할 수 있기 때문에 데이터 전송 세그먼트를 생성하면 해당 문제를 더 쉽게 빨리 해결할 수 있습니다.

페타바이트 규모의 데이터 전송 계획을 마치면 서버에서 Snowball Edge 디바이스로 몇 개의 세그먼트를 전송하여 속도와 총 전송 시간을 보정하는 것이 좋습니다.

## Snowball Edge를 사용하여 대규모 전송 보정

대표적인 데이터 파티션 세트를 전송하여 전송 성능을 조정할 수 있습니다. 정의한 여러 파티션을 선택하고 Snowball Edge 디바이스로 전송합니다. 각 작업의 전송 속도와 총 전송 시간을 기록합니다. 보정 결과가 대상 전송 속도보다 낮으면 동시에 데이터 전송의 여러 부분을 병렬로 복사할 수 있습니다. 이 경우 데이터 세트의 추가 파티션을 사용하여 보정을 반복합니다.

현재 전송 중인 전체 인스턴스의 전송 속도 합계가 감소할 때까지 보정 중에 병렬 복사 작업을 계속 추가합니다. 마지막 활성 인스턴스를 종료하고 새로운 대상 전송 속도를 기록합니다.

다음 시나리오 중 하나를 사용하여 데이터를 병렬로 전송하여 Snowball Edge로 더 빠르게 데이터를 전송할 수 있습니다.

- 단일 Snowball Edge 디바이스에 대해 워크스테이션에서 S3 어댑터의 여러 세션을 사용합니다.
- 단일 Snowball Edge 디바이스에 대해 여러 워크스테이션에서 S3 어댑터의 여러 세션을 사용합니다.
- 여러 Snowball Edge를 대상으로 하는 S3 인터페이스의 여러 세션 사용(단일 또는 여러 워크스테이션 사용).

이 단계를 완료하면 Snowball Edge 디바이스로 데이터를 얼마나 빨리 전송할 수 있는지 알아야 합니다.

# Snowball Edge를 사용하여 대규모 데이터 마이그레이션 계획 생성

Snowball Edge 대규모 데이터 마이그레이션 계획 기능을 사용하면 여러 Snowball Edge 서비스 제품을 사용하여 500TB에서 여러 페타바이트로의 대규모 데이터 마이그레이션을 계획, 추적, 모니터링 및 관리할 수 있습니다.

대규모 데이터 마이그레이션 계획 기능을 사용하여 이동할 데이터의 크기 AWS 및 데이터를 동시에 마이그레이션하는 데 필요한 Snowball Edge 수와 같은 데이터 마이그레이션 목표에 대한 정보를 수집할 수 있습니다. 이 계획을 사용하여 데이터 마이그레이션 프로젝트의 예상 일정과 목표 달성을 위한 권장 작업 주문 일정을 세우도록 합니다.

## Note

현재 데이터 마이그레이션 계획은 500TB보다 큰 가져오기 작업에 사용할 수 있습니다.

## 주제

- [1단계: 마이그레이션 세부 정보 선택](#)
- [2단계: 배송, 보안 및 알림 기본 설정 선택](#)
- [3단계: 계획 검토 및 생성](#)

## 1단계: 마이그레이션 세부 정보 선택


## Note

500TB를 초과하는 데이터 마이그레이션에는 대규모 데이터 마이그레이션 계획을 사용할 수 있습니다. Snowball Edge에서 500TB 미만의 데이터 전송 프로젝트에 대한 작업 주문을 개별적으로 생성합니다. 자세한 내용은 이 설명서의 [Snowball Edge 디바이스 주문 작업 생성을 참조하세요](#).

1. [AWS Snow 패밀리 관리 콘솔](#)에 로그인합니다. 이 AWS Snow 패밀리 관리 콘솔 에서를 처음 사용하는 경우 Snowball Edge 페이지가 AWS 리전 표시됩니다. 그렇지 않으면 기존 작업 목록이 표시됩니다.
2. 처음으로 데이터 마이그레이션 계획을 세우는 경우에는 기본 페이지에서 대규모 데이터 마이그레이션 계획 생성을 선택합니다. 그렇지 않으면 대규모 데이터 마이그레이션 계획을 선택합니다. 데이터 마이그레이션 계획 생성을 선택하여 계획 생성 마법사를 엽니다.



3. 데이터 마이그레이션 계획 이름 지정에 데이터 마이그레이션 계획 이름을 입력합니다. 계획 이름은 최대 64자까지 가능합니다. 유효한 문자는 A~Z, a~z, 0~9 및 -(하이픈)입니다. 계획 이름은 **aws:**로 시작해서는 안 됩니다.
4. 마이그레이션할 총 데이터에 AWS 마이그레이션할 데이터의 양을 입력합니다 AWS.
5. Snow 디바이스에서 Snowball Edge 디바이스를 선택합니다.

 Note

지원되는 디바이스 옵션은 특정 AWS 리전의 가용성에 따라 달라질 수 있습니다.

6. 동시 디바이스의 경우 해당 위치에서 동시에 데이터를 복사할 수 있는 Snowball Edge 수를 입력합니다. 잘 모르겠으면 다음 섹션으로 건너뛰어 동시 사용 가능 디바이스 추정 마법사를 사용하여 이를 확인하는 방법에 대한 정보를 확인합니다.
7. Next(다음)를 선택합니다.

## 동시 사용 가능 디바이스 추정기 마법사 사용

동시 사용 가능 디바이스 추정 마법사는 대규모 데이터 마이그레이션 중에 사용할 수 있는 동시 사용 가능 디바이스 수를 결정하는 데 도움이 됩니다.

사전 조건:

- 개념 증명을 수행하여 데이터 전송 방법론을 테스트하고 환경에서 Snowball Edge 디바이스로 성능을 측정했습니다.
- 네트워크 및 백엔드 스토리지 연결을 알고 있습니다.

### 1단계: 데이터 소스 정보 입력

먼저, 스토리지 소스에서 데이터를 복사하기 위한 이론적 최대 처리량을 결정합니다.

1. 마이그레이션할 총 데이터에는 마이그레이션할 데이터의 양을 입력합니다.

마이그레이션하려는 데이터 양의 측정 단위(GB 또는 TB)를 단위에서 선택합니다.

2. 활성 네트워크 인터페이스 수에는 스토리지 소스에서 데이터를 마이그레이션하는 데 사용할 수 있는 활성 네트워크 인터페이스 수를 입력합니다.

**Number of active network interfaces** [Info](#)

The number of network interfaces that can be used for migrations

Number of active network interfaces used for data migration

3. 스토리지 소스의 네트워크 인터페이스 속도를 네트워크 인터페이스 속도에서 선택합니다. 네트워크 속도 단위는 Gb/s입니다.

**Network interface speed** [Info](#)

The speed of the network interfaces used for migrations

Network interface speed (Gb/s)

4. 최대 네트워크 처리량에는 개념 증명 과정에서 결정한 스토리지 소스의 테스트된 최대 네트워크 처리량을 입력합니다. 처리량 단위는 MB/S입니다.

**Maximum network throughput** [Info](#)

The maximum sustainable throughput for the data source

Maximum tested throughput of data source (MB/s)

5. 스토리지 백엔드 네트워크 사용은 스토리지 소스가 백엔드 스토리지와 네트워크를 공유하는지 여부를 나타냅니다.
- 네트워크가 공유되지 않는 경우 예를 선택합니다. 단일 스트림의 경우 스토리지 상호 연결 속도를 입력할 필요가 없습니다.
  - 네트워크가 공유되는 경우 아니요를 선택합니다. 단일 스트림의 스토리지 상호 연결 속도를 MB/s 단위로 입력합니다.

선택에 따라 마법사는 페이지 하단의 데이터 소스 값에 대한 최대 마이그레이션 처리량(MB/s)을 업데이트합니다.

### Storage backend network usage [Info](#)

**Network shared with storage backend traffic?**  
Is the network used for migration being shared with your storage backend?

Yes ▼

**Speed of storage interconnection for single stream (MB/s)**  
This is a single connection throughput that can be sustained from source to destination

6. Next(다음)를 선택합니다.

2단계: 마이그레이션 워크스테이션 파라미터 입력

yourSnowball Edge를 스토리지 소스(예: Microsoft Windows 서버)에 직접 연결할 수 있습니다. 대신 yourSnowball Edge를 하나 이상의 워크스테이션에 연결하여 스토리지 소스에서 데이터를 복사하도록 선택할 수 있습니다.

1. 마이그레이션 워크스테이션 사용에 워크스테이션 사용 선택 사항을 표시합니다.
  - 워크스테이션을 사용하지 않고 데이터 소스에서 직접 데이터를 전송하려면 없음 - 데이터 소스 직접 사용을 선택하고 다음을 선택합니다.
  - 데이터 전송에 하나 이상의 워크스테이션을 사용하려면 기타 - 워크스테이션 복사 사용을 선택합니다.

### Migration workstation usage [Info](#)

Type of migration source used

Other - Use copy workstation(s) ▼

2. 활성 네트워크 인터페이스 수에 데이터 마이그레이션에 사용할 포트 수를 입력합니다.

**Number of active network interfaces** [Info](#)

The number of network interfaces that can be used for migrations

Number of active network interfaces on the migration workstation

1

3. 네트워크 인터페이스 속도에서 네트워크 인터페이스의 속도(Gb/s)를 선택합니다.

**Network interface speed** [Info](#)

Your workstations Network card speeds

Network interface speed (Gb/s)

10

4. 스토리지 백엔드 네트워크 사용은 워크스테이션이 있는 네트워크가 백엔드 스토리지와 공유되는지 여부를 나타냅니다.
- 공유된 경우 예를 선택합니다.
  - 공유된 경우 아니요를 선택합니다. 단일 스트림의 스토리지 상호 연결 속도를 MB/s 단위로 입력합니다.

**Storage backend network usage** [Info](#)

Network shared with storage backend traffic?

Is the network used for migration being shared with your storage backend?

Yes

Speed of storage interconnection for single stream (MB/s)

This is a single connection throughput that can be sustained from source to destination

입력한 내용에 따라 마법사는 마이그레이션 워크스테이션 수에 권장 사항을 표시합니다. 권장 사항에 동의하지 않는 경우 수치를 수동으로 변경할 수 있습니다. 이 수치는 대규모 데이터 마이그레이션 계획의 동시 사용 가능 디바이스에 표시됩니다.

### Number of migration workstations [Info](#)

Recommended number of migration workstations used

### 3단계: Snowball Edge의 입력 평균 전송 처리량

1. 평균 Snow 디바이스 전송 처리량 필드에 개념 증명 중에 확인한 전송 처리량(Mb/s)을 입력합니다.

### Average Snow device transfer throughput [Info](#)

This is the throughput from your migration workstation to the Snow device you saw during the proof of concept

Average Snow device transfer throughput (MB/s)

평균 처리량을 기준으로 마법사는 마이그레이션 계획 세부 정보에서 권장 동시 사용 가능 Snow 디바이스 수와 최대 동시 사용 가능 디바이스 수를 업데이트합니다.

2. 계속하려면 이 수치 사용을 선택하고 마이그레이션 세부 정보 선택으로 돌아갑니다. 다음을 선택하여 다음 단계([2단계: 배송, 보안 및 알림 기본 설정 선택](#))로 진행합니다.

#### Note

최대 5대의 Snow 디바이스를 동시에 사용할 수 있습니다.

### 2단계: 배송, 보안 및 알림 기본 설정 선택

1. 배송 주소 섹션에서 기존 주소를 선택하거나 새로 생성합니다.

• **Note**

해당 주소의 국가는 디바이스의 목적지 국가와 일치해야 하며 해당 국가에서 유효해야 합니다.

2. 서비스 액세스 유형 선택에서 다음 중 하나를 수행합니다.
  - Snowball Edge가 Snowball Edge 작업에 대한 CloudWatch 지표 및 Amazon SNS 알림을 게시하는 데 필요한 모든 권한을 가진 새 서비스 연결 역할을 생성하도록 허용합니다.
  - 필요한 권한이 있는 기존 서비스 역할을 추가합니다. 이 역할을 설정하는 방법의 예시는 [Example 4: Expected Role Permissions and Trust Policy](#) 섹션을 참조하세요.
3. 알림 전송에서 알림 전송 여부를 선택합니다. 데이터 마이그레이션 계획에 대한 알림을 전송하지 않음을 선택하면 이 계획에서 알림을 받지 못하지만 작업 알림은 계속 받게 된다는 점에 유의하세요.
4. 알림 설정에서
  - 기존 SNS 주제 사용을 선택합니다.
  - 또는 새 SNS 주제 생성을 선택합니다.

### 3단계: 계획 검토 및 생성

1. 계획 세부 정보 및 배송, 보안 및 알림 기본 설정의 정보를 검토하고 필요한 경우 편집합니다.
2. 데이터 마이그레이션 계획 생성을 선택하여 계획을 생성합니다.

## Snowball Edge에서 대규모 데이터 마이그레이션 계획 사용

대규모 데이터 마이그레이션 계획을 생성한 후 결과 일정과 대시보드를 사용하여 나머지 마이그레이션 프로세스를 안내할 수 있습니다.

### 권장 작업 주문 일정

Snowball Edge 대규모 마이그레이션 계획을 생성한 후 권장 작업 주문 일정을 사용하여 새 작업을 생성할 수 있습니다.

**Note**

데이터 크기나 동시 사용 가능 디바이스 수를 수동으로 업데이트하면 일정이 조정됩니다. 작업이 권장 주문 날짜까지 주문되지 않았거나 권장 주문 날짜 이전에 주문된 경우 일정이 자동으로 조정됩니다. 권장 주문 날짜 이전에 작업이 반환되는 경우 일정이 자동으로 조정됩니다.

Recommended job ordering schedule		Jobs ordered				
Recommended job ordering schedule						
This list provides an estimated schedule to place Snow jobs in order to achieve your data migration goals. The estimated ordering schedule is automatically adjusted based on your data migration speed.						
Filter by a date and time range		<input type="button" value="Refresh"/>	<input checked="" type="checkbox"/> Hide Ordered	Actions ▾		
< 1 >						
	Recommended date to order	Number of devices to order	Number of ordered devices	Device type	Status	
<input type="radio"/>	Thu Mar 23 2023	2	-	Snowball Edge Storage Optimized with 210TB	⊙ Not Ordered	
<input type="radio"/>	Fri Mar 31 2023	2	-	Snowball Edge Storage Optimized with 210TB	⊙ Not Ordered	
<input type="radio"/>	Sat Apr 08 2023	2	-	Snowball Edge Storage Optimized with 210TB	⊙ Not Ordered	
<input type="radio"/>	Sun Apr 16 2023	2	-	Snowball Edge Storage Optimized with 80TB	⊙ Not Ordered	
<input type="radio"/>	Mon Apr 24 2023	2	-	Snowball Edge Storage Optimized with 80TB	⊙ Not Ordered	
<input type="radio"/>	Tue May 02 2023	2	-	Snowball Edge Storage Optimized with 80TB	⊙ Not Ordered	
<input type="radio"/>	Wed May 10 2023	2	-	Snowball Edge Storage Optimized with 80TB	⊙ Not Ordered	
<input type="radio"/>	Thu May 18 2023	2	-	Snowball Edge Storage Optimized with 80TB	⊙ Not Ordered	
<input type="radio"/>	Fri May 26 2023	1	-	Snowball Edge Storage Optimized with 80TB	⊙ Not Ordered	
<input type="radio"/>	Fri May 26 2023	1	-	Snowcone SSD	⊙ Not Ordered	

**다음 작업 주문하기**

다음 주문을 하려면 수동으로 작업을 생성한 다음 계획에 추가하지 않고 이전에 주문한 작업을 복제하거나 미리 채워진 작업을 생성할 수 있습니다.

작업을 복제하는 방법은 다음과 같습니다.

1. 권장 작업 주문 일정에서 다음 주문(주문되지 않음 상태의 첫 번째 권장 사항)을 선택한 다음 작업 메뉴에서 작업 복제를 선택합니다. 작업 복제 창이 나타납니다.
2. 작업 복제 창의 주문된 작업 섹션에서 복제할 작업을 선택합니다.
3. 새 작업 세부 정보 섹션에서 주문하려는 디바이스를 선택합니다. 선택한 각 디바이스에서 작업 이름은 선택한 작업에 따라 자동으로 채워집니다. 작업 이름을 덮어쓸 수 있습니다.
4. 확인을 선택하여 선택한 디바이스에 대한 작업을 주문합니다. 시스템이 각 디바이스의 작업을 복제합니다.

새 작업을 생성하는 방법은 다음과 같습니다.

1. 권장 작업 주문 일정에서 다음 주문(주문되지 않음 상태의 첫 번째 권장 사항)을 선택한 다음 작업 메뉴에서 새 작업 생성을 선택합니다. 새 작업 생성 창이 나타납니다.

Recommended date to order	Number of devices to order	Number of ordered devices	Device type	Status
<input checked="" type="radio"/> Thu Mar 23 2023	2	-	Snowball Edge Storage Optimized with 210TB	<input type="radio"/> Not Ordered
<input type="radio"/> Fri Mar 31 2023	2	-	Snowball Edge Storage Optimized with 210TB	<input type="radio"/> Not Ordered
<input type="radio"/> Sat Apr 08 2023	2	-	Snowball Edge Storage Optimized with 210TB	<input type="radio"/> Not Ordered
<input type="radio"/> Sun Apr 16 2023	2	-	Snowball Edge Storage Optimized with 80TB	<input type="radio"/> Not Ordered
<input type="radio"/> Mon Apr 24 2023	2	-	Snowball Edge Storage Optimized with 80TB	<input type="radio"/> Not Ordered
<input type="radio"/> Tue May 02 2023	2	-	Snowball Edge Storage Optimized with 80TB	<input type="radio"/> Not Ordered
<input type="radio"/> Wed May 10 2023	2	-	Snowball Edge Storage Optimized with 80TB	<input type="radio"/> Not Ordered
<input type="radio"/> Thu May 18 2023	2	-	Snowball Edge Storage Optimized with 80TB	<input type="radio"/> Not Ordered
<input type="radio"/> Fri May 26 2023	1	-	Snowball Edge Storage Optimized with 80TB	<input type="radio"/> Not Ordered
<input type="radio"/> Fri May 26 2023	1	-	Snowcone SSD	<input type="radio"/> Not Ordered

2. 디바이스 선택 섹션에서 주문하려는 디바이스를 선택합니다. Continue(계속)을 선택합니다.

### Create New Jobs

#### Device Selection (2/2)

Select which devices you would like to order

- Device type
- Snowball Edge Storage Optimized with 210TB
- Snowball Edge Storage Optimized with 210TB

Cancel Continue

3. 새로 생성 페이지가 나타납니다. 작업 유형, 배송지 주소, 디바이스 유형과 같은 대부분의 파라미터는 계획에 따라 설정됩니다. 시스템이 각 디바이스의 작업을 생성합니다.

작업이 성공적으로 생성되었는지 여부를 확인할 수 있습니다. 성공적으로 생성된 작업은 계획에 자동으로 추가됩니다.



## 주문된 작업 목록

각 계획에는 주문된 작업 목록이 표시됩니다. 처음에는 비어 있습니다. 작업을 주문하기 시작하면 작업 메뉴에서 작업 추가를 선택하여 계획에 추가할 수 있습니다. 여기에 추가하는 작업은 모니터링 대시보드에서 추적됩니다.

마찬가지로 작업 메뉴에서 작업 제거를 선택하여 주문된 작업 목록에서 제거할 수 있습니다.

원활한 데이터 마이그레이션을 위해 계획에 제공된 작업 주문 일정을 사용하는 것이 좋습니다.

## 모니터링 대시보드

계획에 작업을 추가한 후 작업이 수집을 AWS 위해 로 돌아가면 대시보드에서 지표를 볼 수 있습니다. 다음 지표는 진행 상황을 추적하는 데 도움이 될 수 있습니다.

- 로 마이그레이션된 데이터 AWS - AWS 지금까지 로 마이그레이션된 데이터의 양입니다.
- 작업당 마이그레이션된 평균 데이터 - 작업당 평균 데이터 양(테라바이트).
- 전체 Snow 작업 - 주문한 Snowball Edge 작업 수와 주문해야 할 나머지 작업 수를 비교한 값.
- 마이그레이션 작업의 평균 기간 - 작업의 평균 기간(일).
- Snow 작업 상태 - 각 상태의 작업 수.

# AWS OpsHub 를 사용하여 디바이스 관리

이제 Snowball Edge는 디바이스 및 로컬 AWS 서비스를 관리하는 데 사용할 수 있는 AWS OpsHub 있는 사용자 친화적인 도구를 제공합니다. 클라이언트 컴퓨터에서 AWS OpsHub 를 사용하여 단일 또는 클러스터 디바이스 잠금 해제 및 구성, 파일 전송, Snowball Edge에서 실행되는 인스턴스 시작 및 관리와 같은 작업을 수행합니다. AWS OpsHub 를 사용하여 스토리지 최적화 및 컴퓨팅 최적화 Snow 디바이스 유형을 모두 관리할 수 있습니다. 추가 비용 없이 AWS OpsHub 애플리케이션을 사용할 수 있습니다.

AWS OpsHub 는 Snowball API에서 사용할 수 있는 기존 작업을 모두 가져와 그래픽 사용자 인터페이스로 표시합니다. 이 인터페이스를 사용하면 데이터를 로 빠르게 마이그레이션 AWS 클라우드 하고 Snowball Edge에 엣지 컴퓨팅 애플리케이션을 배포할 수 있습니다.

AWS OpsHub 는 Snowball Edge에서 실행 중인 AWS 서비스에 대한 통합 보기를 제공하고를 통해 운영 작업을 자동화합니다 AWS Systems Manager. AWS OpsHub를 사용하면 다양한 수준의 기술 전문 지식을 갖춘 사용자가 많은 수의 Snowball Edge를 관리할 수 있습니다. 몇 번의 클릭만으로 디바이스의 잠금을 해제하고, 파일을 전송하고, Amazon EC2 호환 인스턴스를 관리하고, 디바이스 지표를 모니터링할 수 있습니다.

Snow 디바이스가 사이트에 도착하면 노트북과 같은 클라이언트 시스템에서 AWS OpsHub 애플리케이션을 다운로드, 설치 및 시작합니다. 설치 후 디바이스를 잠금 해제하고 로컬에서 지원되는 AWS 서비스를 사용하여 디바이스를 관리할 수 있습니다.는 디바이스의 스토리지 용량 및 활성 인스턴스와 같은 주요 지표를 요약하는 대시보드를 AWS OpsHub 제공합니다. 또한 Snowball Edge에서 지원되는 다양한 AWS 서비스를 제공합니다. 몇 분 안에 파일을 디바이스로 전송할 수 있습니다.

## 주제

- [AWS OpsHub Snowball Edge용 다운로드](#)
- [를 사용하여 Snowball Edge 디바이스 잠금 해제 AWS OpsHub](#)
- [AWS OpsHub 의 PGP 서명 확인\(선택 사항\)](#)
- [를 사용하여 Snowball Edge에서 AWS 서비스 관리 AWS OpsHub](#)
- [를 사용하여 디바이스 재부팅 AWS OpsHub](#)
- [를 사용하여 프로필 관리 AWS OpsHub](#)
- [를 사용하여 디바이스 종료 AWS OpsHub](#)
- [를 사용하여 디바이스 별칭 편집 AWS OpsHub](#)
- [OpsHub를 사용한 공개 키 인증서 관리](#)
- [Snowball Edge에 대한 업데이트 가져오기](#)

- [AWS OpsHub 애플리케이션 업데이트](#)
- [를 사용하여 관리 작업 자동화 AWS OpsHub](#)
- [를 사용하여 디바이스의 NTP 시간 서버 설정 AWS OpsHub](#)

## AWS OpsHub Snowball Edge용 다운로드

다운로드 AWS OpsHub

1. [AWS Snowball 리소스 웹 사이트](#)로 이동합니다.

The screenshot shows the AWS OpsHub website interface. On the left is a navigation menu with links for 'What's New', 'OpsHub', 'Snowball Edge Client', 'Featured Content', 'Documentation', 'Webinars', 'Videos', 'Blog Posts', and 'Free Training'. The main content area is titled 'OpsHub' and contains a description: 'OpsHub is a graphical user interface you can use to manage Snowball devices. OpsHub makes it easy to setup and manage Snowball devices enabling you to rapidly deploy edge computing workloads and simplify data migration to the cloud. With just a few clicks in OpsHub, you have the full functionality of the Snow Family of devices at your fingertips; you can unlock and configure devices, drag-and-drop data to devices, launch applications, and monitor device metrics.' Below the description is a link to 'OpsHub documentation'. At the bottom, there is a table of download links for different operating systems:

	OpsHub
Windows 7 or higher	<a href="#">Download</a>
Mac OS X 10.10 or higher	<a href="#">Download</a>
Linux (Ubuntu version 14 or higher, and Fedora version 24 or higher)	<a href="#">Download</a>
	<a href="#">(Signature)</a>

2. AWS OpsHub 섹션에서 운영 체제에 맞는 다운로드를 선택하고 설치 단계를 따릅니다.

## 를 사용하여 Snowball Edge 디바이스 잠금 해제 AWS OpsHub

디바이스가 사이트에 도착하면 첫 번째 단계는 디바이스를 연결하고 잠금을 해제하는 것입니다. AWS OpsHub 를 통해 다음 방법으로 디바이스에 로그인하고 잠금을 해제하고 디바이스를 관리할 수 있습니다.

- 로컬 - 디바이스에 로컬로 로그인하려면 디바이스의 전원을 켜고 로컬 네트워크에 연결해야 합니다. 그런 다음 잠금 해제 코드와 매니페스트 파일을 제공합니다.
- 원격 - 디바이스에 원격으로 로그인하려면 디바이스의 전원을 켜고 네트워크를 통해 *device-order-region*.amazonaws.com에 연결할 수 있는지 확인해야 합니다. 그런 다음 디바이스에

AWS 계정 연결된에 대한 AWS Identity and Access Management (IAM) 자격 증명(액세스 키 및 보안 키)을 제공합니다.

원격 관리를 활성화하고 관련 계정을 생성하는 방법에 대한 자세한 내용은 [Snowball Edge에서 Snowball Edge 디바이스 관리 활성화](#)을 참조하십시오.

## 주제

- [를 사용하여 로컬에서 Snowball Edge 디바이스 잠금 해제 AWS OpsHub](#)
- [를 사용하여 원격으로 Snowball Edge 디바이스 잠금 해제 AWS OpsHub](#)

## 를 사용하여 로컬에서 Snowball Edge 디바이스 잠금 해제 AWS OpsHub

로컬에서 디바이스를 연결하고 잠금을 해제하려면

1. 디바이스의 덮개를 열고 전원 코드를 찾아 디바이스를 전원에 연결합니다.
2. 네트워크 케이블(일반적으로 이더넷 RJ45 케이블)을 사용하여 디바이스를 네트워크에 연결하고 전면 패널을 연 다음 디바이스 전원을 켭니다.
3. AWS OpsHub 애플리케이션을 엽니다. 처음 사용하는 사용자인 경우 언어를 선택하라는 메시지가 표시됩니다. 그런 다음 다음을 선택합니다.
4. OpsHub 시작하기 페이지에서 로컬 디바이스에 로그인을 선택한 다음 로그인을 선택합니다.



## Get started with OpsHub

Sign into local devices  
You'll need an unlock code and manifest file

Sign into remote devices  
You'll need an access key & secret key

Sign in

5. 로컬 디바이스에 로그인 페이지에서 Snowball Edge 유형을 선택한 다음 로그인을 선택합니다.
6. 로그인 페이지에서 디바이스 IP 주소와 잠금 해제 코드를 입력합니다. 디바이스 매니페스트를 선택하려면 파일 선택을 선택한 다음 로그인을 선택합니다.



## Sign into your Snowball Edge

Sign in with an unlock code and manifest file


Device IP address

Eg 12.34.45.678

Unlock code

7c0e1-bab84-f7675-0a2b6-bfcc3

Manifest file

 Choose file

No file chosen

Back

Sign in

7. 선택적으로 디바이스의 자격 증명을 프로필로 저장할 수 있습니다. 프로필 이름을 지정하고 프로필 이름 저장을 선택합니다. 프로필에 대한 자세한 내용은 [를 사용하여 프로필 관리 AWS OpsHub](#) 섹션을 참조하십시오.
8. 로컬 디바이스 탭에서 디바이스를 선택하여 디바이스에서 실행 중인 네트워크 인터페이스 및 AWS 서비스와 같은 세부 정보를 확인합니다. 이 탭에서 클러스터에 대한 세부 정보를 보거나 AWS Command Line Interface ()와 마찬가지로 디바이스를 관리할 수도 있습니다AWS CLI. 자세한 내용은 [를 사용하여 Snowball Edge에서 AWS 서비스 관리 AWS OpsHub](#) 단원을 참조하십시오.

가 AWS Snowball Edge Device Management 설치된 디바이스의 경우 원격 관리 활성화를 선택하여 기능을 켤 수 있습니다. 자세한 내용은 [AWS Snowball Edge Device Management 를 사용하여 Snowball Edge 관리](#) 단원을 참조하십시오.

## 를 사용하여 원격으로 Snowball Edge 디바이스 잠금 해제 AWS OpsHub

Snowball Edge를 잠금 해제하려면

원격으로 디바이스를 연결하고 잠금을 해제하려면

1. 디바이스의 덮개를 열고 전원 코드를 찾아 디바이스를 전원에 연결합니다.
2. 이더넷 케이블(일반적으로 RJ45 케이블)을 사용하여 디바이스를 네트워크에 연결하고 전면 패널을 연 다음 디바이스 전원을 켭니다.

### Note

원격으로 잠금을 해제하려면 디바이스가 `device-order-region.amazonaws.com`에 연결할 수 있어야 합니다.

3. AWS OpsHub 애플리케이션을 엽니다. 처음 사용하는 사용자인 경우 언어를 선택하라는 메시지가 표시됩니다. 그런 다음 다음을 선택합니다.
4. OpsHub 시작하기 페이지에서 원격 디바이스에 로그인을 선택한 다음 로그인을 선택합니다.



## Get started with OpsHub

Sign into local devices  
You'll need an unlock code and manifest file

Sign into remote devices  
You'll need an access key & secret key

**Sign in**

5. 원격 디바이스에 로그인 페이지에서 디바이스에 연결된 AWS 계정 의 AWS Identity and Access Management (IAM) 자격 증명(액세스 키 및 비밀 키)을 입력한 다음 로그인을 선택합니다.





## Sign into remote devices

Sign in with an access key and secret key

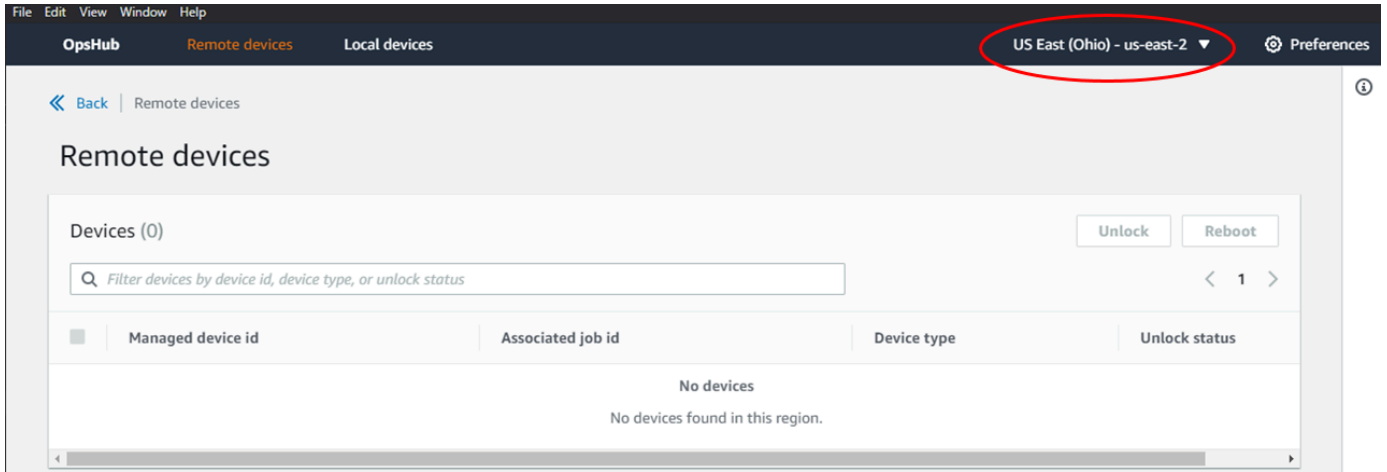
Access key

Secret key

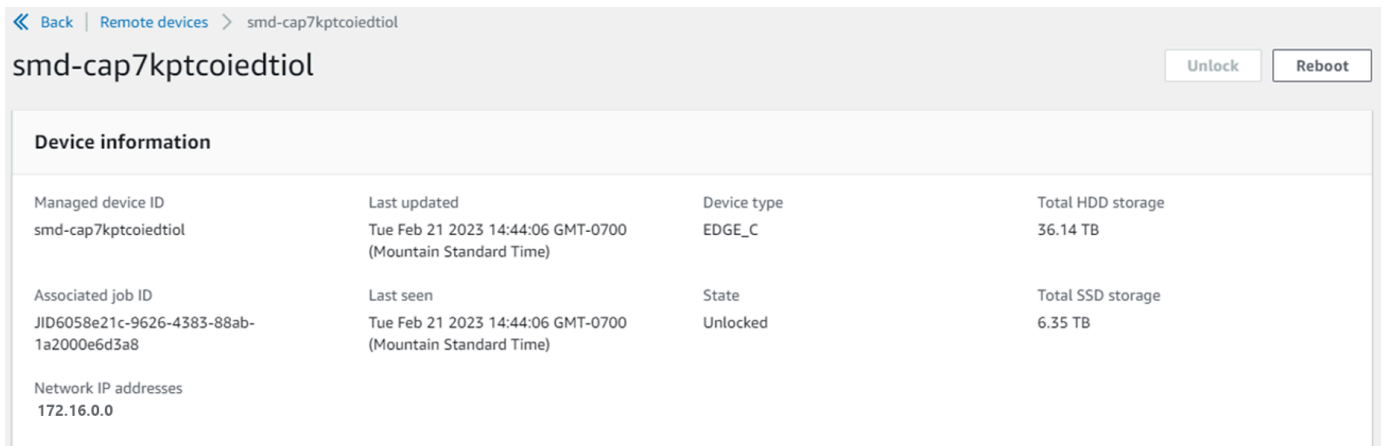
Back

Sign in

- 원격 디바이스 탭 상단에서 원격으로 잠금을 해제할 Snow 디바이스의 리전을 선택합니다.



- 원격 디바이스 탭에서 디바이스를 선택하면 상태 및 네트워크 인터페이스와 같은 세부 정보를 볼 수 있습니다. 그런 다음 잠금 해제를 선택하여 디바이스의 잠금을 해제합니다.



원격 디바이스의 세부 정보 페이지에서 디바이스를 재부팅하고 AWS Command Line Interface ()와 마찬가지로 관리할 수도 있습니다AWS CLI. 다른에서 원격 디바이스를 보려면 탐색 모음에서 현재 리전을 AWS 리전선택한 다음 보려는 리전을 선택합니다. 자세한 내용은 [를 사용하여 Snowball Edge에서 AWS 서비스 관리 AWS OpsHub](#) 단원을 참조하십시오.

## AWS OpsHub 의 PGP 서명 확인(선택 사항)

Linux 운영 체제용 AWS OpsHub 애플리케이션 설치 관리자 패키지는 암호화 방식으로 서명됩니다. 퍼블릭 키를 사용하여 설치 관리자 패키지가 원본이며 수정되지 않았는지 확인할 수 있습니다. 파일에 손상이나 변경이 있을 경우 확인에 실패합니다. GPG(GNU Privacy Guard)를 사용하여 설치 관리자 패키지의 서명을 확인할 수 있습니다. 이 확인은 선택 사항입니다. 애플리케이션의 서명을 확인하기로 선택한 경우 언제든지 수행할 수 있습니다.

[AWS Snowball Edge 리소스](#) 또는 [Snowball Edge 리소스](#)에서 Linux 운영 체제 설치 관리자의 SIGNATURE 파일을 다운로드할 수 있습니다.

Linux 운영 체제용에서 AWS OpsHub 설치 패키지를 확인하려면

1. 다음 퍼블릭 키를 복사해 파일에 저장하고 파일 이름을 지정합니다. 예를 들어 opshub-public-key.pgp입니다.

```
-----BEGIN PGP PUBLIC KEY BLOCK-----
xsFNBF/hGf8BEAC9HCDV8uljDX02Jxspi6kmPu4xqf4ZZLQsSqJcHU61oL/c
/zAN+mUqJT9aJ1rr0QFGVD1bMogecUPf1TW1DkEEpG8ZbX5P8vR+EE10/rW/
WtqizSudy6qy59ZRK+YVSDx7DZyuJmI07j00UADCL+95ZQN9vqwHNjBHsgfQ
l/1Tqhy81ozTZXCi/+u+99YLaugJIP6ZYIeDfpxnghqyVtaappBFTAyfG67Y
N/5mea1VqJzd8liFpIFQn1+X7U2x6emDbM01yJWV3aMmPwhtQ7iBdt5a4x82
EF5bZJ8HSRMvANDILD/9VTN8VfUQGKFjFY2GdX9ERwvfTb47bbv9Z28V1284
4lW2w1B1007Fo02v/Y0ukrN3VHCpmJQ51IiqZbYRa0DVK6UR5QNvUlJ5fwWs
4qW9UDPHt/HDuaMrMFCejEn/7wvRUrGVtZCT9F56A1/dwRSxBejQQEb1AC8j
uuyi7gJaPdyNntR0EFTD7i02L6X2jB4YLfvGxP7Xeq1Y37t8NKF8CYTp0ry/
Wvw0iKZFbo4AkiI0aLyBCK9HBXhUKa9x06g0nhh1UFQrPGrk60RPQKqL76HA
E2ewzGda90w1RBuAt2nRQpyNYjoASBvz/cAr3e0nuWsIzopZIenrxI5ffcjY
f6UWA/OK3ITHtYHewVhseDyEqTQ4MUIWQs4NAwARAQABzTlBV1MgT3BzSHVi
IGZvciBTbm93IEZhbWlseSA8YXdzLW9wc2h1Yi1zaWduZXJAYW1hem9uLmNv
bT7CwY0EEAEIACAFA1/hGf8GCwkHCAMCBBUICgIEFgIBAAIZAQIbAwIeAQAh
CRAhgC9adPNF8RYhBDcvpelIaY930b0vqiGBz1p080XxGbcP+gPZX7LzKc1Y
w9CT3UHgkAIaw0SXYktujzoYVxAz8/j3jEkCY0dKnfyqvWZDiJAXnzmxWwbg
cxg1g0GXNXCM4lAd68CmbA0LoLTaWSQX30ZbswzhbtX2ADAlOpV8RLBik7fm
bS9FYuubDRhfYRQq0fPjUGXFiEgwg6aMFxsrlLlV4QD7t+6ftFIe/mxLbjR4
iMgtr8FIPXbgn05YYY/LeF4NIgX4iLEqRbAnfWjPzqQ1spFWAotIzDmZqby+
WdWThrH4K1rwtYM8sDhqRnMnqJrGFZzk7aDhVPwF+F0VMmPeEN5JRazEeUr1
VZaSw6mu0n4FMGSXuwGgdvmkqnMe6I5/xLdU4IOPNhp0UmakDW0q/a1dREDE
ZLMQDMINphmeQno4inGmwbRo63gitD4ZNR5sWwfuwty251o8Ekv7jkkp3mSv
pdxn5tptttnPaSPcSIX/4ED119Tu0i7aup+v30t7eikYDSZG6g9+jHB3Va9e
/VWShFSgy8Jm2+qq/ujUQDAGTCfSuY9jg1ITsog6ayEza/2upDJ1m+40HK4p
8DrEzP/3jTahT8q5ofFWSRDL17d31TSU+JBmPE3mz311FNXgi08w+taY320z
+irHtb3iSiiukbjS8s0maVgzszRqS9mhaEn4LL0zoqrUicmXgTyFB7n2LuYv
07vxM05xxhGQwsF2BBABCAAJBQJf4RoCAhsDACEJEBFVzT/tDi5FiEEi+09
V+UAYN9Gnw36EVm/NP+00LnnEQ/+J4C0Mn8j0AebXrwBiFs83sQo2q+WHL1S
MRc1g5gRFDXs6h1Gv+TGXRen7j1oeaddWvg0tUBxqmC0jr+8AKH00tiBWSu0
lsS8JU5rindEsKURkTwcG2wyZFoe1z1E8xPkLRSRN5ZbbgKsTz1611HgCCId
Do+WJdDkWGwXmtDvzjM32EI/PVBd108ga9aPwXdhLw0dKAjZ4JrJXLUQJjRI
IVDSyM0bEH0UM6a/+mWNZazNfo0LsGWqGva6Xn5WJwLr1S78vPNf03BQYU0
YRjaVQR+kPtB9aSAZNi5sWfk6NrrNd1Q78d067uhhejsjRt7Mja2fEL4Kb1X
nK4U/ps7X103o/VjblneZ0hJK6kAKU172tnPJTJ31Jb0xX73wsMWDYZRZVcK
```

```

9X9+GFrpwKHKKPjpm0t/FRxNepvqR172TkgBPqGH2TM0FdB1f/uQprvqge
PBbS0JrmBIH9/anIqgtMdtcNQB/0erLdCdQI5af0uD10LcLwdJwG9/bSrfwT
TVEE3WbXmJ8pZgmZlHUiZE6V2DSadV/YItk50I0j jr0VH0Hv1FMwGCEAIFzf
9P/pNi8hpEmLRphRi0VVcdQ30bH0M0gPHu5V9f1IhyCL1zU3LjYTHkq0yJD5
YDA1x01MYq3DcSM5130VBbLmuVS2GpcTCYq1gQA6h/zzMwz+/70wU0EX+EZ
/wEQA0AY8ULmcJIQWIr14V0jy1pJeD3qw7wd+QsBzJ+m0p0B/3ZFAhQiN01
9yCD1HeiZeAmWYX90IXrNiIdcHy+WTAp4G+NaMpqE52qhbDjz+IbvLp11yDH
bYEHpjnTHXEy21bvKAJ0Kkw/2RcQ0i4dodGnq5icyYj+9gcuHvnVwbrQ96Ia
0D7c+b5T+bzFqk90nIcztrMRuhDLJnJpi70jpvQwfq/TkkZA+mzupxfSkq/Y
N9qXNEToT/VI2gn/LS0X4Ar112KxBjzNESQkwGSiWSYtMA5J+Tj5ED0uZ/qe
omNb1A1D4bm7Na8NAoLxCtAiDq/f3To9Xb181Hsnd0mfLCb/BVgP4edQKTii
C/OZHy9QJ1fMn0aq7JVLQAuvQNEL88RKW6YZBqkPd3P6zdc7sWDLTMXM0d3I
e6NUvU7pw0E9NyRfUF+oT4s9wAJhAodinAi8Zi9rEfhK1VCJ76j7bcQyZe0
jXD3IJ7T+X2XA8M/BmypwMw0Soljzhwh044RAasr/fAzpKNPB318JwcQunIz
u2N3CeJ+zrsomjcPxzehwsSVq11zaL2ureJBL0KkBgYxUJYXpbS01ax1TsFG
091dAN0s9Ej8CND37GsNnuygj0gWXbX6MNgbvPs3H3zi/AbMunQ1VB1w07JX
zdM1hBQZh6w+NeiEsK1T6wHi7IhxABEBAAHcWXYEGAEIAAkFA1/hGf8CGwwA
IQkQIYHPWnTzRfEWIQQ3L6XpSGmPd9Gzr6ohgc9adPNF8TMBD/9TbU/+PVbF
ywKvwi3GL01pY7BXn81QaHyunMGUavm080faRR0ynkH0ZqLHCp6bIajF0fvF
b7c0Jamzx8Hg+SIId16yRpRY+fA4RQ6PNnmT93ZgWW3EbjPyJG1m0/rt03SR
+0yn4/1dlg2KfBX4ppMoPCMKUdWxGrimDETXsGihwZ0gmCZqXe81K122PYkSN
JQQ+L1fjKvCaxfPKEjXYTbIbfyyhCR6NzA0VZxCrzSz2xDrYWp/V002K1xda
0ix6r2aEHf+xYEuh0aBt80HY5nXTuRReCVU789MUVtCMqD2u6amdo4BR0kWA
QNg4yavKwV+LVtyYh2Iju9VSyv4xL1Q4xKHvcAUrSH73bHG7b7jkUJckD0f4
twhjJk/Lfwe6RdnVo2WoeTvE93w+NAq2FXmviG7e1t10XfQecvQU3QNbrvH
U8B96W0w8UXJdvTKg4f0NbjSw7iJ3x5naixQ+rA8hLV8x0gn2LX6wvxT/SEu
mn20KX+fPtJELK7v/NheFLX1jsKLXYo4jHrkfIXNsNUhg/x2E71kAjbET3s+
t9kCtxt2iXDDZvpIbmG04QkvLFvoroASmN6+8fupe3e+e2yN0e6xGTuE60gX
I2+X1p1g9IduDYTpO120X1eHyyMqGEEIb4g0iis1oTp5oi3EuAYRGf1XuqAT
VA19bKnpkBsJ0A==
=tD2T
-----END PGP PUBLIC KEY BLOCK-----

```

- GNU Privacy Guard와 같은 암호화 소프트웨어 제품군을 사용하여 퍼블릭 키를 키링으로 가져오고 반환된 키 값을 기록합니다.

```
gpg --import opshub-public-key.gpg
```

### Example 명령의 출력

```
gpg: key 1655BBDE2B770256: public key "AWS OpsHub for Snowball Edge <aws-opshub-
signer@amazon.com>" imported
```

```
gpg: Total number processed: 1
gpg:             imported: 1
```

3. 지문을 확인합니다. *key-value*을 이전 단계의 값으로 바꿔야 합니다. GPG를 사용하여 지문을 확인하는 것이 좋습니다.

```
gpg --fingerprint key-value
```

다음과 비슷한 출력이 반환됩니다.

```
pub  rsa4096 2020-12-21 [SC]
    372F A5E9 4869 8F77 D1B3  AFAA 2181 CF5A 74F3 45F1
uid  [ unknown] AWS OpsHub for Snowball Edge <aws-opshub-
    signer@amazon.com>
sub  rsa4096 2020-12-21 [E]
```

지문은 다음과 일치해야 합니다.

```
372F A5E9 4869 8F77 D1B3  AFAA 2181 CF5A 74F3 45F1
```

지문이 일치하지 않으면 AWS OpsHub 애플리케이션을 설치하지 마십시오. 지원에 문의하십시오.

4. 설치 관리자 패키지를 확인하고, 아직 다운로드하지 않은 경우 인스턴스의 아키텍처 및 운영 체제에 따라 SIGNATURE 파일을 다운로드합니다.
5. 설치 관리자 패키지 서명을 확인합니다. 반드시 *signature-filename* 및 *OpsHub-download-filename*를 SIGNATURE 파일 및 AWS OpsHub 애플리케이션을 다운로드할 때 지정한 값으로 바꿉니다.

GPG

```
gpg --verify signature-filename OpsHub-download-filename
```

다음과 비슷한 출력이 반환됩니다.

GPG

```
gpg: Signature made Mon Dec 21 13:44:47 2020 PST
gpg:             using RSA key 1655BBDE2B770256
gpg: Good signature from "AWS OpsHub for Snowball Edge <aws-opshub-
    signer@amazon.com>" [unknown]
```

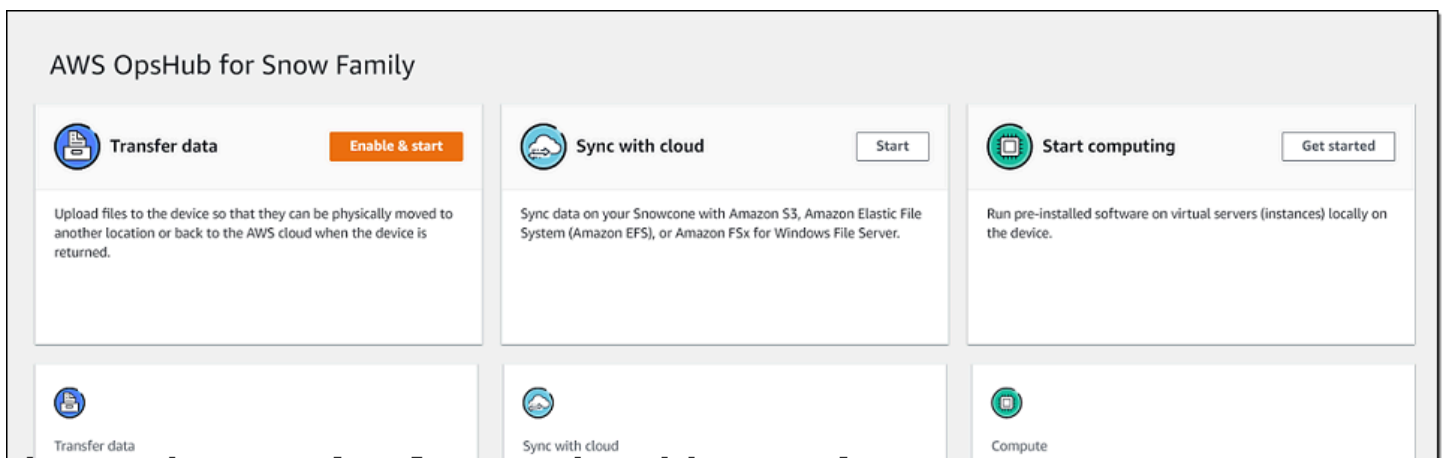
```
gpg: WARNING: This key is not certified with a trusted signature!
gpg:          There is no indication that the signature belongs to the owner.
Primary key fingerprint: 9C93 4C3B 61F8 C434 9F94 5CA0 1655 BBDE 2B77 0256
```

GPG 사용 시, 출력에 BAD signature 문구가 포함된 경우 절차를 올바르게 수행했는지 확인합니다. 이 응답이 계속 표시되면 문의 지원 하고 에이전트를 설치하지 마십시오. 신뢰에 대한 경고 메시지는 서명이 유효하지 않다는 의미가 아니라 퍼블릭 키를 확인하지 않았다는 의미입니다. 사용자 또는 사용자가 신뢰하는 사람이 서명한 키만 신뢰됩니다.

## 를 사용하여 Snowball Edge에서 AWS 서비스 관리 AWS OpsHub

를 사용하면 Snowball Edge에서 AWS 서비스를 사용하고 관리할 AWS OpsHub 수 있습니다. 현재는 다음 리소스를 AWS OpsHub 지원합니다.

- Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스 - Amazon EC2 호환 인스턴스를 사용하면 처리를 위해 AWS 클라우드 에 소프트웨어를 보내지 않고도 가상 서버에 설치된 소프트웨어를 실행할 수 있습니다.
- NFS(Network File System) - 파일 공유를 사용하여 데이터를 디바이스로 이동합니다. 디바이스를 로 배송 AWS 하여 로 데이터를 전송 AWS 클라우드하거나 DataSync를 사용하여 다른 AWS 클라우드 위치로 전송할 수 있습니다.
- Snowball Edge의 Amazon S3 호환 스토리지 - 복원력 향상, 확장 및 견고한 모바일 엣지 및 연결 이 끊긴 환경으로 확장된 Amazon S3 API 기능 세트를 통해 안전한 객체 스토리지를 제공합니다. Snowball Edge에서 Amazon S3 호환 스토리지를 사용하면 데이터를 저장하고 엣지 컴퓨팅을 위해 Snowball Edge에서 고가용성 애플리케이션을 실행할 수 있습니다.



## 주제

- [를 사용하여 Snowball Edge에서 Amazon EC2-compatible 인스턴스 시작 AWS OpsHub](#)
- [를 사용하여 Snowball Edge에서 Amazon EC2-compatible 인스턴스 중지 AWS OpsHub](#)
- [를 사용하여 Snowball Edge에서 Amazon EC2-compatible 인스턴스 시작 AWS OpsHub](#)
- [에서 EC2-compatible 인스턴스의 키 페어 작업 AWS OpsHub](#)
- [를 사용하여 Amazon EC2-compatible 인스턴스 종료 AWS OpsHub](#)
- [를 사용하여 Snowball Edge에서 로컬로 스토리지 볼륨 사용 AWS OpsHub](#)
- [를 사용하여 Amazon EC2-compatible AMI로 이미지 가져오기 AWS OpsHub](#)
- [를 사용하여 Snowball Edge에서 스냅샷 삭제 AWS OpsHub](#)
- [를 사용하여 Snowball Edge에서 AMI 등록 취소 AWS OpsHub](#)
- [를 사용하여 Snowball Edge에서 Amazon EC2 클러스터 관리 AWS OpsHub](#)
- [를 사용하여 Snowball Edge에서 Amazon S3 호환 스토리지 설정 AWS OpsHub](#)
- [를 사용하여 Amazon S3 어댑터 스토리지 관리 AWS OpsHub](#)
- [를 사용하여 NFS 인터페이스 관리 AWS OpsHub](#)

## 를 사용하여 Snowball Edge에서 Amazon EC2-compatible 인스턴스 시작 AWS OpsHub

다음 단계에 따라를 사용하여 Amazon EC2-compatible 인스턴스를 시작합니다 AWS OpsHub.

Amazon EC2 호환 인스턴스를 시작하려면

1. AWS OpsHub 애플리케이션을 엽니다.
2. 대시보드의 Start computing(컴퓨팅 시작) 섹션에서 Get started(시작하기)를 선택합니다. 또는 상단의 Services(서비스) 메뉴를 선택한 다음 Compute(EC2)(컴퓨팅(EC2))를 선택하여 Compute(컴퓨팅) 페이지를 엽니다. 모든 컴퓨팅 리소스가 Resources(리소스) 섹션에 표시됩니다.
3. 디바이스에서 Amazon EC2 호환 인스턴스를 실행 중인 경우 인스턴스 아래의 인스턴스 이름 옆에 해당 인스턴스가 표시됩니다. 이 페이지에서 각 인스턴스의 세부 정보를 볼 수 있습니다.
4. 인스턴스 시작을 선택합니다. 인스턴스 시작 마법사가 열립니다.
5. 디바이스에서 Amazon EC2 호환 인스턴스를 시작할 Snow 디바이스를 선택합니다.

## Launch instance ✕

Device

192.0.2.0 ▼

Image (AMI)

snow-al2-test-ami-1.0.2 ▼

Instance type

sbe-c.small ▼

Create public IP address (VNI)  Use existing IP address (VNI)  Do not attach IP address

Physical network interface

SFP+:a.bc-1d2ef456gg678gi9j ▼

IP Address assignment

DHCP ▼

Key pair

Create key pair  Use existing key pair  Do not attach key pair

Name

test-instance-key-pair

The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

Create key pair

Cancel **Launch**



6. Image(AMI)(이미지(AMI))에서 Amazon 머신 이미지(AMI)를 목록에서 선택합니다. 이 AMI가 인스턴스를 시작하는 데 사용됩니다.
7. Instance type(인스턴스 유형)에서 인스턴스 유형을 목록에서 선택합니다.
8. 인스턴스에 IP 주소를 연결하는 방법을 선택합니다. 다음과 같은 옵션이 있습니다:
  - 퍼블릭 IP 주소 생성(VNI) - 물리적 네트워크 인터페이스를 사용하여 새 IP 주소를 생성하려면 이 옵션을 선택합니다. 물리적 네트워크 인터페이스와 IP 주소 할당을 선택합니다.
  - 기존 IP 주소 사용(VNI) - 기존 IP 주소를 사용하고 기존 가상 네트워크 인터페이스를 사용하려면 이 옵션을 선택합니다. 물리적 네트워크 인터페이스와 가상 네트워크 인터페이스를 선택합니다.
  - IP 주소 연결 안 함 - IP 주소를 연결하지 않으려면 이 옵션을 선택합니다.
9. 인스턴스에 키 페어를 연결하는 방법을 선택합니다. 다음과 같은 옵션이 있습니다:
 

키 페어 생성 - 새 키 페어를 생성하고 이 키 페어로 새 인스턴스를 시작하려면 이 옵션을 선택합니다.

기존 키 페어 사용 - 기존 키 페어를 사용하여 인스턴스를 시작하려면 이 옵션을 선택합니다.

IP 주소 연결 안 함 - 키 페어를 연결하지 않으려면 이 옵션을 선택합니다. 이 AMI에 내장된 암호를 이미 알고 있지 않다면 이 인스턴스에 연결할 수 없다는 점을 인정해야 합니다.

자세한 내용은 [에서 EC2-compatible 인스턴스의 키 페어 작업 AWS OpsHub](#) 섹션을 참조하세요.
10. 시작을 선택합니다. Compute instances(컴퓨팅 인스턴스) 섹션에서 인스턴스가 시작되는 것을 볼 수 있습니다. State(상태)가 Pending(대기 중)으로 나타나며 완료되면 Running(실행 중)으로 변경됩니다.

## 를 사용하여 Snowball Edge에서 Amazon EC2-compatible 인스턴스 중지 AWS OpsHub

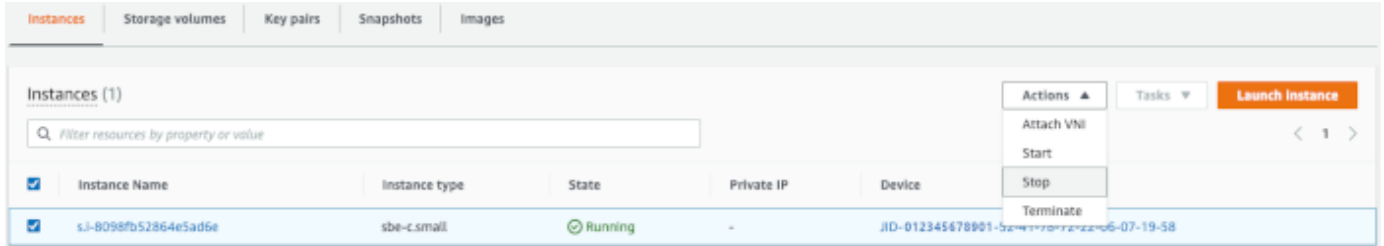
다음 단계에 따라를 사용하여 Amazon EC2-compatible 인스턴스를 중지 AWS OpsHub 합니다.

Amazon EC2 호환 인스턴스를 중지하려면

1. AWS OpsHub 애플리케이션을 엽니다.
2. 대시보드의 Start computing(컴퓨팅 시작) 섹션에서 Get started(시작하기)를 선택합니다. 또는 상단의 Services(서비스) 메뉴를 선택한 다음 Compute(EC2)(컴퓨팅(EC2))를 선택하여 Compute(컴퓨팅) 페이지를 엽니다.

모든 컴퓨팅 리소스가 Resources(리소스) 섹션에 표시됩니다.

3. 디바이스에서 Amazon EC2 호환 인스턴스를 실행 중인 경우 인스턴스 아래의 인스턴스 이름 옆에 해당 인스턴스가 표시됩니다.
4. 중지할 인스턴스를 선택하고 작업 메뉴를 선택한 다음 중지를 선택합니다. State(상태)가 Stopping(중지 중)으로 변경되고 완료되면 Stopped(중지됨)로 변경됩니다.



## 를 사용하여 Snowball Edge에서 Amazon EC2-compatible 인스턴스 시작 AWS OpsHub

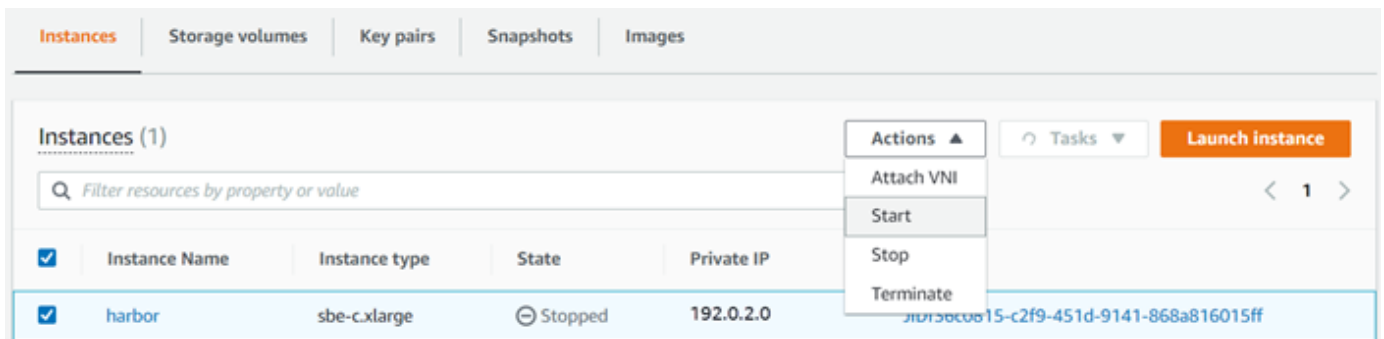
다음 단계에 따라를 사용하여 Amazon EC2-compatible 인스턴스를 시작합니다 AWS OpsHub.

Amazon EC2 호환 인스턴스를 시작하려면

1. AWS OpsHub 애플리케이션을 엽니다.
2. 대시보드의 Start computing(컴퓨팅 시작) 섹션에서 Get started(시작하기)를 선택합니다. 또는 상단의 Services(서비스) 메뉴를 선택한 다음 Compute(EC2)(컴퓨팅(EC2))를 선택하여 Compute(컴퓨팅) 페이지를 엽니다.

컴퓨팅 리소스가 Resources(리소스) 섹션에 표시됩니다.

3. 인스턴스 이름 옆의 인스턴스에서 시작하려는 인스턴스를 찾습니다.
4. 인스턴스를 선택한 후 Start(시작)를 선택합니다. State(상태)가 Pending(대기 중)으로 변경되며 완료되면 Running(실행 중)으로 변경됩니다.



## 에서 EC2-compatible 인스턴스의 키 페어 작업 AWS OpsHub

Amazon EC2 호환 인스턴스를 시작하고 SSH를 사용하여 연결하려는 경우 키 페어를 제공해야 합니다. Amazon EC2를 사용하여 새 키 페어를 생성하거나 기존 키 페어를 가져오거나 키 페어를 관리할 수 있습니다.

키 페어를 생성하거나 가져오거나 관리하려면

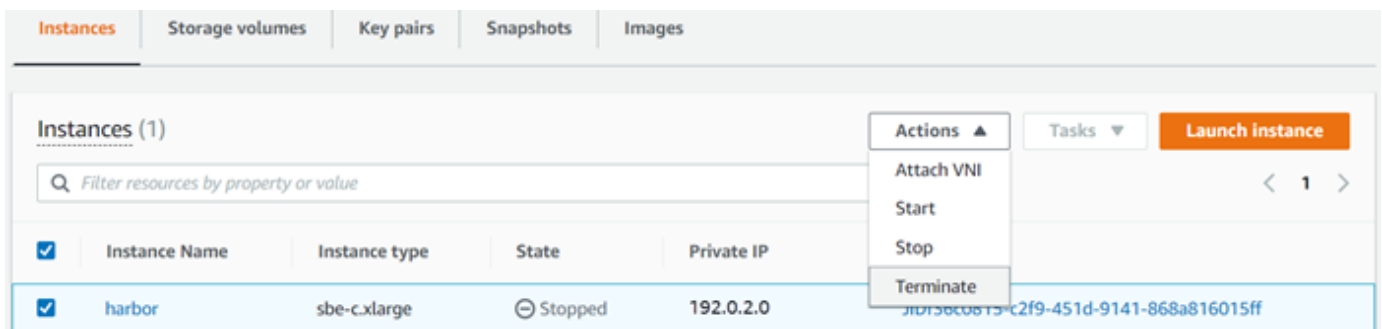
1. AWS OpsHub 대시보드에서 컴퓨팅을 엽니다.
2. 탐색 창에서 컴퓨팅(EC2) 페이지를 선택한 다음 키 페어 탭을 선택합니다. 키 페어를 생성하거나 가져오거나 관리할 수 있는 Amazon EC2 콘솔로 리디렉션됩니다.
3. 키 페어를 생성하거나 가져오는 방법에 대한 자세한 내용은 Amazon EC2 사용 설명서의 [Amazon EC2 키 페어 및 Linux 인스턴스](#)를 참조하십시오.

## 를 사용하여 Amazon EC2-compatible 인스턴스 종료 AWS OpsHub

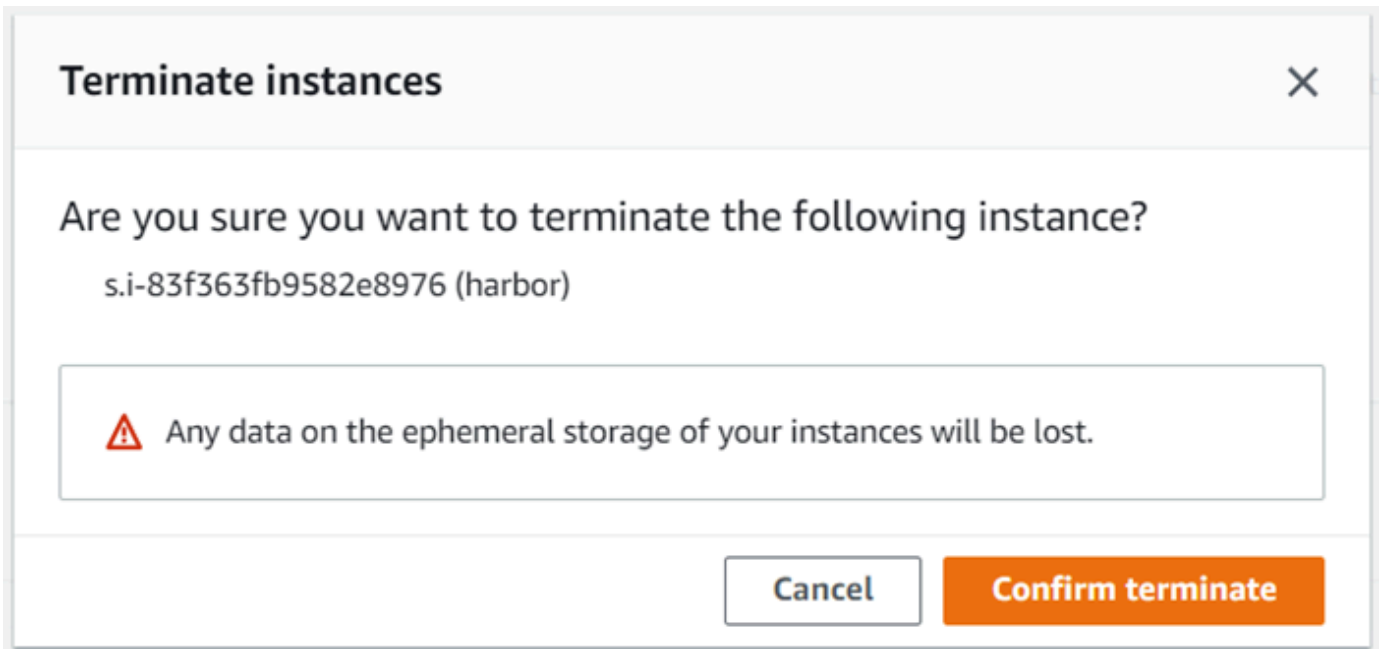
Amazon EC2 호환 인스턴스를 종료한 후에는 인스턴스를 다시 시작할 수 없습니다.

Amazon EC2 호환 인스턴스를 종료하려면

1. AWS OpsHub 애플리케이션을 엽니다.
2. 대시보드의 Start computing(컴퓨팅 시작) 섹션에서 Get started(시작하기)를 선택합니다. 또는 상단의 Services(서비스) 메뉴를 선택한 다음 Compute(EC2)(컴퓨팅(EC2))를 선택하여 Compute(컴퓨팅) 페이지를 엽니다. Resources(리소스) 섹션에서 모든 컴퓨팅 리소스를 볼 수 있습니다.
3. 인스턴스 이름 옆의 인스턴스에서 종료하려는 인스턴스를 찾습니다.
4. 인스턴스를 선택하고 작업 메뉴를 선택합니다. 작업 메뉴에서 종료를 선택합니다.



5. 인스턴스 종료 창에서 종료 확인을 선택합니다.



#### Note

인스턴스가 종료된 후에는 다시 시작할 수 없습니다.

State(상태)가 Terminating(종료 중)으로 변경되고 완료되면 Terminated(종료됨)으로 변경됩니다.

## 를 사용하여 Snowball Edge에서 로컬로 스토리지 볼륨 사용 AWS OpsHub

Amazon EC2 호환 인스턴스는 Amazon EBS 볼륨을 스토리지로 사용합니다. 이 절차에서는를 사용하여 스토리지 볼륨을 생성하고 인스턴스에 연결합니다 AWS OpsHub.

스토리지 볼륨을 생성하려면

1. AWS OpsHub 애플리케이션을 엽니다.
2. 대시보드의 Start computing(컴퓨팅 시작) 섹션에서 Get started(시작하기)를 선택합니다. 또는 상단의 Services(서비스) 메뉴를 선택한 다음 Compute(EC2)(컴퓨팅(EC2))를 선택하여 Compute(컴퓨팅) 페이지를 엽니다.
3. Storage volumes(스토리지 볼륨) 탭을 선택합니다. 디바이스에 스토리지 볼륨이 있는 경우 볼륨에 대한 세부 정보가 Storage volumes(스토리지 볼륨) 아래에 표시됩니다.
4. Create volume(볼륨 생성)을 선택하여 Create volume(볼륨 생성) 페이지를 엽니다.

<< Back | Compute (EC2) > Create volume  
**Create Volume**  
 Device  
 Select the device on which you wish to create the volume.  
 JID5a11d1db-8b98-4f37-80bf-97af46e45eb2 - 10.24.34.0  
 Size  
 Define the size of the volume, in GiBs.  
 100  
 Volume Type  
 Select a performance type for your volume.  
 Capacity-optimized HDD volume (sbg1)  
 Cancel Submit

5. 볼륨을 생성할 디바이스를 선택하고 생성할 크기(GiB)를 입력한 다음 볼륨 유형을 선택합니다.
6. 제출을 선택합니다. State(상태)가 Creating(생성 중)으로 나타나며 완료되면 Available(사용 가능)로 변경됩니다. Volumes(볼륨) 탭에서 볼륨 및 세부 정보를 볼 수 있습니다.

스토리지 볼륨을 인스턴스에 연결하려면

1. 생성한 볼륨을 선택한 다음 Attach volume(볼륨 연결)을 선택합니다.

**Attach volume: s.vol-0a12b34def5678901** ×  
 Compute Instance Id  
 The Compute instance you wish to attach the volume to.  
 s.i-0123a456b789c012d  
 Volume Device Name  
 The device name of the volume.  
 /dev/sdh  
 Cancel Attach

2. 컴퓨팅 인스턴스 ID에서 볼륨을 연결할 인스턴스를 선택합니다.
3. Volume Device Name(볼륨 디바이스 이름)에 볼륨의 디바이스 이름(예: **/dev/sdh** 또는 **xvdh**)을 입력합니다.
4. 연결을 선택합니다.

볼륨이 더 이상 필요하지 않으면 인스턴스에서 볼륨을 분리한 다음 삭제할 수 있습니다.

## 를 사용하여 Amazon EC2-compatible AMI로 이미지 가져오기 AWS OpsHub

이미지 스냅샷을 Snowball Edge 디바이스로 가져와서 Amazon EC2 호환 Amazon Machine Image(AMI)로 등록할 수 있습니다. 스냅샷은 기본적으로 AMI 또는 다른 스토리지 볼륨을 생성하는 데 사용할 수 있는 스토리지 볼륨의 복사본입니다. 이렇게 하면 외부 소스의 자체 이미지를 디바이스로 가져와 Amazon EC2 호환 인스턴스로 시작할 수 있습니다.

다음 단계에 따라 이미지 가져오기를 완료합니다.

1. 디바이스의 Amazon S3 버킷에 스냅샷을 업로드합니다.
2. 스냅샷을 가져오고 내보내는 데 사용되는 기능인 Amazon S3, Amazon EC2 및 VM 가져오기/내보내기에 대한 액세스를 허용하는 데 필요한 권한을 설정합니다.
3. S3 버킷의 스냅샷을 디바이스에 이미지로 가져옵니다.
4. 이미지를 Amazon EC2 호환 AMI로 등록합니다.
5. AMI를 Amazon EC2 호환 인스턴스로 시작합니다.

### Note

Snowball Edge에 스냅샷을 업로드할 때 다음 제한 사항에 유의하세요.

- Snowball Edge는 현재 RAW 이미지 형식의 스냅샷 가져오기만 지원합니다.
- Snowball Edge는 현재 크기가 1GB~1TB인 스냅샷 가져오기만 지원합니다.

## 1단계: 스냅샷을 디바이스의 S3 버킷에 업로드

스냅샷을 가져오기 전에 디바이스의 Amazon S3에 스냅샷을 업로드해야 합니다. 디바이스 또는 클러스터에서 사용할 수 있는 Amazon S3에서만 스냅샷을 가져올 수 있기 때문입니다. 가져오기 프로세스 중에 이미지를 저장할 디바이스의 S3 버킷을 선택합니다.

스냅샷을 Amazon S3에 업로드

- S3 버킷을 생성하려면 [Amazon S3 스토리지 생성](#)을 참조하십시오.

S3 버킷에 스냅샷을 업로드하려면 [Amazon S3 스토리지에 파일 업로드](#)를 참조하십시오.

## 2단계: S3 버킷에서 스냅샷 가져오기

스냅샷이 Amazon S3에 업로드되면 디바이스로 가져올 수 있습니다. 가져왔거나 가져오는 과정에 있는 모든 스냅샷은 스냅샷 탭에 표시됩니다.

스냅샷을 디바이스로 가져오려면

1. AWS OpsHub 애플리케이션을 엽니다.
2. 대시보드의 Start computing(컴퓨팅 시작) 섹션에서 Get started(시작하기)를 선택합니다. 또는 상단의 Services(서비스) 메뉴를 선택한 다음 Compute(EC2)(컴퓨팅(EC2))를 선택하여 Compute(컴퓨팅) 페이지를 엽니다. 모든 컴퓨팅 리소스가 Resources(리소스) 섹션에 표시됩니다.
3. 디바이스로 가져온 모든 스냅샷을 보려면 스냅샷 탭을 선택합니다. Amazon S3의 이미지 파일은 디바이스에 스냅샷으로 가져오는 .raw 파일입니다. 스냅샷 ID 또는 스냅샷 상태를 기준으로 필터링하여 특정 스냅샷을 찾을 수 있습니다. 스냅샷 ID를 선택하여 해당 스냅샷의 세부 정보를 볼 수 있습니다.
4. 가져오려는 스냅샷을 선택하고 스냅샷 가져오기를 선택하여 스냅샷 가져오기 페이지를 엽니다.
5. 디바이스에서 가져오려는 Snow Family 디바이스의 IP 주소를 선택합니다.
6. 설명 가져오기 및 스냅샷 설명에 각각에 대한 설명을 입력합니다.
7. 역할 목록에서 가져오기에 사용할 역할을 선택합니다. Snowball Edge는 VM Import/Export를 사용하여 스냅샷을 가져옵니다. 이 역할을 수 AWS 임하고 이를 사용하여 사용자를 대신하여 스냅샷을 가져옵니다. 이 역할이 구성되어 있지 않은 경우 AWS Snowball Edge(로컬 IAM 역할을 생성할 수 AWS OpsHub 있는의 IAM을 엽니다 AWS Identity and Access Management . 또한 역할에는 가져오기를 수행하는 데 필요한 VM 가져오기/내보내기 권한이 있는 정책이 필요합니다. 역할에 이 정책을 연결해야 합니다. 이에 대한 자세한 내용은 [로컬에서 IAM 사용](#) 참조하십시오.

다음은 정책의 예입니다.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "vmie.amazonaws.com"
      },
    },
  ],
}
```

```

    "Action": "sts:AssumeRole"
  }
]
}

```

에 로그인 AWS Management Console 하고 <https://console.aws.amazon.com/iam/> IAM 콘솔을 엽니다.

생성하는 역할에는 Amazon S3에 액세스할 수 있는 최소 권한이 있어야 합니다. 다음은 최소 정책의 예입니다.

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetBucketLocation",
        "s3:GetObject",
        "s3:ListBucket",
        "s3:GetMetadata"
      ],
      "Resource": [
        "arn:aws:s3:::import-snapshot-bucket-name",
        "arn:aws:s3:::import-snapshot-bucket-name/*"
      ]
    }
  ]
}

```

8. S3 탐색을 선택하고 가져오려는 스냅샷이 포함된 S3 버킷을 선택합니다. 스냅샷을 선택하고 제출을 선택합니다. 스냅샷이 디바이스에 다운로드되기 시작합니다. 스냅샷 ID를 선택하여 세부 정보를 볼 수 있습니다. 이 페이지에서 가져오기 프로세스를 취소할 수 있습니다.

### 3단계: 스냅샷을 Amazon EC2 호환 AMI로 등록

스냅샷으로 가져온 이미지에서 Amazon EC2 호환 AMI를 생성하는 프로세스를 등록이라고 합니다. 디바이스로 가져온 이미지를 Amazon EC2 호환 인스턴스로 시작하려면 먼저 등록해야 합니다.



## 스냅샷으로 가져온 이미지를 등록하려면

1. AWS OpsHub 애플리케이션을 엽니다.
2. 대시보드의 Start computing(컴퓨팅 시작) 섹션에서 Get started(시작하기)를 선택합니다. 또는 상단의 Services(서비스) 메뉴를 선택한 다음 Compute(EC2)(컴퓨팅(EC2))를 선택하여 Compute(컴퓨팅) 페이지를 엽니다. 모든 컴퓨팅 리소스가 Resources(리소스) 섹션에 표시됩니다.
3. 이미지(Images) 탭을 선택합니다. 이미지를 이름, ID 또는 상태별로 필터링하여 특정 이미지를 찾을 수 있습니다.
4. 등록하려는 이미지를 선택하고 이미지 등록을 선택합니다.

The screenshot shows the AWS OpsHub console interface for managing images. At the top, there are buttons for 'Deregister image' and 'Register image'. Below is a search bar with the placeholder text 'Filter resources by property or value'. The main content is a table with columns: Name, Id, State, and Description. The 'testing' image is selected, indicated by a blue checkmark in the first column.

	Name	Id	State	Description
<input type="checkbox"/>	test-image	s.ami-8087a170e24a3124d	Available	test-image-oopshub-1.15.6
<input type="checkbox"/>	SnowAL2	s.ami-88833957b18097f46	Available	-
<input type="checkbox"/>	test for 1.15.2	s.ami-8c4d8b29618568414	Available	-
<input checked="" type="checkbox"/>	testing	s.ami-8de47d2e397937318	Available	tesing
<input type="checkbox"/>	test-windows	s.ami-8f12682f18b1de4e2	Available	windows-test

5. 이미지 등록 페이지에서 이름 및 설명을 입력합니다.
6. 루트 볼륨의 경우 루트 디바이스의 이름을 지정합니다.

블록 디바이스 섹션에서 볼륨 크기 및 볼륨 유형을 변경할 수 있습니다.

7. 인스턴스 종료 시 EBS 볼륨이 삭제되기를 원하는 경우 종료 시 삭제를 선택합니다.
8. 볼륨을 더 추가하고자 하는 경우 새 볼륨 추가를 선택합니다.
9. 준비가 되면 제출을 선택합니다.

## 4단계: Amazon EC2 호환 AMI 시작

- 자세한 내용은 [Amazon EC2 호환 인스턴스 시작하기](#)를 참조하십시오.

## 를 사용하여 Snowball Edge에서 스냅샷 삭제 AWS OpsHub

스냅샷이 더 이상 필요하지 않으면 삭제할 수 있습니다. Amazon S3의 이미지 파일은 디바이스에 스냅샷으로 가져오는 .raw 파일입니다. 삭제하려는 스냅샷을 이미지에서 사용하는 경우 해당 스냅샷을 삭제할 수 없습니다. 가져오기가 완료된 후 디바이스의 Amazon S3에 업로드한 .raw 파일을 삭제할 수도 있습니다.

### 스냅샷 삭제

1. AWS OpsHub 애플리케이션을 엽니다.
2. 대시보드의 Start computing(컴퓨팅 시작) 섹션에서 Get started(시작하기)를 선택합니다. 또는 상단의 Services(서비스) 메뉴를 선택한 다음 Compute(EC2)(컴퓨팅(EC2))를 선택하여 Compute(컴퓨팅) 페이지를 엽니다. 모든 컴퓨팅 리소스가 Resources(리소스) 섹션에 표시됩니다.
3. 스냅샷 탭을 선택하면 가져온 모든 스냅샷을 볼 수 있습니다. 스냅샷 ID 또는 스냅샷 상태를 기준으로 필터링하여 특정 스냅샷을 찾을 수 있습니다.
4. 삭제할 스냅샷을 선택하고 삭제를 선택합니다. 여러 스냅샷을 선택할 수 있습니다.

Snapshot ID	State	Capacity	Description	Start Time
s.snap-01a2b3c4567d8e9f0g	Completed	8 GBs	Snow marketplace AL2 AMI 1.0.2	Thu, 09 Jun 2022 16:27:29 GMT

5. 스냅샷 삭제 확인 대화 상자에서 스냅샷 삭제를 선택합니다. 삭제에 성공하면 스냅샷 탭의 목록에서 스냅샷이 제거됩니다.

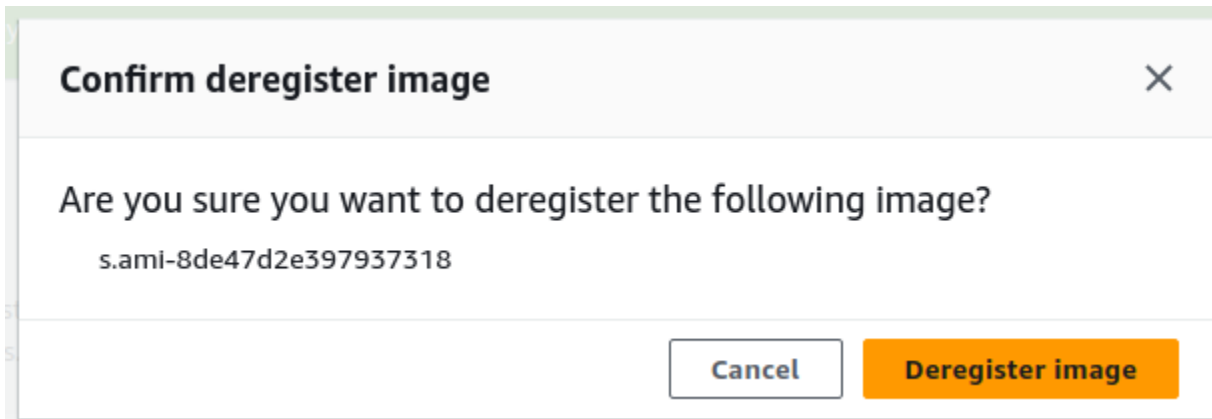
## 를 사용하여 Snowball Edge에서 AMI 등록 취소 AWS OpsHub

### AMI 등록 해제

1. AWS OpsHub 애플리케이션을 엽니다.
2. 대시보드의 Start computing(컴퓨팅 시작) 섹션에서 Get started(시작하기)를 선택합니다. 또는 상단의 Services(서비스) 메뉴를 선택한 다음 Compute(EC2)(컴퓨팅(EC2))를 선택하여 Compute(컴퓨팅) 페이지를 엽니다. 모든 컴퓨팅 리소스가 Resources(리소스) 섹션에 표시됩니다.
3. 이미지(Images) 탭을 선택합니다. 모든 이미지가 나열됩니다. 이미지를 이름, ID 또는 상태별로 필터링하여 특정 이미지를 찾을 수 있습니다.
4. 등록 취소하려는 이미지를 선택하고 등록 취소를 선택합니다.

	Name	Id	State	Description
<input type="checkbox"/>	test-image	s.ami-8087a170e24a3124d	Available	test-image-oopshub-1.15.6
<input type="checkbox"/>	SnowAL2	s.ami-88833957b18097f46	Available	-
<input type="checkbox"/>	test for 1.15.2	s.ami-8c4d8b29618568414	Available	-
<input checked="" type="checkbox"/>	testing	s.ami-8de47d2e397937318	Available	tesing
<input type="checkbox"/>	test-windows	s.ami-8f12682f18b1de4e2	Available	windows-test

5. 이미지 등록 취소 확인 창에서 이미지 ID를 확인하고 이미지 등록 취소를 선택합니다. 등록 취소가 성공하면 이미지가 이미지 목록에서 제거됩니다.



## 를 사용하여 Snowball Edge에서 Amazon EC2 클러스터 관리 AWS OpsHub

Amazon EC2 클러스터는 디바이스 클러스터로 함께 프로비저닝되는 디바이스 그룹입니다. 클러스터를 사용하려면 디바이스의 AWS 서비스가 기본 엔드포인트에서 실행 중이어야 합니다. 또한 통신할 특정 디바이스를 클러스터에서 선택해야 합니다. 디바이스별로 클러스터를 사용합니다.

Amazon EC2 클러스터를 생성하려면

1. Snow 디바이스에 연결하고 로그인합니다. 디바이스에 로그인하는 방법에 대한 자세한 내용은 [를 사용하여 Snowball Edge 디바이스 잠금 해제 AWS OpsHub](#) 섹션을 참조하십시오.
2. 디바이스 선택 페이지에서 Snowball Edge 클러스터를 선택하고 다음을 선택합니다.
3. Connect to your device(디바이스에 연결) 페이지에서 디바이스의 IP 주소와 클러스터에 있는 다른 디바이스의 IP 주소를 입력합니다.

4. Add another device(다른 디바이스 추가)를 선택하여 디바이스를 추가하고 Next(다음)를 선택합니다.
5. Provide the keys(키 제공) 페이지에서 디바이스 클라이언트 잠금 해제 코드를 입력하고 Unlock device(디바이스 잠금 해제)를 선택합니다.

Snowball Edge 디바이스는 256비트 암호화를 사용하여 데이터의 보안과 완전한 연계보관성(chain-of-custody)을 보장합니다.

6. 선택적으로 이름을 입력하여 프로필을 생성한 다음 프로필 이름 저장을 선택할 수 있습니다. 모든 클러스터를 볼 수 있는 대시보드가 표시됩니다.

이제 AWS 서비스 사용 및 클러스터 관리를 시작할 수 있습니다. 개별 인스턴스를 관리하는 것과 동일한 방식으로 클러스터의 인스턴스를 관리합니다. 지침은 [를 사용하여 Snowball Edge에서 AWS 서비스 관리 AWS OpsHub](#) 섹션을 참조하세요.

## 를 사용하여 Snowball Edge에서 Amazon S3 호환 스토리지 설정 AWS OpsHub

Amazon S3 compatible storage on Snowball Edge 서비스는 기본적으로 활성화되어 있지 않습니다. 디바이스 또는 클러스터에서 서비스를 시작하려면 각 디바이스에 가상 네트워크 인터페이스(vNIC) 2개를 생성하여 s3control 및 s3api 엔드포인트에 연결해야 합니다.

### 주제

- [에 대한 Snowball Edge의 Amazon S3 호환 스토리지 사전 조건 AWS OpsHub](#)
- [에서 Amazon S3 compatible storage on Snowball Edge 단순 설정 옵션 사용 AWS OpsHub](#)
- [를 사용하여 Amazon S3 compatible storage on Snowball Edge 고급 설정 옵션 사용 AWS OpsHub](#)
- [를 사용하여 자동으로 시작하도록 Snowball Edge에서 Amazon S3 호환 스토리지 구성 AWS OpsHub](#)
- [를 사용하여 Snowball Edge의 Amazon S3 호환 스토리지에서 버킷 생성 AWS OpsHub](#)
- [를 사용하여 Snowball Edge 버킷의 Amazon S3 호환 스토리지에 파일 및 폴더 업로드 AWS OpsHub](#)
- [를 AWS OpsHub 사용하여 Snowball Edge 버킷의 Amazon S3 호환 스토리지에서 파일 및 폴더 제거 AWS OpsHub](#)
- [Snowball Edge의 Amazon S3 호환 스토리지에서 버킷 삭제](#)

## 에 대한 Snowball Edge의 Amazon S3 호환 스토리지 사전 조건 AWS OpsHub

를 사용하여 디바이스 또는 클러스터를 설정하려면 먼저 다음을 AWS OpsHub수행합니다.

- Snowball Edge 디바이스의 전원을 켜고 네트워크에 연결합니다.
- 로컬 컴퓨터에서 [AWS OpsHub](#)의 최신 버전을 다운로드하고 설치합니다. 디바이스 또는 클러스터에 연결하여 매니페스트 파일로 잠금을 해제합니다. 자세한 내용은 [디바이스 잠금 해제](#) 섹션을 참조하십시오.

## 에서 Amazon S3 compatible storage on Snowball Edge 단순 설정 옵션 사용 AWS OpsHub

네트워크에서 DHCP를 사용하는 경우 간편 설정 옵션을 사용하십시오. 이 옵션을 사용하면 서비스를 시작할 때 각 디바이스에 vNIC가 자동으로 생성됩니다.

1. 에 로그인 AWS OpsHub한 다음 스토리지 관리를 선택합니다.

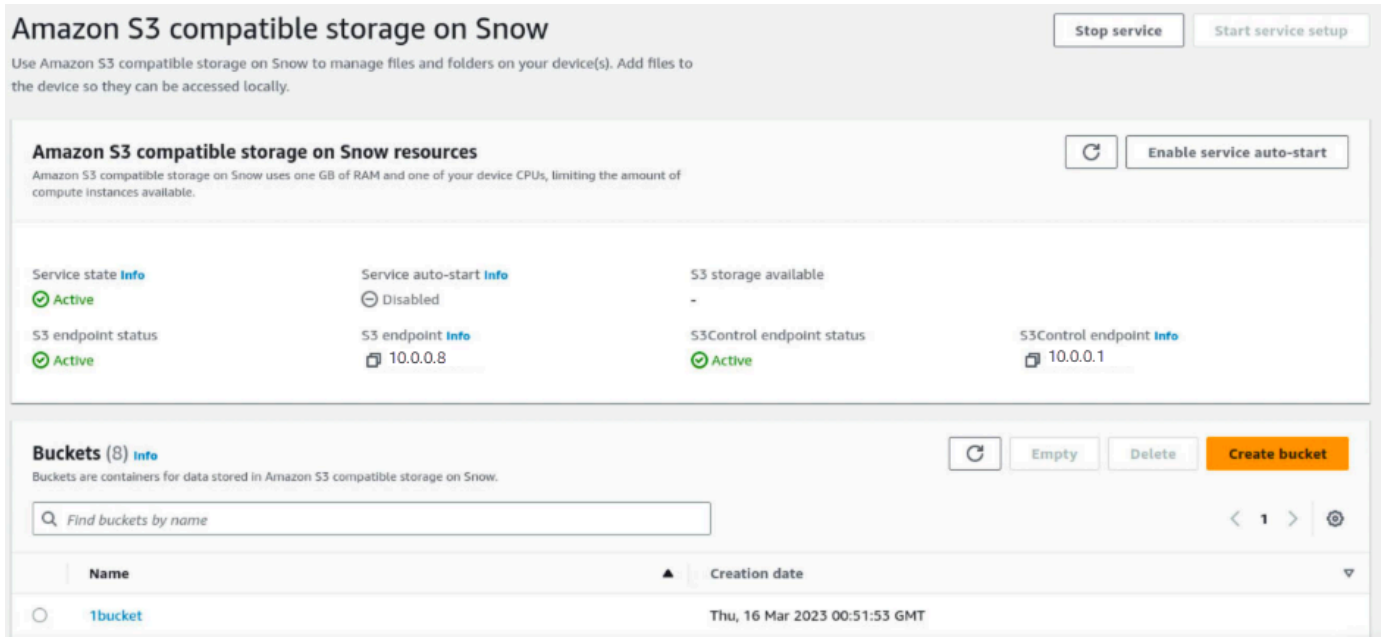
그러면 Amazon S3 compatible storage on Snowball Edge 랜딩 페이지로 이동합니다.

2. 시작 서비스 설정 유형에서 간편을 선택합니다.
3. 서비스 시작을 선택합니다.

### Note

이 작업을 완료하는 데 몇 분 정도 소요되며 사용 중인 디바이스 수에 따라 달라집니다.

서비스가 시작되면 서비스 상태가 활성화되고 엔드포인트가 생깁니다.



## 를 사용하여 Amazon S3 compatible storage on Snowball Edge 고급 설정 옵션 사용 AWS OpsHub

네트워크에서 고정 IP 주소를 사용하거나 기존 VNI를 재사용하려는 경우 고급 설정 옵션을 사용합니다. 이 옵션을 사용하면 각 디바이스의 VNIC를 수동으로 생성합니다.

1. 에 로그인 AWS OpsHub한 다음 스토리지 관리를 선택합니다.

그러면 Amazon S3 compatible storage on Snowball Edge 랜딩 페이지로 이동합니다.

2. 시작 서비스 설정 유형에서 고급을 선택합니다.
3. VNIC를 생성해야 하는 디바이스를 선택합니다.

클러스터의 경우 Snowball Edge에서 Amazon S3 호환 스토리지 서비스를 시작하려면 최소 쿼럼의 디바이스가 필요합니다. 3노드 클러스터의 경우 쿼럼은 2입니다.

### Note

클러스터 설정에서 서비스를 처음 시작하려면 클러스터의 모든 디바이스를 구성하고 서비스를 시작할 수 있어야 합니다. 쿼럼을 충족하면 이후에 시작할 때 일부 디바이스를 사용할 수 있지만 서비스는 성능이 저하된 상태로 시작됩니다.

4. 각 디바이스에 대해 기존 VNIC를 선택하거나 VNI 생성을 선택합니다.

각 디바이스에는 오브젝트 작업을 위한 S3 엔드포인트용 VNIC와 버킷 작업을 위한 S3Control 엔드포인트용 VNIC가 하나 필요합니다.

5. VNIC를 생성하는 경우 물리적 네트워크 인터페이스를 선택하고 상태 IP 주소 및 서브넷 마스크를 입력한 다음 가상 네트워크 인터페이스 생성을 선택합니다.
6. VNICs를 생성한 후 디바이스 시작을 선택합니다.

#### Note

이 작업을 완료하는 데 몇 분 정도 소요되며 사용 중인 디바이스 수에 따라 달라집니다.

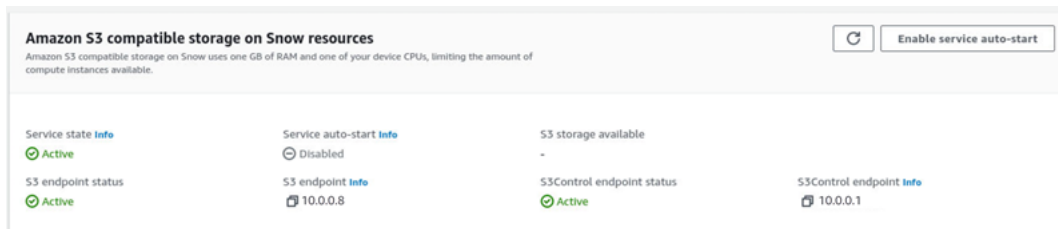
서비스가 시작되면 서비스 상태가 활성화되고 엔드포인트가 생깁니다.

를 사용하여 자동으로 시작하도록 Snowball Edge에서 Amazon S3 호환 스토리지 구성 AWS OpsHub

1. 에 로그인 AWS OpsHub한 다음 스토리지 관리를 선택합니다.

그러면 Amazon S3 compatible storage on Snowball Edge 랜딩 페이지로 이동합니다.

2. Snow 리소스의 Amazon S3 호환 스토리지에서 서비스 자동 시작 활성화를 선택합니다. 시스템은 서비스가 향후에 자동으로 시작되도록 구성합니다.



를 사용하여 Snowball Edge의 Amazon S3 호환 스토리지에서 버킷 생성 AWS OpsHub

AWS OpsHub 인터페이스를 사용하여 Snowball Edge 디바이스에 Amazon S3 버킷을 생성합니다.

1. 를 엽니다 AWS OpsHub.
2. 스토리지 관리에서 시작하기를 선택합니다. Snow의 Amazon S3 호환 스토리지 페이지가 표시됩니다.
3. 버킷에서 버킷 생성을 선택합니다. 버킷 생성 화면이 나타납니다.

- 버킷 이름에서 버킷의 이름을 입력합니다.

#### Note

버킷 이름은 Snowball 디바이스 또는 클러스터 내에서 고유해야 하며 공백이나 대문자를 포함할 수 없습니다.

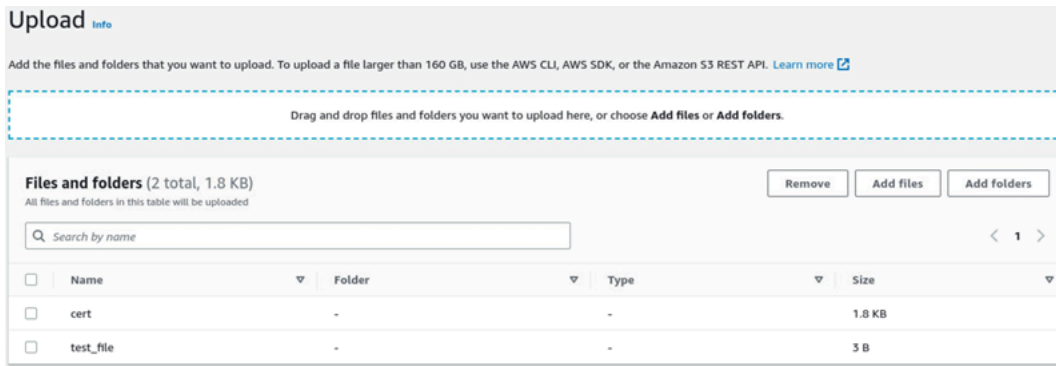
- 버킷 생성을 선택합니다. 시스템에서 버킷을 생성하면 Amazon S3 compatible storage on Snow 페이지의 버킷에 표시됩니다.

## 를 사용하여 Snowball Edge 버킷의 Amazon S3 호환 스토리지에 파일 및 폴더 업로드 AWS OpsHub

AWS OpsHub 인터페이스를 사용하여 Snowball Edge 버킷의 Amazon S3 호환 스토리지에 파일과 폴더를 업로드합니다. 파일 및 폴더는 개별적으로 업로드하거나 함께 업로드할 수 있습니다.

- AWS OpsHub을 엽니다.
- 스토리지 관리의 버킷에서 파일을 업로드할 버킷을 선택합니다. 버킷 페이지가 나타납니다.
- 버킷 페이지에서 파일 업로드를 선택합니다. 업로드 페이지가 나타납니다.





4. 운영 체제 파일 관리자에서 AWS OpsHub 창으로 드래그하여 파일 또는 폴더를 업로드하거나 다음을 수행합니다.
  - a. 파일 추가 또는 폴더 추가를 선택합니다.
  - b. 업로드할 파일이나 폴더를 선택합니다. 열기를 선택합니다.

시스템이 선택한 파일 및 폴더를 디바이스의 버킷에 업로드합니다. 업로드가 완료되면 파일 및 폴더 목록에 파일 및 폴더 이름이 나타납니다.

## 를 AWS OpsHub 사용하여 Snowball Edge 버킷의 Amazon S3 호환 스토리지에서 파일 및 폴더 제거 AWS OpsHub

AWS OpsHub 인터페이스를 사용하여 Snowball Edge 디바이스의 버킷에서 파일과 폴더를 제거하고 영구적으로 삭제합니다.

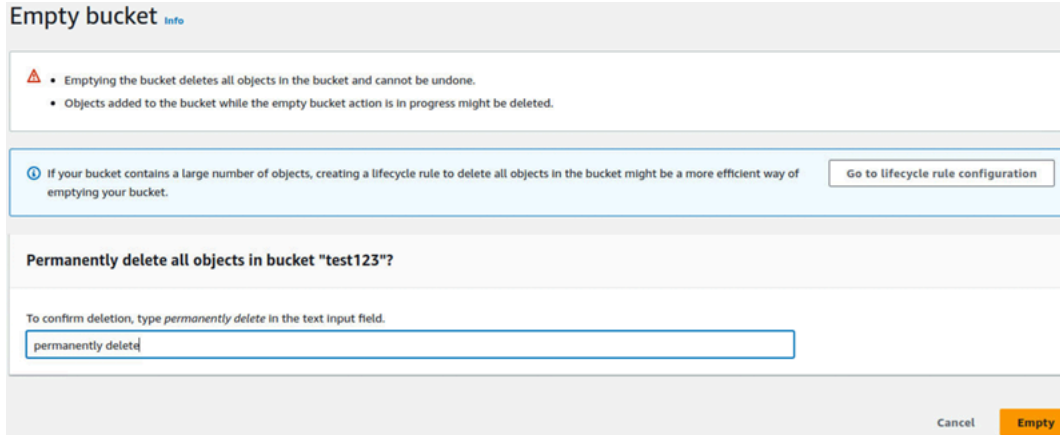
1. 를 엽니다 AWS OpsHub.
2. 스토리지 관리의 버킷에서 파일 및 폴더를 삭제할 버킷의 이름을 선택합니다. 버킷 페이지가 나타납니다.
3. 파일 및 폴더에서 영구 삭제할 파일 및 폴더의 확인란을 선택합니다.
4. 제거를 선택합니다. 시스템이 디바이스의 버킷에서 파일 또는 폴더를 제거합니다.

## Snowball Edge의 Amazon S3 호환 스토리지에서 버킷 삭제

디바이스에서 버킷을 삭제하려면 버킷이 비어 있어야 합니다. 버킷에서 파일 및 폴더를 제거하거나 빈 버킷 도구를 사용합니다. 파일 및 폴더를 제거하려면 [를 AWS OpsHub 사용하여 Snowball Edge 버킷의 Amazon S3 호환 스토리지에서 파일 및 폴더 제거 AWS OpsHub](#)을 참조하십시오.

## 빈 버킷 도구를 사용하려면

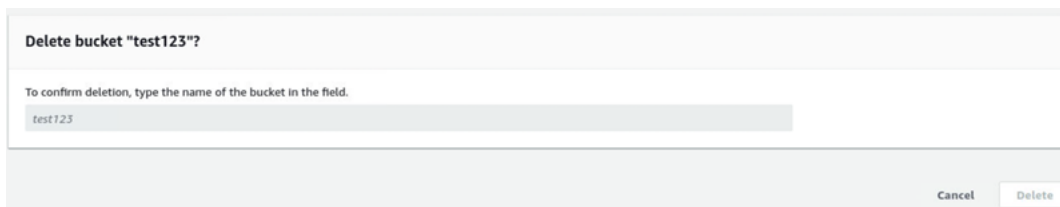
1. 를 엽니다 AWS OpsHub.
2. 스토리지 관리의 버킷에서 비울 버킷의 라디오 버튼을 선택합니다.
3. 비우기를 선택합니다. 빈 버킷 페이지가 나타납니다.



4. 빈 버킷 페이지의 텍스트 상자에 **permanently delete**를 입력합니다.
5. 비우기를 선택합니다. 시스템이 버킷을 비웁니다.

## 빈 버킷 삭제

1. 스토리지 관리의 버킷에서 삭제할 버킷의 라디오 버튼을 선택합니다.
2. 삭제를 선택합니다. 버킷 삭제 페이지가 나타납니다.



3. 버킷 삭제 페이지의 텍스트 상자에 버킷 이름을 입력합니다.
4. 삭제를 선택합니다. 시스템이 디바이스에서 버킷을 삭제합니다.

## 를 사용하여 Amazon S3 어댑터 스토리지 관리 AWS OpsHub

AWS OpsHub 를 사용하여 가져오기 및 내보내기 작업을 위한 Amazon S3(S3) 스토리지를 생성하고 관리할 수 있습니다.

## 주제

- [를 사용하여 Amazon S3 스토리지에 액세스 AWS OpsHub](#)
- [를 사용하여 Amazon S3 스토리지에 파일 업로드 AWS OpsHub](#)
- [를 사용하여 Amazon S3 스토리지에서 파일 다운로드 AWS OpsHub](#)
- [를 사용하여 Amazon S3 스토리지에서 파일 삭제 AWS OpsHub](#)

## 를 사용하여 Amazon S3 스토리지에 액세스 AWS OpsHub

디바이스에 파일을 업로드하고 로컬에서 파일에 액세스할 수 있습니다. 디바이스의 다른 위치로 물리적으로 이동하거나 디바이스가 반환될 AWS 클라우드 때 로 다시 가져올 수 있습니다.

Snowball Edge는 Amazon S3 버킷을 사용하여 디바이스에 파일을 저장하고 관리합니다.

S3 버킷에 액세스하려면

1. AWS OpsHub 애플리케이션을 엽니다.
2. 대시보드의 Manage file storage(파일 스토리지 관리) 섹션에서 Get started(시작하기)를 선택합니다.

Amazon S3 전송 메커니즘을 사용하여 디바이스를 주문한 경우, 해당 디바이스는 파일 및 객체 스토리지 페이지의 버킷 섹션에 표시됩니다. 파일 및 객체 스토리지 페이지에서 각 버킷의 세부 정보를 볼 수 있습니다.

### Note

NFS 전송 메커니즘을 사용하여 디바이스를 주문한 경우 NFS 서비스가 구성되고 활성화 되면 탑재 지점 섹션 아래에 버킷 이름이 표시됩니다. NFS 인터페이스 사용에 대한 자세한 내용은 섹션을 참조하세요 [를 사용하여 NFS 인터페이스 관리 AWS OpsHub](#).

## File & object storage

Use Amazon S3 to manage files and objects stored on your device. Add files to the device so they can be accessed locally. For import jobs, the files will be transferred to AWS when the device is sent back.

### Resources

#### Storage available

925.85 GB available of 925.93 GB

99%

Select a bucket below to start transferring files to your device.

### Buckets (7)

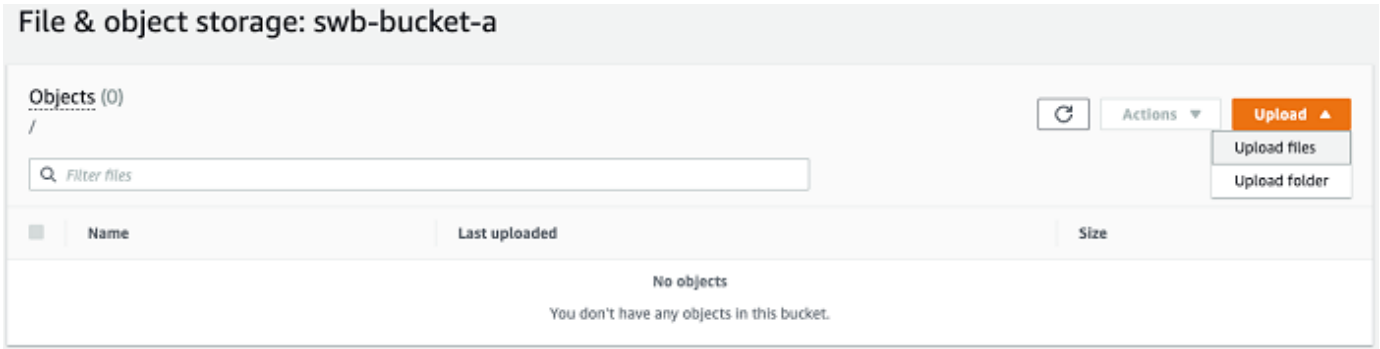
Filter buckets

Bucket name	Date created
sbw-output	Mon, 12 Oct 2009 17:50:30 GMT
swb-bucket-a	Mon, 12 Oct 2009 17:50:30 GMT
swb-bucket-b	Mon, 12 Oct 2009 17:50:30 GMT
swb-bucket-c	Mon, 12 Oct 2009 17:50:30 GMT
swb-bucket-d	Mon, 12 Oct 2009 17:50:30 GMT
swb-bucket-e	Mon, 12 Oct 2009 17:50:30 GMT
swb-bucket-f	Mon, 12 Oct 2009 17:50:30 GMT

## 를 사용하여 Amazon S3 스토리지에 파일 업로드 AWS OpsHub

### 파일을 업로드하려면

1. 대시보드의 Manage file storage(파일 스토리지 관리) 섹션에서 Get started(시작하기)를 선택합니다. 디바이스에 Amazon S3 버킷이 있는 경우 해당 버킷은 파일 스토리지 페이지의 버킷 섹션에 표시됩니다. 이 페이지에서 각 버킷의 세부 정보를 볼 수 있습니다.
2. 파일을 업로드할 버킷을 선택합니다.
3. 업로드를 선택한 다음 파일 업로드를 선택하거나 버킷에 파일을 끌어 놓고 확인을 선택합니다.



**Note**

대용량 파일을 업로드하려면 AWS CLI를 사용하여 Amazon S3의 멀티파트 업로드 기능을 사용할 수 있습니다. S3 CLI 설정 구성에 대한 자세한 내용은 [CLI S3 구성](#)을 참조하십시오. 멀티파트 업로드에 대한 자세한 내용은 Amazon Simple Storage Service 사용 설명서에서 [멀티파트 업로드 개요](#)를 참조하십시오.

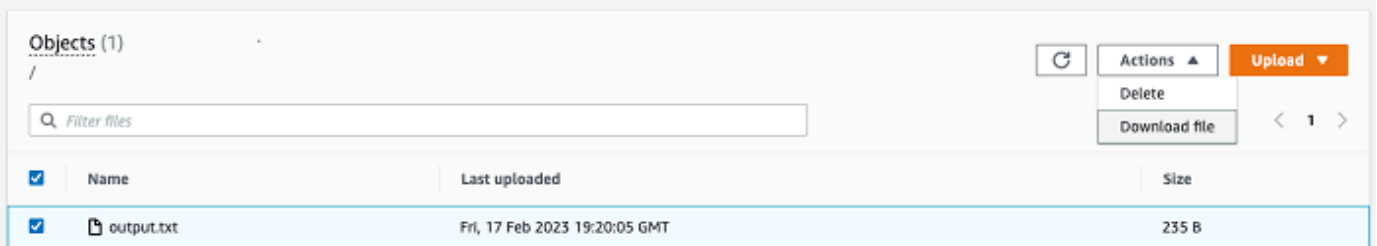
를 사용하여 로컬 시스템에서 Snowball Edge로 폴더를 업로드할 수 AWS OpsHub 있습니다. 폴더 크기가 너무 크면 OpsHub가 파일/폴더 선택을 읽는 데 시간이 좀 걸립니다. OpsHub가 파일과 폴더를 읽는 동안 진행률 추적기는 표시되지 않습니다. 하지만 업로드 프로세스가 시작되면 진행률 추적기가 표시됩니다.

## 를 사용하여 Amazon S3 스토리지에서 파일 다운로드 AWS OpsHub

파일을 다운로드하려면

1. 대시보드의 Manage file storage(파일 스토리지 관리) 섹션에서 Get started(시작하기)를 선택합니다. 디바이스에 S3 버킷이 있는 경우 파일 스토리지 페이지의 버킷 섹션에 해당 S3 버킷이 표시됩니다. 이 페이지에서 각 버킷의 세부 정보를 볼 수 있습니다.
2. 파일을 다운로드할 버킷을 선택하고 다운로드할 파일을 찾습니다. 하나 이상의 파일을 선택합니다.

### File & object storage: swb-bucket-a



3. Actions(작업) 메뉴에서 Download(다운로드)를 선택합니다.
4. 파일을 다운로드할 위치를 선택하고 OK(확인)를 선택합니다.

## 를 사용하여 Amazon S3 스토리지에서 파일 삭제 AWS OpsHub

파일이 더 이상 필요하지 않은 경우 Amazon S3 버킷에서 삭제할 수 있습니다.

### 파일 삭제

1. 대시보드의 Manage file storage(파일 스토리지 관리) 섹션에서 Get started(시작하기)를 선택합니다. 디바이스에 Amazon S3 버킷이 있는 경우 해당 버킷은 파일 스토리지 페이지의 버킷 섹션에 표시됩니다. 이 페이지에서 각 버킷의 세부 정보를 볼 수 있습니다.
2. 파일을 삭제할 버킷을 선택하고 삭제할 파일을 찾습니다.
3. [Actions] 메뉴에서 [Delete]를 선택합니다.
4. 나타나는 대화 상자에서 Confirm delete(삭제 확인)를 선택합니다.

## 를 사용하여 NFS 인터페이스 관리 AWS OpsHub

NFS(Network File System) 인터페이스를 사용하여 디바이스가 운영 체제의 로컬 스토리지인 것처럼 Snowball Edge에 파일을 업로드합니다. 이렇게 하면 파일 복사, 드래그 앤 드롭 또는 기타 그래픽 사용자 인터페이스 기능과 같은 운영 체제의 기능을 사용할 수 있으므로 데이터를 보다 사용자 친화적으로 전송할 수 있습니다. 디바이스의 각 S3 버킷은 NFS 인터페이스 엔드포인트로 사용할 수 있으며 데이터를 복사하도록 탑재할 수 있습니다. NFS 인터페이스는 가져오기 작업에 사용할 수 있습니다.

디바이스 주문 작업이 생성될 때 Snowball Edge 디바이스가 포함하도록 구성된 경우 NFS 인터페이스를 사용할 수 있습니다. 디바이스가 NFS 인터페이스를 포함하도록 구성되지 않은 경우 S3 어댑터 또는 Snowball Edge의 Amazon S3 호환 스토리지를 사용하여 데이터를 전송합니다. S3 어댑터에 대한 자세한 내용은 [를 사용하여 Amazon S3 어댑터 스토리지 관리 AWS OpsHub](#) 섹션을 참조하세요. Snowball Edge의 Amazon S3 호환 스토리지에 대한 자세한 내용은 섹션을 참조하세요 [를 사용하여 Snowball Edge에서 Amazon S3 호환 스토리지 설정 AWS OpsHub](#).

시작되면 NFS 인터페이스가 1GB의 메모리와 1개의 CPU를 사용합니다. 이로 인해 Snowball Edge에서 실행되는 다른 서비스의 수 또는 실행할 수 있는 EC2-compatible 인스턴스의 수가 제한될 수 있습니다.

NFS 인터페이스를 통해 전송되는 데이터는 전송 중에 암호화되지 않습니다. NFS 인터페이스를 구성할 때 CIDR 블록을 제공할 수 있으며 Snowball Edge는 해당 블록에 주소가 있는 클라이언트 컴퓨터에서 NFS 인터페이스에 대한 액세스를 제한합니다.

디바이스의 파일은 AWS로 반환되면 Amazon S3로 전송됩니다. 자세한 내용은 [Amazon S3로 작업 가져오기 Snowball Edge 작동](#) 참조하세요.

컴퓨터 운영 체제에서 NFS를 사용하는 방법에 대한 자세한 내용은 운영 체제 설명서를 참조하세요.

NFS 인터페이스를 사용할 때는 다음 사항에 유의하세요.

- NFS 인터페이스는 디바이스의 데이터 스토리지를 위한 로컬 버킷을 제공합니다. 가져오기 작업의 경우 로컬 버킷의 데이터를 Amazon S3로 가져오지 않습니다.
- 파일 이름은 Snowball Edge의 로컬 S3 버킷에 있는 객체 키입니다. 키 이름은 최대 1,024바이트 길이와 UTF-8 인코딩을 사용하는 유니코드 문자의 시퀀스입니다. 가능한 경우 NFSv4.1을 사용하고 파일 이름을 유니코드 UTF-8 형식으로 인코딩하여 데이터를 성공적으로 가져오는 것이 좋습니다. UTF-8 인코딩으로 인코딩되지 않은 파일 이름은 사용하는 NFS 인코딩에 따라 S3에 업로드되지 않거나 다른 파일 이름으로 S3에 업로드될 수 있습니다.
- 파일 경로의 최대 길이는 1,024자 미만이어야 합니다. Snowball Edge는 1024자보다 큰 파일 경로를 지원하지 않습니다. 이 파일 경로 길이를 초과하면 파일 가져오기 오류가 발생합니다.
- 자세한 내용은 Amazon Simple Storage Service 사용 설명서의 [객체 키](#)를 참조하세요.
- NFS 기반 전송의 경우 Snowball Edge에서 Amazon S3로 가져올 때 표준 POSIX 스타일 메타데이터가 객체에 추가됩니다. 또한 현재 NFS 옵션을 사용한 Amazon S3 for Snowball Edge 가져오기에 대한 내부 가져오기 메커니즘의 일부를 사용 AWS DataSync 하므로 메타데이터 "x-amz-meta-user-agent aws-datasync"가 표시됩니다.
- 단일 Snowball Edge 디바이스를 사용하여 최대 4천만 개의 파일을 전송할 수 있습니다. 단일 작업에서 4천만 개 이상의 파일을 전송해야 하는 경우 파일을 일괄 처리하여 전송할 때마다 파일 수를 줄이십시오. 향상된 NFS 인터페이스 또는 S3 인터페이스를 사용하는 Snowball Edge 디바이스의 경우 개별 파일 크기는 최대 5TB까지 원하는 크기일 수 있습니다.

명령줄 인터페이스(CLI) 도구인 Snowball Edge 클라이언트를 사용하여 NFS 인터페이스를 구성하고 관리할 수도 있습니다. 자세한 내용은 [NFS 인터페이스 관리](#)를 참조하세요.

## 주제

- [Windows 운영 체제에서 NFS 서비스 시작](#)
- [를 사용하여 NFS 인터페이스 자동 구성 AWS OpsHub](#)
- [를 사용하여 NFS 인터페이스 수동 구성 AWS OpsHub](#)
- [를 사용하여 Snowball Edge에서 NFS 엔드포인트 관리 AWS OpsHub](#)
- [클라이언트 컴퓨터에 NFS 엔드포인트 탑재](#)
- [를 사용하여 NFS 인터페이스 중지 AWS OpsHub](#)

## Windows 운영 체제에서 NFS 서비스 시작

클라이언트 컴퓨터에서 Windows 10 Enterprise 또는 Windows 7 Enterprise 운영 체제를 사용하는 경우 AWS OpsHub 애플리케이션에서 NFS를 구성하기 전에 클라이언트 컴퓨터에서 NFS 서비스를 시작합니다.

1. 클라이언트 컴퓨터에서 시작을 열고 제어판을 선택한 다음 프로그램을 선택합니다.
2. Windows 기능 사용/사용 안 함을 선택합니다.

### Note

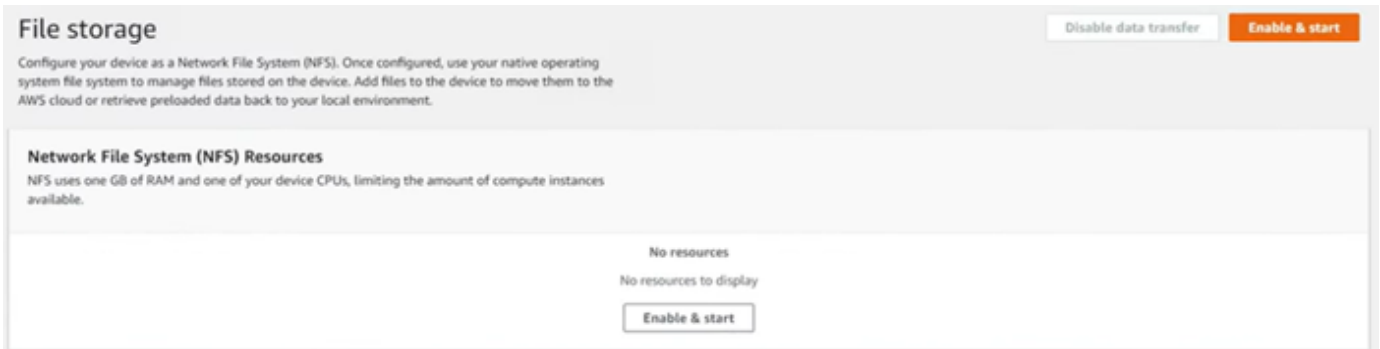
Windows 기능을 켜려면 컴퓨터의 관리자 사용자 이름과 암호를 제공해야 할 수 있습니다.

3. Services for NFS(NFS용 서비스)에서 Client for NFS(NFS용 클라이언트)를 선택하고 확인을 선택합니다.

## 를 사용하여 NFS 인터페이스 자동 구성 AWS OpsHub

NFS 인터페이스는 기본적으로 Snowball Edge 디바이스에서 실행되지 않으므로 디바이스에서 데이터 전송을 활성화하려면 NFS 인터페이스를 시작해야 합니다. 클릭 몇 번으로 Snowball Edge에서 NFS 인터페이스를 빠르고 자동으로 구성할 수 있습니다. NFS 인터페이스를 직접 구성할 수도 있습니다. 자세한 내용은 [를 사용하여 NFS 인터페이스 수동 구성 AWS OpsHub](#) 단원을 참조하십시오.


1. 대시보드의 데이터 전송 섹션에서 활성화 및 시작을 선택합니다. 이 작업을 완료하는 데 1~2분이 걸릴 수 있습니다.



2. NFS 서비스가 시작되면 대시보드에 NFS 인터페이스의 IP 주소가 표시되고 데이터 전송 섹션에 서비스가 활성 상태로 나타납니다.
3. 탐색기에서 열기(Windows 또는 Linux 운영 체제를 사용하는 경우)를 선택하여 운영 체제의 파일 브라우저에서 파일 공유를 열고 Snowball Edge로 파일 전송을 시작합니다. 클라이언트 컴퓨터에



서 파일 공유로 파일을 복사하여 붙여 넣거나 끌어서 놓을 수 있습니다. Windows 운영 체제에서 파일 공유는 다음과 같습니다. buckets(\\12.123.45.679)(Z:).

 Note

Linux 운영 체제에서 NFS 엔드포인트를 탑재하려면 루트 권한이 필요합니다.

## 를 사용하여 NFS 인터페이스 수동 구성 AWS OpsHub

NFS 인터페이스는 기본적으로 Snowball Edge 디바이스에서 실행되지 않으므로 디바이스에서 데이터 전송을 활성화하려면 NFS 인터페이스를 시작해야 합니다. Snowball Edge 디바이스에서 실행되는 가상 네트워크 인터페이스(VNI)의 IP 주소를 제공하고 필요한 경우 파일 공유에 대한 액세스를 제한하여 NFS 인터페이스를 수동으로 구성할 수 있습니다. NFS 인터페이스를 수동으로 구성하기 전에 Snowball Edge 디바이스에서 가상 네트워크 인터페이스(VNI)를 설정합니다. 자세한 내용은 [컴퓨팅 인스턴스를 위한 네트워크 구성](#)을 참조하세요.

Snowball Edge 디바이스가 NFS 인터페이스를 자동으로 구성하도록 할 수도 있습니다. 자세한 내용은 [를 사용하여 NFS 인터페이스 자동 구성 AWS OpsHub](#) 단원을 참조하십시오.

1. 대시보드의 데이터 전송 섹션 하단에서 수동 구성을 선택합니다.

2. 활성화 및 시작을 선택하여 NFS 시작 마법사를 엽니다. 물리적 네트워크 인터페이스 필드가 채워집니다.

## Start NFS ✕

Physical network interface

RJ45: s.ni-8459d6c7273eed333 ▼

Create IP address (VNI)  Use existing IP address (VNI)

IP Address assignment

DHCP ▼

Restrict NFS to allowed hosts  Allow all hosts

Allowed hosts

Provide a set of CIDR blocks allowed to connect to the NFS service.

192.0.2.0/24 ✕

0.0.0.0/0 ✕

Add allowed hosts

Allow instances on this device to access NFS

Enable

Cancel Start NFS

3. IP 주소(VNI) 생성을 선택하거나 기존 IP 주소 사용을 선택합니다.

4. IP 주소 생성(VNI)을 선택한 경우 IP 주소 할당 목록 상자에서 DHCP 또는 고정 IP를 선택합니다.

#### Important

DHCP 네트워크를 사용하는 경우 DHCP 서버가 NFS 인터페이스의 IP 주소를 재할당할 수 있습니다. 이는 디바이스 연결이 끊기고 IP 주소가 재활용된 후에 발생할 수 있습니다. 허용된 호스트 범위를 설정하고 클라이언트 주소가 변경되면 다른 클라이언트가 해당 주소를 선택할 수 있습니다. 이 경우 새로운 클라이언트가 공유에 액세스할 수 있습니다. 이를 방지하려면 DHCP 예약 또는 고정 IP 주소를 사용하십시오.

기존 IP 주소 사용을 선택한 경우 가상 네트워크 인터페이스 목록 상자에서 가상 네트워크 인터페이스를 선택합니다.

5. NFS 인터페이스에 대한 액세스를 제한하고 허용된 네트워크 주소 블록을 제공하거나 네트워크의 모든 디바이스가 Snowball Edge의 NFS 인터페이스에 액세스하도록 허용하도록 선택합니다.
  - Snowball Edge의 NFS 인터페이스에 대한 액세스를 제한하려면 NFS를 허용된 호스트로 제한을 선택합니다. 허용된 호스트에 CIDR 블록 세트를 입력합니다. 둘 이상의 CIDR 블록에 대한 액세스를 허용하려면 다른 블록 세트를 입력합니다. 블록 세트를 제거하려면 블록이 포함된 필드 옆의 X를 선택합니다. 허용된 호스트 추가를 선택합니다.

#### Note

허용된 호스트로 NFS 제한을 선택하고 허용된 CIDR 블록을 제공하지 않으면 Snowball Edge는 NFS 인터페이스 탑재에 대한 모든 요청을 거부합니다.

- 네트워크의 모든 디바이스가 NFS 인터페이스에 액세스하도록 허용하려면 모든 호스트 허용을 선택합니다.
6. Snowball Edge에서 실행되는 EC2-compatible 인스턴스가 NFS 어댑터에 액세스하도록 허용하려면 활성화를 선택합니다.
  7. NFS 시작을 선택합니다. 시작하는 데 1~2분 정도 걸릴 수 있습니다.

#### Important

NFS 인터페이스가 시작되는 동안에는 Snowball Edge를 끄지 마십시오.

NFS(Network File System) 리소스 섹션에서 NFS 인터페이스의 상태가 활성으로 표시됩니다. 인터페이스를 클라이언트 컴퓨터에 로컬 스토리지로 탑재하려면 나열된 IP 주소가 필요합니다.

## 를 사용하여 Snowball Edge에서 NFS 엔드포인트 관리 AWS OpsHub

Snowball Edge의 각 S3 버킷은 엔드포인트로 표시되며 마운트 경로에 나열됩니다. NFS 인터페이스가 시작된 후 엔드포인트를 탑재하여 해당 엔드포인트로 또는 엔드포인트에서 파일을 전송합니다. 한 번에 하나의 엔드포인트만 탑재할 수 있습니다. 다른 엔드포인트를 탑재하려면 먼저 현재 엔드포인트의 탑재를 해제합니다.

### 엔드포인트를 탑재하려면

1. 경로 탑재 섹션에서 다음 중 하나를 수행하여 엔드포인트를 선택합니다.
  - 엔드포인트 필터링 필드에 버킷 이름 전체 또는 일부를 입력하여 항목의 사용 가능한 엔드포인트 목록을 필터링한 다음 엔드포인트를 선택합니다.
  - 탑재 경로 목록에서 탑재할 엔드포인트를 선택합니다.
2. NFS 엔드포인트 탑재를 선택합니다. Snowball Edge는 사용할 엔드포인트를 탑재합니다.

### 엔드포인트를 탑재 해제하려면

1. 탑재 경로 섹션에서 탑재를 해제할 엔드포인트를 선택합니다.
2. 엔드포인트 탑재 해제를 선택합니다. Snowball Edge는 엔드포인트를 탑재 해제하며 더 이상 사용할 수 없습니다.

#### Note

엔드포인트를 탑재 해제하기 전에 엔드포인트에서 또는 엔드포인트로 데이터가 복사되지 않았는지 확인합니다.

## 클라이언트 컴퓨터에 NFS 엔드포인트 탑재

NFS 인터페이스가 시작되고 엔드포인트가 탑재된 후 엔드포인트를 클라이언트 컴퓨터에 로컬 스토리지로 탑재합니다.

1. 탑재 경로에서 탑재할 엔드포인트의 복사 아이콘을 선택합니다. 엔드포인트를 탑재할 때 운영 체제에 붙여넣습니다.
2. 다음은 Windows, Linux 및 macOS 운영 체제에 대한 기본 탑재 명령입니다.

- Windows:

```
mount -o nolock rsize=128 wsize=128 mtype=hard nfs-interface-ip-address:/
buckets/BucketName *
```

- Linux:

```
mount -t nfs nfs-interface-ip-address:/buckets/BucketName mount_point
```

- macOS:

```
mount -t nfs -o vers=3,rsize=131072,wsize=131072,nolocks,hard,retrans=2 nfs-
interface-ip-address:/buckets/$bucketname mount_point
```

## 를 사용하여 NFS 인터페이스 중지 AWS OpsHub

파일 전송을 완료하면 Snowball Edge 디바이스에서 NFS 인터페이스를 중지합니다.

1. 대시보드에서 서비스를 선택한 다음 파일 스토리지를 선택합니다.
2. 파일 스토리지 페이지에서 데이터 전송 비활성화를 선택합니다. 대개 NFS 엔드포인트가 대시보드에서 사라지는 데 최대 2분이 걸립니다.

## 를 사용하여 디바이스 재부팅 AWS OpsHub

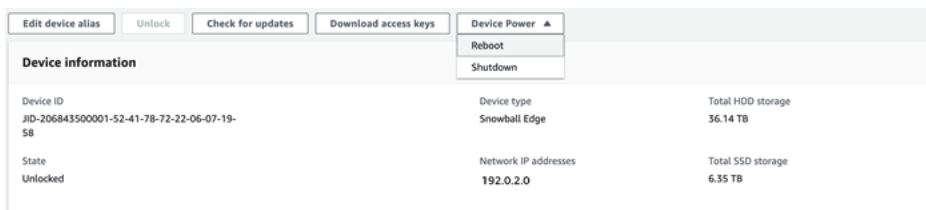
를 사용하여 Snow 디바이스 AWS OpsHub 를 재부팅하려면 다음 단계를 따르세요.

**⚠ Important**

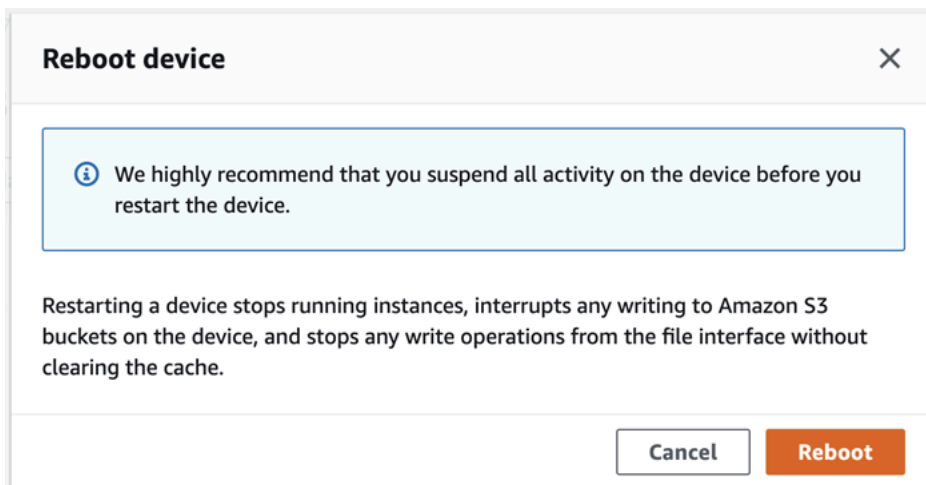
디바이스 재부팅 전에 디바이스의 모든 활동을 일시 중지하는 것이 좋습니다. 디바이스를 재부팅하면 인스턴스 실행이 중지되고 디바이스의 Amazon S3 버킷에 대한 모든 쓰기가 중단됩니다.

## 디바이스를 재부팅하려면

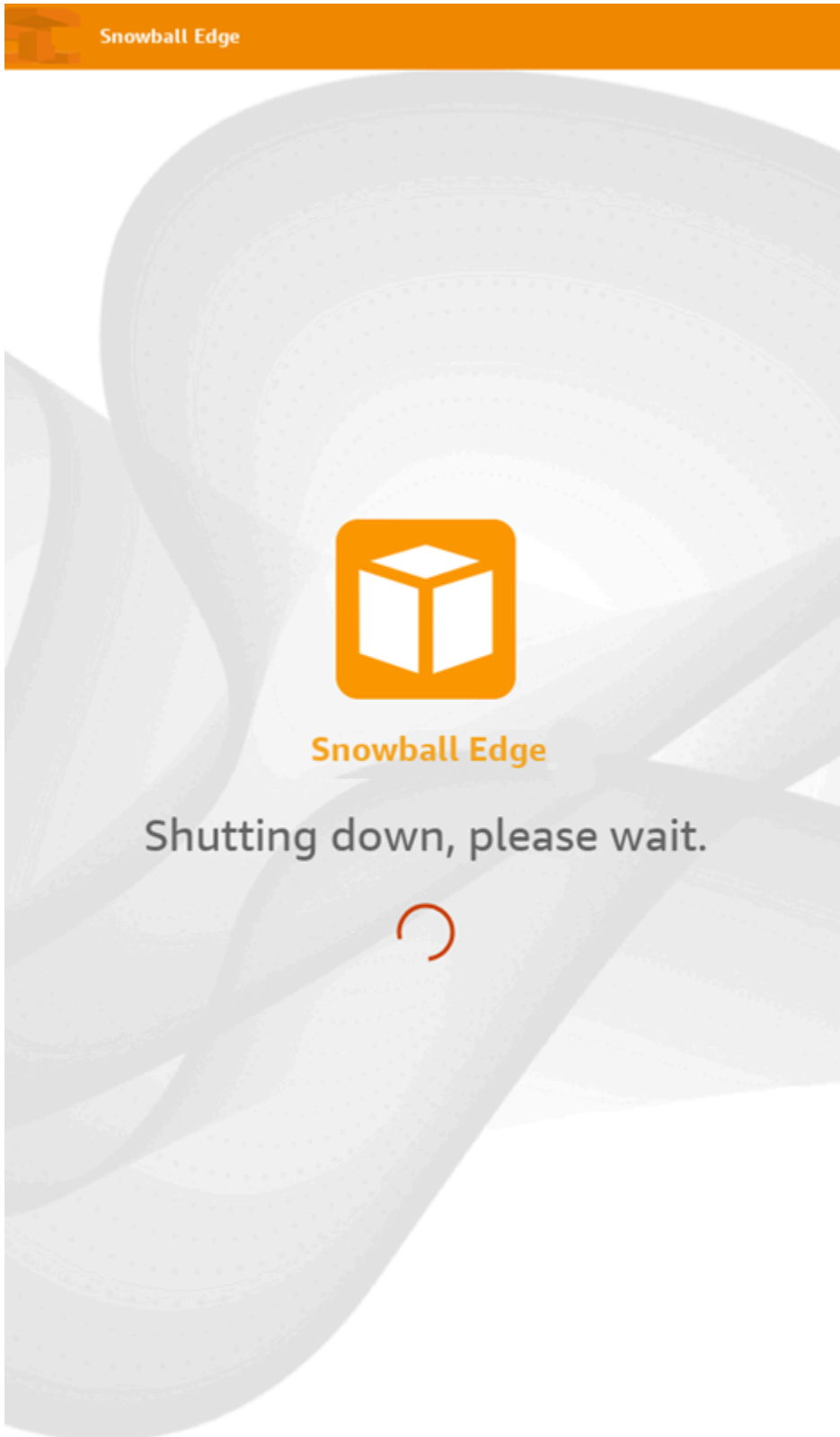
1. AWS OpsHub 대시보드의 디바이스에서 디바이스를 찾습니다. 그런 다음 디바이스를 선택하여 디바이스 세부 정보 페이지를 엽니다.
2. 디바이스 전원 메뉴를 선택한 다음 재부팅을 선택합니다. 대화 상자가 나타납니다.



3. 대화 상자에서 재부팅을 선택합니다. 디바이스가 재부팅되기 시작합니다.



디바이스가 종료되는 동안 LCD 화면에 디바이스가 종료되었음을 알리는 메시지가 표시됩니다.



## 를 사용하여 프로필 관리 AWS OpsHub

로컬 파일 시스템에 있는 자격 증명의 영구 스토리지에 대한 프로필을 생성할 수 있습니다. 를 사용하면 디바이스 IP 주소 AWS OpsHub, 잠금 해제 코드 및 매니페스트 파일을 사용하여 디바이스를 잠금 해제할 때마다 새 프로파일을 생성할 수 있습니다.

Snowball Edge 클라이언트를 사용하여 언제든지 프로필을 생성할 수도 있습니다. [Snowball Edge 클라이언트의 프로필 구성](#)을 참조하십시오.

프로필을 생성하려면

1. 로컬에서 단말기를 잠금 해제하고 [를 사용하여 Snowball Edge 디바이스 잠금 해제 AWS OpsHub](#)의 지침에 따라 로그인합니다.
2. 프로필 이름을 지정하고 프로필 이름 저장을 선택합니다.

## 를 사용하여 디바이스 종료 AWS OpsHub

를 사용하여 Snow 디바이스 AWS OpsHub 를 종료하려면 다음 단계를 따르세요.

### ⚠ Important

디바이스 종료 전에 디바이스의 모든 활동을 일시 중지하는 것이 좋습니다. 디바이스를 종료하면 인스턴스 실행이 중지되고 디바이스의 Amazon S3 버킷에 대한 모든 쓰기가 중단됩니다.

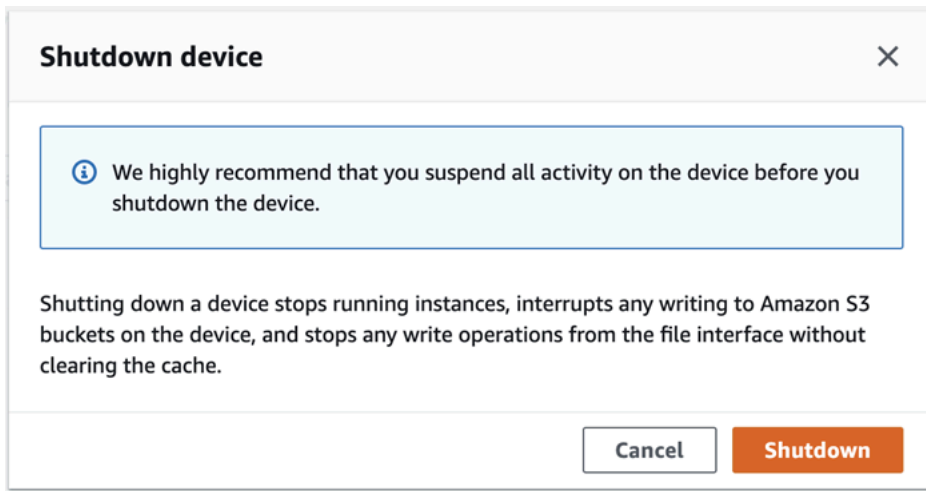
디바이스를 종료하려면

1. AWS OpsHub 대시보드의 디바이스에서 디바이스를 찾습니다. 그런 다음 디바이스를 선택하여 디바이스 세부 정보 페이지를 엽니다.
2. 디바이스 전원 메뉴를 선택한 다음 종료를 선택합니다. 대화 상자가 나타납니다.

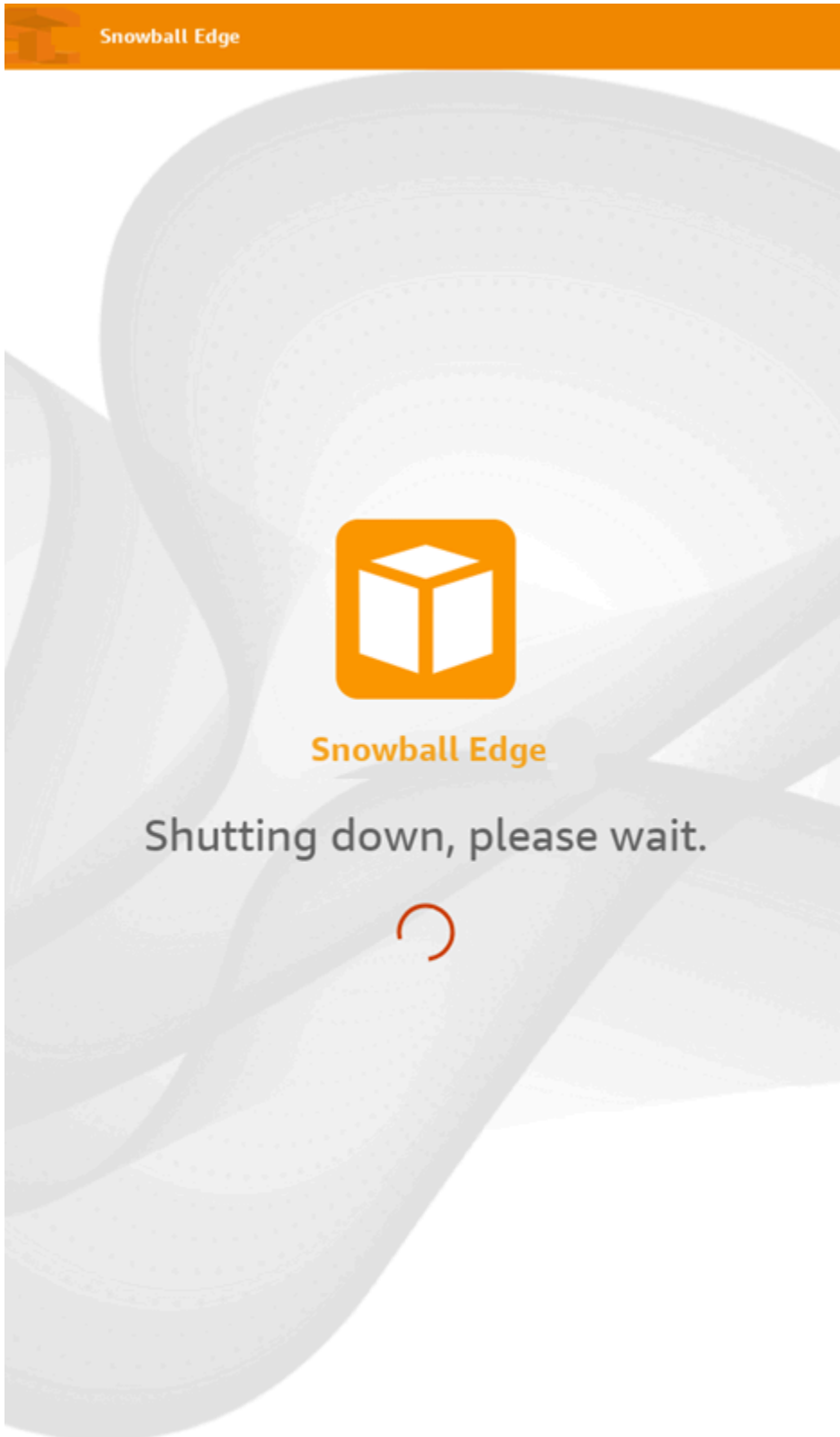


3. 대화 상자에서 종료를 선택합니다. 디바이스가 종료되기 시작합니다.





디바이스가 종료되는 동안 LCD 화면에 디바이스가 종료되었음을 알리는 메시지가 표시됩니다.

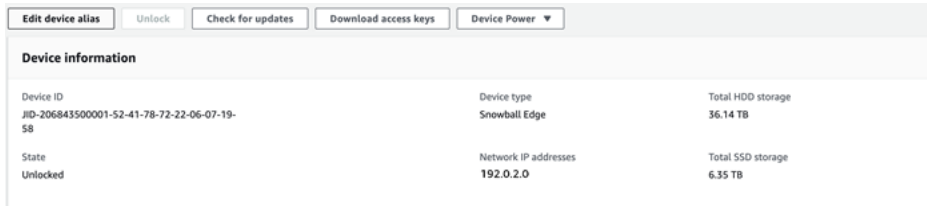


## 를 사용하여 디바이스 별칭 편집 AWS OpsHub

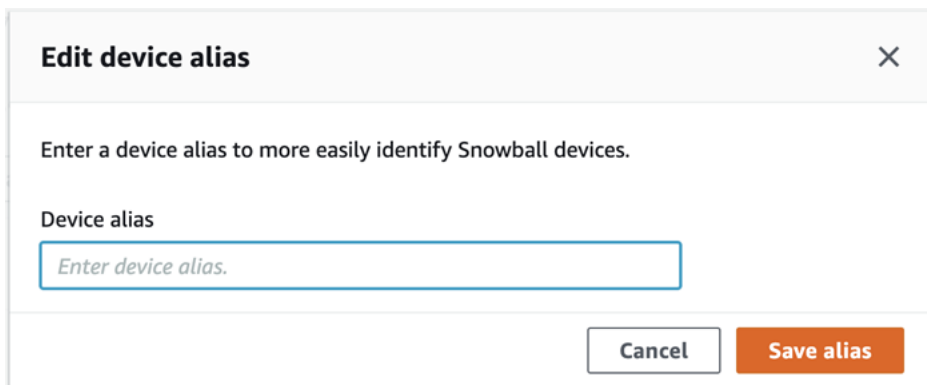
다음 단계에 따라를 사용하여 디바이스 별칭을 편집합니다 AWS OpsHub.

단말기의 별칭을 편집하려면

1. AWS OpsHub 대시보드의 디바이스에서 디바이스를 찾습니다. 디바이스를 선택하여 디바이스 세부 정보 페이지를 엽니다.
2. Edit device alias(디바이스 별칭 편집) 탭을 선택합니다.



3. Device alias(디바이스 별칭)에 새 이름을 입력하고 Save alias(별칭 저장)를 선택합니다.



## OpsHub를 사용한 공개 키 인증서 관리

퍼블릭 키 인증서를 제공하여 HTTPS 프로토콜을 통해 Snowball Edge 디바이스 또는 Snowball Edge 디바이스 클러스터에서 실행되는 AWS 서비스와 안전하게 상호 작용할 수 있습니다. HTTPS 프로토콜을 사용하여 IAM, Amazon EC2, S3 어댑터, Snowball Edge의 Amazon S3 호환 스토리지, Amazon EC2 Systems Manager 및 Snowball Edge 디바이스와 같은 AWS 서비스와 상호 작용할 수 있습니다. AWS STS 있습니다. 디바이스 클러스터의 경우 단일 인증서가 필요하며 클러스터의 모든 디바이스에서 생성할 수 있습니다. Snowball Edge 디바이스에서 인증서를 생성하고 디바이스를 잠금 해제하면 Snowball Edge 클라이언트 명령을 사용하여 인증서를 나열하고, 가져오고, 삭제할 수 있습니다.

Snowball Edge 디바이스는 다음 이벤트가 발생할 때 인증서를 생성합니다.

- Snowball Edge 디바이스 또는 클러스터가 처음으로 잠금 해제됩니다.

- 인증서를 삭제한 후 Snowball Edge 디바이스 또는 클러스터가 잠금 해제됩니다(의 delete-certificate 명령 또는 인증서 갱신 사용 AWS OpsHub).
- 인증서가 만료되면 Snowball Edge 디바이스 또는 클러스터가 재부팅되고 잠금이 해제됩니다.

새 인증서가 생성될 때마다 이전 인증서는 더 이상 유효하지 않습니다. 인증서는 생성된 날로부터 1년 동안 유효합니다.

Snowball Edge 클라이언트를 사용하여 퍼블릭 키 인증서를 관리할 수도 있습니다. 자세한 내용은 [퍼블릭 키 인증서 요청](#)을 참조하십시오.

## 주제

- [OpsHub를 사용하여 퍼블릭 키 인증서 다운로드](#)
- [OpsHub를 사용하여 퍼블릭 키 인증서 갱신](#)

## OpsHub를 사용하여 퍼블릭 키 인증서 다운로드

활성 공개 키 인증서를 컴퓨터에 다운로드할 수 있습니다.

1. AWS OpsHub 대시보드의 디바이스에서 디바이스를 찾습니다. 디바이스를 선택하여 디바이스 세부 정보 페이지를 엽니다.
2. 디바이스 세부 정보 페이지에서 인증서 관리 메뉴를 선택합니다. 메뉴에서 인증서 다운로드를 선택합니다.
3. 다운로드할 인증서 파일의 이름을 지정하고 컴퓨터에서 다운로드할 위치를 선택할 수 있는 창이 나타납니다. 저장을 선택합니다.

## OpsHub를 사용하여 퍼블릭 키 인증서 갱신

퍼블릭 키 인증서를 갱신하기 전에 Snowball Edge 디바이스와의 모든 데이터 전송을 중지하고 실행 중인 EC2-compatible을 중지합니다. 자세한 내용은 이 설명서의 [Amazon EC2-compatible 인스턴스 중지](#)를 참조하세요.

1. AWS OpsHub 대시보드의 디바이스에서 디바이스를 찾습니다. 디바이스를 선택하여 디바이스 세부 정보 페이지를 엽니다.
2. 디바이스 세부 정보 페이지에서 인증서 관리 메뉴를 선택합니다. 메뉴에서 인증서 갱신을 선택합니다.

3. 인증서 갱신 창에서 필드에 **Renew**를 입력하고 갱신을 선택합니다. Snowball Edge 디바이스는 기존 퍼블릭 키 인증서를 삭제하고 디바이스 또는 클러스터를 재부팅합니다.

## Renew certificate ✕

### The following certificate will be deleted:

arn:aws:snowball-device:::certificate/example



**Stop all activity on the Snow device or cluster before proceeding.**

Clicking **Renew** will automatically reboot **all devices attached to this certificate** and terminate any ongoing data transfers and other running processes. A new certificate will be generated when you unlock the device or cluster after it reboots.

To confirm, enter **Renew** in the field and then choose **Renew**

Cancel

Renew

## Snowball Edge에 대한 업데이트 가져오기

디바이스의 업데이트를 확인하고 설치할 수 있습니다. 버전.

### 디바이스 업데이트

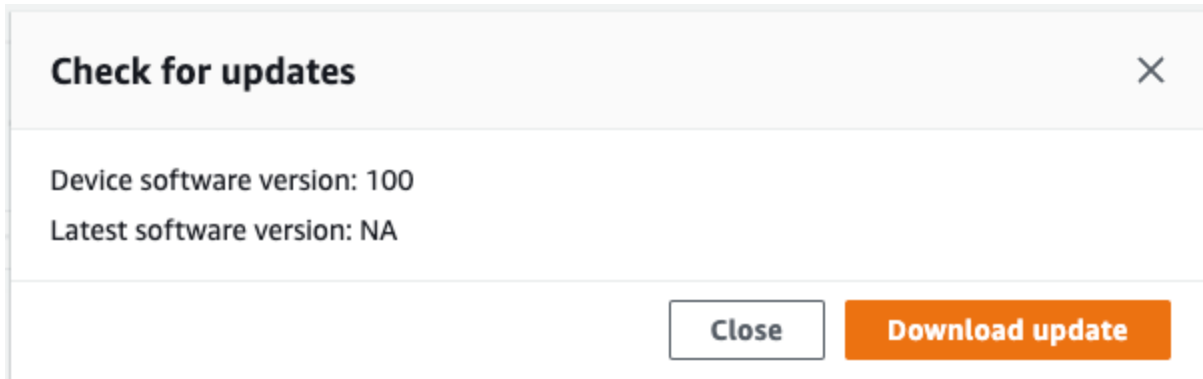
를 사용하여 Snow 디바이스 AWS OpsHub 를 업데이트하려면 다음 단계를 따르세요.

### 디바이스를 업데이트하려면

1. AWS OpsHub 대시보드의 디바이스에서 디바이스를 찾습니다. 디바이스를 선택하여 디바이스 세부 정보 페이지를 엽니다.

2. Check for updates(업데이트 확인) 탭을 선택합니다.

Check for updates(업데이트 확인) 페이지에는 디바이스의 현재 소프트웨어 버전과 최신 소프트웨어 버전(있는 경우)이 표시됩니다.

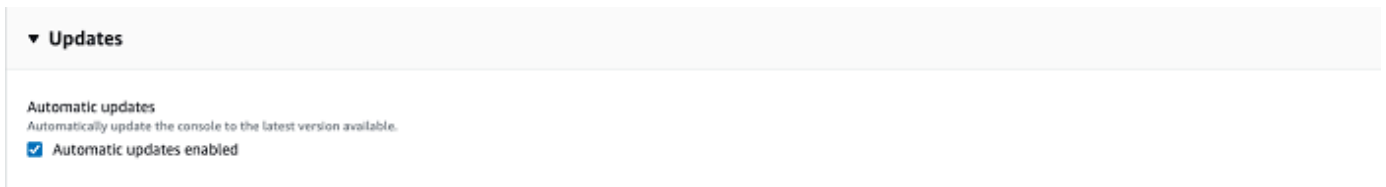


3. 업데이트가 있는 경우 업데이트 다운로드를 선택합니다. 그렇지 않으면 [Close]를 선택합니다.

## AWS OpsHub 애플리케이션 업데이트

에 자동 업데이트가 활성화되어 있는지 확인하려면 AWS OpsHub

1. AWS OpsHub 대시보드에서 기본 설정을 선택합니다.
2. 업데이트 탭을 엽니다.
3. 자동 업데이트 활성화가 선택되어 있는지 확인합니다. 자동 업데이트는 기본적으로 활성화되어 있습니다.



자동 업데이트 활성화를 선택하지 않으면 AWS OpsHub 애플리케이션의 최신 버전이 표시되지 않습니다.

## 를 사용하여 관리 작업 자동화 AWS OpsHub

AWS OpsHub 를 사용하여 Snowball Edge에서 자주 수행하는 운영 작업을 자동화할 수 있습니다. 가상 서버 재시작, Amazon EC2 호환 인스턴스 중지 등과 같이 리소스에서 수행할 반복적인 활동에 대해

작업을 만들 수 있습니다. 운영 작업을 안전하게 수행하고 AWS 리소스에서 작업을 대량으로 실행하는 자동화 문서를 제공합니다. 일반적인 IT 워크플로우를 예약할 수도 있습니다.

### Note

클러스터에서는 작업 자동화가 지원되지 않습니다.

작업을 사용하려면 Amazon EC2 Systems Manager 서비스를 먼저 시작해야 합니다. 자세한 내용은 [Snowball Edge에서 Snowball Edge 디바이스 관리 활성화](#)를 참조하세요.

## 주제

- [를 사용하여 작업 생성 및 시작 AWS OpsHub](#)
- [에서 작업 세부 정보 보기 AWS OpsHub](#)
- [에서 작업 삭제 AWS OpsHub](#)

## 를 사용하여 작업 생성 및 시작 AWS OpsHub

작업을 생성할 때 작업을 실행할 리소스 유형을 지정한 다음 작업을 실행하는 지침이 포함된 작업 문서를 제공합니다. 작업 문서는 YAML 또는 JSON 형식입니다. 그런 다음 작업에 필요한 파라미터를 제공하고 작업을 시작합니다.

### 태스크를 만들려면

1. 대시보드의 Launch task(작업 시작) 섹션에서 Get started(시작하기)를 선택하여 Tasks(작업) 페이지를 엽니다. 작업을 생성한 경우 Tasks(작업) 아래에 표시됩니다.
2. Create task(작업 생성)를 선택하고 작업에 대한 세부 정보를 제공합니다.
3. Name(이름)에 고유한 작업 이름을 입력합니다.

### Tip

이름은 3~128자 이내로 작성해야 합니다. 유효한 문자는 a-z, A-Z, 0-9, ., \_ 및 -입니다.

4. 선택적으로 Target type - optional(대상 유형 - 선택 사항) 목록에서 대상 유형을 선택할 수 있습니다. 작업을 실행할 리소스 유형입니다.

예를 들어, Amazon EC2 호환 인스턴스에서 실행할 태스크에는 `/AWS::EC2::Instance`를 지정하고 모든 리소스 유형에서 실행할 작업에는 `/`를 지정할 수 있습니다.

5. 콘텐츠 섹션에서 YAML 또는 JSON을 선택하고 작업을 실행하는 스크립트를 제공합니다. YAML 또는 JSON 형식의 두 가지 옵션이 있습니다. 예를 보려면 [의 작업 예제 AWS OpsHub](#) 섹션을 참조하십시오.
6. 생성(Create)을 선택합니다. 그러면 생성한 작업이 Tasks(작업) 페이지에 나타납니다.

### 작업을 시작하려면

1. 대시보드의 Launch task(작업 시작) 섹션에서 Get started(시작하기)를 선택하여 Tasks(작업) 페이지를 엽니다. 작업이 Tasks(작업)아래에 나타납니다.
2. 작업을 선택하여 Start task(작업 시작) 페이지를 엽니다.
3. 대상에서 실행하도록 Simple execution(단순 실행)을 선택합니다.

여러 대상에서 안전하게 실행하고 동시성 및 오류 임계값을 정의하려면 Rate control(속도 제어)을 선택합니다. 이 옵션의 경우 Rate control(속도 제어) 섹션에서 추가 대상 및 오류 임계값 정보를 제공합니다.

4. 필요한 입력 파라미터를 제공하고 Start task(작업 시작)를 선택합니다.

작업의 상태가 Pending(대기 중)으로 나타나며 작업이 성공적으로 실행되면 Success(성공)로 변경됩니다.

### 의 작업 예제 AWS OpsHub

다음 예시는 Amazon EC2 호환 인스턴스를 다시 시작합니다. 두 개의 입력 파라미터 endpoint 및 instance ID가 필요합니다.

#### YAML 예

```
description: Restart EC2 instance
schemaVersion: '0.3'
parameters:
  Endpoint:
    type: String
    description: (Required) EC2 Service Endpoint URL
  Id:
    type: String
    description: (Required) Instance Id
mainSteps:
  - name: restartInstance
```



```

action: aws:executeScript
description: Restart EC2 instance step
inputs:
  Runtime: python3.7
  Handler: restart_instance
  InputPayload:
    Endpoint: "{{ Endpoint }}"
    Id: "{{ Id }}"
  TimeoutSeconds: 30
  Script: |-
    import boto3
    import time
    def restart_instance(payload, context):
        ec2_endpoint = payload['Endpoint']
        instance_id = payload['Id']
        ec2 = boto3.resource('ec2', endpoint_url=ec2_endpoint)
        instance = ec2.Instance(instance_id)
        if instance.state['Name'] != 'stopped':
            instance.stop()
            instance.wait_until_stopped()
        instance.start()
        instance.wait_until_running()
        return {'InstanceState': instance.state}

```

## JSON 예

```

{
  "description" : "Restart EC2 instance",
  "schemaVersion" : "0.3",
  "parameters" : {
    "Endpoint" : {
      "type" : "String",
      "description" : "(Required) EC2 Service Endpoint URL"
    },
    "Id" : {
      "type" : "String",
      "description" : "(Required) Instance Id"
    }
  },
  "mainSteps" : [ {
    "name" : "restartInstance",
    "action" : "aws:executeScript",

```

```

"description" : "Restart EC2 instance step",
"inputs" : {
  "Runtime" : "python3.7",
  "Handler" : "restart_instance",
  "InputPayload" : {
    "Endpoint" : "{{ Endpoint }}",
    "Id" : "{{ Id }}"
  },
  "TimeoutSeconds" : 30,
  "Script" : "import boto3\nimport time\ndef restart_instance(payload, context):\n\n    ec2_endpoint = payload['Endpoint']\n    instance_id = payload['Id']\n    ec2 = boto3.resource('ec2', endpoint_url=ec2_endpoint)\n    instance = ec2.Instance(instance_id)\n    if instance.state['Name'] != 'stopped':\n    instance.stop()\n    instance.wait_until_stopped()\n    instance.start()\n    instance.wait_until_running()\n    return {'InstanceState': instance.state}"
}
} ]
}

```

## 에서 작업 세부 정보 보기 AWS OpsHub

작업을 실행하는 데 필요한 설명 및 파라미터와 같은 관리 작업의 세부 정보를 볼 수 있습니다.

작업의 세부 정보를 보려면

1. 대시보드의 Launch task(작업 시작) 섹션에서 Get started(시작하기)를 선택하여 Tasks(작업) 페이지를 엽니다.
2. Tasks(작업) 페이지에서 세부 정보를 보려는 작업을 찾아 선택합니다.
3. View details(세부 정보 보기)를 선택하고 탭 중 하나를 선택하여 세부 정보를 확인합니다. 예를 들어 Parameters(파라미터) 탭에는 스크립트의 입력 파라미터가 표시됩니다.

## 에서 작업 삭제 AWS OpsHub

관리 작업을 삭제하려면 다음 단계를 수행하십시오.

## 작업을 삭제하려면

1. 대시보드의 Launch task(작업 시작) 섹션에서 Get started(시작하기)를 선택하여 Tasks(작업) 페이지를 엽니다.
2. 삭제할 작업을 찾습니다. 작업을 선택한 다음 Delete(삭제)를 선택합니다.

## 를 사용하여 디바이스의 NTP 시간 서버 설정 AWS OpsHub

다음 단계에 따라 디바이스에서 시간을 동기화해야 하는 시간 서버를 확인하고 업데이트하십시오.

### 시간 소스를 확인하려면

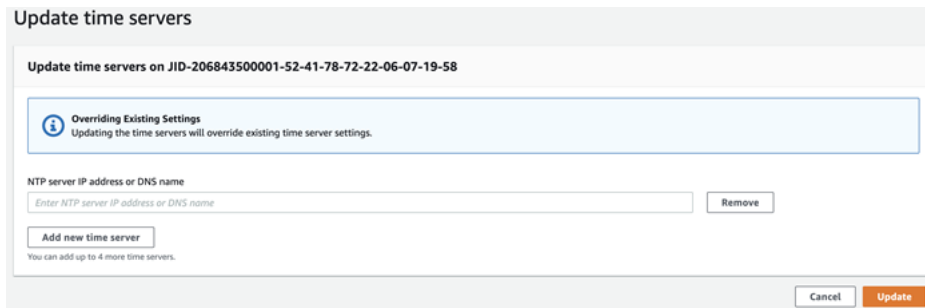
1. AWS OpsHub 대시보드의 디바이스에서 디바이스를 찾습니다. 디바이스를 선택하여 디바이스 세부 정보 페이지를 엽니다.
2. 디바이스가 시간을 동기화하고 있는 시간 소스 목록이 시간 소스 테이블에 표시됩니다.

시간 소스 테이블에는 다음과 같은 네 개의 열이 있습니다.

- 주소: 시간 원본의 DNS 이름/IP 주소
- 상태: 디바이스와 해당 시간 소스 간의 현재 연결 상태로, 가능한 상태는 다음과 같이 5가지입니다.
  - 현재: 현재 시간 소스는 시간을 동기화하는 데 사용되고 있습니다.
  - 결합: 시간 소스가 현재 소스와 결합됩니다.
  - 제외: 결합 알고리즘에 의해 시간 소스가 제외됩니다.
  - 분실됨: 시간 소스와의 연결이 끊어졌습니다.
  - 사용할 수 없음: 결합 알고리즘이 잘못된 티커로 간주되거나 변동성이 너무 큰 잘못된 시간 소스입니다.
- 유형: NTP(Network Time Protocol) 소스는 서버 또는 피어일 수 있습니다. 서버는 사용자가 update-time-server 명령을 사용하여 설정할 수 있지만, 피어는 클러스터의 다른 Snowball Edge 디바이스로만 설정할 수 있으며 클러스터가 연결되면 자동으로 설정됩니다.
- 계층: 소스의 계층입니다. 계층 1은 로컬로 연결된 기준 클럭이 있는 소스를 나타냅니다. 계층 1 소스와 동기화된 소스는 계층 2에 있습니다. 계층 2 소스와 동기화된 소스는 계층 3에 설정되는 방식으로 계속됩니다.

## 타임 서버를 업데이트하려면

1. AWS OpsHub 대시보드의 디바이스에서 디바이스를 찾습니다. 디바이스를 선택하여 디바이스 세부 정보 페이지를 엽니다.
2. 디바이스가 시간을 동기화하고 있는 시간 소스 목록이 시간 소스 테이블에 표시됩니다.
3. 시간 소스 테이블에서 시간 서버 업데이트를 선택합니다.
4. 디바이스에서 시간을 동기화할 시간 서버의 DNS 이름 또는 IP 주소를 입력하고 업데이트를 선택합니다.



## 지원되는 NTP 디바이스 유형 및 소프트웨어 버전

NTP는 모든 버전 2 스토리지 및 컴퓨팅 디바이스 유형에서 사용할 수 없습니다. 하지만 소프트웨어 버전 77 이상의 Snowball Edge 버전 3 스토리지 및 컴퓨팅 디바이스 유형은 NTP를 지원합니다. NTP가 활성화되었는지 확인하려면 Snowball Edge CLI 명령 `describe-time-sources`을 사용하십시오.

# Snowball Edge 클라이언트 구성 및 사용

Snowball Edge 클라이언트는 Snowball Edge 또는 Snowball Edge 클러스터로 작업하는 데 사용할 수 있는 AWS의 명령줄 인터페이스(CLI) 도구입니다. 클라이언트를 사용하여 Snowball Edge 또는 디바이스 클러스터를 잠금 해제하고, Snowball Edge를 설정하고, 디바이스에서 서비스를 시작 및 중지하고, 디바이스와 데이터를 주고받을 수 있습니다. Snowball Edge 클라이언트는 Linux, macOS 및 Windows 운영 체제에서 실행되는 컴퓨터와 호환됩니다.

## 주제

- [Snowball Edge 클라이언트 다운로드 및 설치](#)
- [Snowball Edge 클라이언트의 프로필 구성](#)
- [Snowball Edge 클라이언트 버전 찾기](#)
- [Snowball Edge에 대한 자격 증명 가져오기](#)
- [Snowball Edge에서 서비스 시작](#)
- [Snowball Edge에서 서비스 중지](#)
- [Snowball Edge에서 로그 보기 및 다운로드](#)
- [Snowball Edge 상태 보기](#)
- [Snowball Edge에서 실행되는 서비스의 위반 상태](#)
- [Snowball Edge의 기능 상태 보기](#)
- [Snowball Edge에 대한 시간 서버 설정](#)
- [Snowball Edge NFC 태그를 검증하기 위한 QR 코드 가져오기](#)
- [MTU 크기 업데이트](#)

## Snowball Edge 클라이언트 다운로드 및 설치

[AWS Snowball Edge 리소스](#)에서 Snowball Edge 클라이언트를 다운로드할 수 있습니다. 해당 페이지에서 운영 체제에 대한 설치 패키지를 찾을 수 있습니다.

아래 지침에 따라 클라이언트를 설치하고 구성합니다.

## Linux

### Note

Snowball Edge 클라이언트는 64비트 리눅스 배포판에서만 지원됩니다.

1. `snowball-client-linux.tar.gz` 파일의 압축을 풉니다. `/snowball-client-linux-build_number/bin` 폴더 구조를 생성하고 해당 폴더에서 파일을 추출합니다.
2. 다음 명령을 실행하여 폴더를 구성합니다.

```
chmod u+x snowball-client-linux-build_number/bin/snowballEdge
```

```
chmod u+x snowball-client-linux-build_number/jre/bin/java
```

3. 운영 체제의 `$PATH` 환경 변수에 `/snowball-client-linux-build_number/bin`을 추가하여 모든 디렉터리에서 Snowball Edge 클라이언트 명령을 실행합니다. 자세한 내용은 디바이스의 운영 체제 또는 셸 설명서를 참조하세요.

## macOS

1. `snowball-client-mac.tar.gz` 파일의 압축을 풉니다. `/snowball-client-linux-build_number/bin` 폴더 구조를 생성하고 해당 폴더에서 파일을 추출합니다.
2. 다음 명령을 실행하여 폴더를 구성합니다.

```
chmod u+x snowball-client-mac-build_number/bin/snowballEdge
```

```
chmod u+x snowball-client-mac-build_number/jre/bin/java
```

3. 운영 체제의 \$PATH 환경 변수에 `/snowball-client-mac-build_number/bin`을 추가하여 모든 디렉터리에서 Snowball Edge 클라이언트 명령을 실행합니다. 자세한 내용은 디바이스의 운영 체제 또는 셸 설명서를 참조하세요.

## Windows

클라이언트는 Microsoft MSI(Software Installer) 파일로 패키징됩니다. 파일을 열고 설치 마법사의 프롬프트를 따릅니다. 클라이언트를 설치한 후 별도의 준비 없이 모든 디렉터리에서 클라이언트를 실행할 수 있습니다.

## Snowball Edge 클라이언트의 프로필 구성

Snowball Edge 클라이언트에 대한 명령을 실행할 때마다 매니페스트 파일, 잠금 해제 코드 및 Snowball Edge의 IP 주소를 제공합니다. 명령을 실행할 때마다 이를 제공하는 대신 `configure` 명령을 사용하여 매니페스트 파일 경로, 29자 잠금 해제 코드 및 엔드포인트(Snowball Edge의 IP 주소)를 프로파일로 저장할 수 있습니다. 구성 후 명령에 프로파일 이름을 포함시켜 각 명령에 이러한 값을 수동으로 입력할 필요 없이 Snowball Edge 클라이언트 명령을 사용할 수 있습니다. Snowball Edge 클라이언트를 구성하면 해당 정보가 일반 텍스트 JSON 형식으로 `home directory/.aws/snowball/config/snowball-edge.config`에 저장됩니다. 이 파일을 생성할 수 있도록 환경이 구성되어 있는지 확인합니다.

### Important

구성 파일에 액세스할 수 있는 사람은 누구나 Snowball Edge 디바이스 또는 클러스터의 데이터에 액세스할 수 있습니다. 이 파일에 대한 로컬 액세스 제어를 관리하는 것은 관리자 책임 중 하나입니다.

AWS OpsHub 를 사용하여 프로필을 생성할 수도 있습니다. 에서 생성된 프로필 AWS OpsHub 은 Snowball Edge 클라이언트에서 사용할 수 있으며에서 생성된 프로필 AWS OpsHub 은 Snowball Edge 클라이언트에서 사용할 수 있습니다. 자세한 내용은 [프로필 관리](#)를 참조하세요.

### 프로필을 생성하려면

1. 운영 체제의 명령줄 인터페이스에 명령을 입력합니다. `profile-name` 파라미터의 값은 프로필의 이름입니다. Snowball Edge 클라이언트 명령을 실행할 때 나중에 제공합니다.

```
snowballEdge configure --profile profile-name
```

2. Snowball Edge 클라이언트는 각 파라미터에 대한 프롬프트를 표시합니다. 메시지가 표시되면 환경 및 Snowball Edge에 대한 정보를 입력합니다.

### Note

endpoint 파라미터의 값은 Snowball Edge의 IP 주소이며 앞에 붙습니다 `https://`. 디바이스 전면의 LCD 화면에서 Snowball Edge 디바이스의 IP 주소를 찾을 수 있습니다.

### Example **configure** 명령의 출력

```
Configuration will stored at home directory\.aws\snowball\config\snowball-
edge.config
Snowball Edge Manifest Path: /Path/to/manifest/file
Unlock Code: 29 character unlock code
Default Endpoint: https://192.0.2.0
```

Snowball Edge 클라이언트는 잠금 해제 코드가 매니페스트 파일에 올바른지 확인합니다. 일치하지 않으면 명령이 중지되고 프로필이 생성되지 않습니다. 잠금 해제 코드와 매니페스트 파일을 확인하고 명령을 다시 실행합니다.

프로파일을 사용하려면 명령 구문 `--profile profile-name` 뒤에를 포함합니다.

독립형 Snowball Edge를 여러 개 사용하는 경우 각각에 대한 프로파일을 생성할 수 있습니다. 다른 프로파일을 생성하려면 `configure` 명령을 다시 실행하고, `--profile` 파라미터에 다른 값을 제공하고, 다른 디바이스에 대한 정보를 제공합니다.

### Example에서 snowball-edge.config 파일

이 예시에서는 세 개의 프로필(SnowDevice1profile, SnowDevice2profile, SnowDevice3profile)이 포함된 프로필 파일을 보여 줍니다.

```
{"version":1,"profiles":
  {
    "SnowDevice1profile":
      {
```



```
    "name": "SnowDevice1profile",
    "jobId": "JID12345678-136f-45b4-b5c2-847db8adc749",
    "unlockCode": "db223-12345-dbe46-44557-c7cc2",
    "manifestPath": "C:\\Users\\Administrator\\.aws\\ops-hub\\manifest\\
\\JID12345678-136f-45b4-b5c2-847db8adc749_manifest-1670622989203.bin",
    "defaultEndpoint": "https://10.16.0.1",
    "isCluster": false,
    "deviceIps": []
  },
},
"SnowDevice2profile":
{
  "name": "SnowDevice2profile",
  "jobId": "JID12345678-fdb2-436a-a4ff-7c510dec1bae",
  "unlockCode": "b893b-54321-0f65c-6c5e1-7f748",
  "manifestPath": "C:\\Users\\Administrator\\.aws\\ops-hub\\manifest\\JID12345678-
fdb2-436a-a4ff-7c510dec1bae_manifest-1670623746908.bin",
  "defaultEndpoint": "https://10.16.0.2",
  "isCluster": false,
  "deviceIps": []
},
"SnowDevice3profile":
{
  "name": "SnowDevice3profile",
  "jobId": "JID12345678-c384-4a5e-becd-ab5f38888463",
  "unlockCode": "64c89-13524-4d054-13d93-c1b80",
  "manifestPath": "C:\\Users\\Administrator\\.aws\\ops-hub\\manifest\\JID12345678-
c384-4a5e-becd-ab5f38888463_manifest-1670623999136.bin",
  "defaultEndpoint": "https://10.16.0.3",
  "isCluster": false,
  "deviceIps": []
}
}
```

프로필을 편집하거나 삭제하려면 텍스트 편집기에서 프로필 파일을 편집합니다.

프로필을 편집하려면

1. 텍스트 편집기에서 *home directory*\\.aws\\snowball\\config에 있는 snowball-edge.config를 엽니다.

**Note**

이 파일을 읽고 쓸 수 있게 액세스할 수 있도록 환경이 구성되어 있는지 확인합니다.

2. 필요에 따라 파일을 편집합니다. 예를 들어 프로필과 연결된 Snowball Edge의 IP 주소를 변경하려면 `defaultEndpoint` 항목을 변경합니다.
3. 파일을 저장하고 닫습니다.

**프로필을 삭제하려면**

1. 텍스트 편집기를 사용하여, `home directory`\.aws\snowball\config에서 `snowball-edge.config`을 엽니다.

**Note**

이 파일을 읽고 쓸 수 있게 액세스할 수 있도록 환경이 구성되어 있는지 확인합니다.

2. 프로필 이름이 들어 있는 줄, 프로필 이름 뒤에 오는 중괄호 { }, 해당 괄호 안의 내용을 삭제합니다.
3. 파일을 저장하고 닫습니다.

## Snowball Edge 클라이언트 버전 찾기

`version` 명령을 사용하여 Snowball Edge 명령줄 인터페이스(CLI) 클라이언트의 버전을 확인합니다.

**사용량**

```
snowballEdge version
```

**출력 예**

```
Snowball Edge client version: 1.2.0 Build 661
```

## Snowball Edge에 대한 자격 증명 가져오기

snowballEdge list-access-keys 및 snowballEdge get-secret-access-key 명령을 사용하여 Snowball Edge에서의 관리자 사용자의 자격 증명을 가져올 수 AWS 계정 있습니다. 이러한 자격 증명을 사용하여 AWS Identity and Access Management (IAM 사용자) 및 역할을 생성하고를 사용하거나 AWS SDK와 AWS CLI 함께 사용할 때 요청을 인증할 수 있습니다. 이러한 자격 증명은 Snowball Edge에 대한 개별 작업에만 연결되고, 디바이스 또는 디바이스 클러스터에서만 사용할 수 있습니다. 디바이스 한 대 또는 여러 대는 AWS 클라우드내에서 IAM 권한이 없습니다.

### Note

Snowball Edge AWS CLI 에서를 사용하는 경우 CLI를 구성할 때 이러한 자격 증명을 사용해야 합니다. 에 대한 자격 증명 구성에 대한 자세한 내용은 AWS Command Line Interface 사용 설명서의 [구성](#)을 [AWS CLI](#) AWS CLI참조하세요.

사용량(구성된 Snowball Edge 클라이언트)

```
snowballEdge list-access-keys
```

Example출력

```
{
  "AccessKeyIds" : [ "AKIAIOSFODNN7EXAMPLE" ]
}
```

사용량(구성된 Snowball Edge 클라이언트)

```
snowballEdge get-secret-access-key --access-key-id Access Key
```

Example출력

```
[snowballEdge]
aws_access_key_id = AKIAIOSFODNN7EXAMPLE
aws_secret_access_key = wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY
```

## Snowball Edge에서 서비스 시작

Snowball Edge 디바이스는 여러 가지 서비스를 지원합니다. 여기에는 컴퓨팅 인스턴스, NFS(Network File System) 인터페이스, Snowball Edge Device Management 및 포함됩니다 AWS IoT Greengrass. Amazon S3 어댑터 서비스, Amazon EC2 AWS STS 및 IAM은 기본적으로 시작되며 중지하거나 다시 시작할 수 없습니다. 그러나 NFS 인터페이스인 Snowball Edge Device Management 및 `start-service` 명령과 함께 서비스 ID를 사용하여 시작할 AWS IoT Greengrass 수 있습니다. 각 서비스의 서비스 ID를 가져오려면 `list-services` 명령을 사용하면 됩니다.

이 명령을 실행하기 전에 가상 네트워크 인터페이스를 하나 생성하여 시작하려는 서비스에 바인딩합니다. 자세한 내용은 [Snowball Edge에서 가상 네트워크 인터페이스 생성](#) 단원을 참조하십시오.

```
snowballEdge start-service --service-id service_id --virtual-network-interface-arns virtual-network-interface-arn --profile profile-name
```

### Example `start-service` 명령의 출력

Starting the AWS service on your Snowball Edge. You can determine the status of the AWS service using the `describe-service` command.

## Snowball Edge에서 서비스 중지

Snowball Edge에서 실행되는 서비스를 중지하려면 `stop-service` 명령을 사용할 수 있습니다.

Amazon S3 어댑터, Amazon EC2 AWS STS 및 IAM 서비스는 중지할 수 없습니다.

### Warning

남은 버퍼 상태 데이터가 디바이스에 쓰여지기 전에 네트워크 파일 시스템(NFS)이 중지되면 데이터가 손실될 수 있습니다. NFS 서비스 사용에 대한 자세한 정보는 [Snowball Edge에서 NFS 인터페이스 관리](#) 섹션을 참조하세요.

### Note

Amazon S3 compatible storage on Snowball Edge 서비스를 중지하면 디바이스 또는 클러스터의 S3 버킷에 저장된 데이터에 대한 액세스가 비활성화됩니다. Snowball Edge의 Amazon

S3 호환 스토리지가 다시 시작되면 액세스가 복원됩니다. Snowball Edge에서 Amazon S3 호환 스토리지가 활성화된 디바이스의 경우 Snowball Edge 디바이스의 전원을 켜 후 서비스를 시작하는 것이 좋습니다. 이 설명서의 [Snowball Edge 설정](#)을 참조하세요.

```
snowballEdge stop-service --service-id service_id --profile profile-name
```

### Example **stop-service** 명령의 출력

Stopping the AWS service on your Snowball Edge. You can determine the status of the AWS service using the describe-service command.

## Snowball Edge에서 로그 보기 및 다운로드

온프레미스 데이터 센터와 Snowball Edge 간에 데이터를 전송하면 로그가 자동으로 생성됩니다. 디바이스에 데이터를 전송하는 동안 예기치 않은 오류가 발생할 경우 다음 명령을 사용하여 로그 사본을 로컬 서버에 저장할 수 있습니다.

다음은 로그와 관련된 세 가지 명령입니다.

- **list-logs** - 로그 목록을 JSON 형식으로 반환합니다. 이 목록은 로그 크기(바이트), 로그의 ARN, 로그의 서비스 ID, 로그 유형을 보고합니다.

### 사용량

```
snowballEdge list-logs --profile profile-name
```

### Example **list-logs** 명령의 출력

```
{
  "Logs" : [ {
    "LogArn" : "arn:aws:snowball-device::log/s3-storage-JIEXAMPLE2f-1234-4953-a7c4-dfEXAMPLE709",
    "LogType" : "SUPPORT",
    "ServiceId" : "s3",
    "EstimatedSizeBytes" : 53132614
  }, {
```

```

    "LogArn" : "arn:aws:snowball-device::log/fileinterface-JIDEXAMPLEf-1234-4953-
a7c4-dfEXAMPLE709",
    "LogType" : "CUSTOMER",
    "ServiceId" : "fileinterface",
    "EstimatedSizeBytes" : 4446
  }]
}

```

- `get-log` - 특정 로그의 사본을 Snowball Edge에서 지정된 경로의 디바이스로 다운로드합니다. CUSTOMER 로그는 .zip 형식으로 저장되므로, 이 로그 유형의 압축을 풀어 콘텐츠를 볼 수 있습니다. SUPPORT 로그는 암호화되므로 AWS Support만 읽을 수 있습니다. 로그의 이름과 경로를 지정할 수 있습니다.

### 사용량

```

snowballEdge get-log --log-arn arn:aws:snowball-device::log/fileinterface-
JIDEXAMPLEf-1234-4953-a7c4-dfEXAMPLE709 --profile profile-name

```

### Example `get-log` 명령의 출력

```

Logs are being saved to download/path/snowball-edge-logs-1515EXAMPLE88.bin

```

- `get-support-logs` - 모든 SUPPORT 유형 로그의 사본을 Snowball Edge에서 지정된 경로의 서비스로 다운로드합니다.

### 사용량

```

snowballEdge get-support-logs --profile profile-name

```

### Example `get-support-logs` 명령의 출력

```

Logs are being saved to download/path/snowball-edge-logs-1515716135711.bin

```

### Important

CUSTOMER 유형에는 자체 데이터에 대한 민감한 정보가 포함되었을 수 있습니다. 민감할 수 있는 이 정보를 보호하려면 작업을 마친 후 이러한 로그를 삭제하는 것이 좋습니다.

# Snowball Edge 상태 보기

describe-device 명령을 사용하여 Snowball Edge의 상태 및 일반 상태를 확인할 수 있습니다.

```
snowballEdge describe-device --profile profile-name
```

## Example describe-device 명령의 출력

```
    {
  "DeviceId": "JID-EXAMPLE12345-123-456-7-890",
  "UnlockStatus": {
    "State": "UNLOCKED"
  },
  "ActiveNetworkInterface": {
    "IpAddress": "192.0.2.0"
  },
  "PhysicalNetworkInterfaces": [
    {
      "PhysicalNetworkInterfaceId": "s.ni-EXAMPLEd9ecbf03e3",
      "PhysicalConnectorType": "RJ45",
      "IpAddressAssignment": "STATIC",
      "IpAddress": "0.0.0.0",
      "Netmask": "0.0.0.0",
      "DefaultGateway": "192.0.2.1",
      "MacAddress": "EX:AM:PL:E0:12:34"
    },
    {
      "PhysicalNetworkInterfaceId": "s.ni-EXAMPLE4c3840068f",
      "PhysicalConnectorType": "QSFP",
      "IpAddressAssignment": "STATIC",
      "IpAddress": "0.0.0.0",
      "Netmask": "0.0.0.0",
      "DefaultGateway": "192.0.2.2",
      "MacAddress": "EX:AM:PL:E0:56:78"
    },
    {
      "PhysicalNetworkInterfaceId": "s.ni-EXAMPLE0a3a6499fd",
      "PhysicalConnectorType": "SFP_PLUS",
      "IpAddressAssignment": "DHCP",
      "IpAddress": "192.168.1.231",
      "Netmask": "255.255.255.0",
      "DefaultGateway": "192.0.2.3",
```

```

    "MacAddress": "EX:AM:PL:E0:90:12"
  }
]
}

```

## Snowball Edge에서 실행되는 서비스의 위반 상태

describe-service 명령을 사용하여 Snowball Edge 디바이스에서 실행하는 서비스의 상태 및 일반 상태를 확인할 수 있습니다. 먼저 list-services 명령을 실행하여 실행 중인 서비스를 볼 수 있습니다.

### • list-services

#### 사용량

```
snowballEdge list-services --profile profile-name
```

#### Example **list-services** 명령의 출력

```

{
  "ServiceIds" : [ "greengrass", "fileinterface", "s3", "ec2", "s3-snow" ]
}

```

### • describe-service

이 명령은 서비스의 상태 값을 반환합니다. 반환되는 값에는 서비스 사용 시 발생하는 문제를 해결하는 데 도움이 될 수 있는 상태 정보도 포함되어 있습니다. 이러한 상태는 다음과 같습니다.

- ACTIVE - 서비스가 실행 중이며 사용 가능합니다.
- ACTIVATING - 서비스가 시작되는 중이지만 아직 사용할 수는 없습니다.
- DEACTIVATING - 서비스가 종료되는 중입니다.
- DEGRADED - Snowball Edge의 Amazon S3 호환 스토리지의 경우 이 상태는 클러스터에 있는 하나 이상의 디스크 또는 디바이스가 다운되었음을 나타냅니다. Snowball Edge의 Amazon S3 호환 스토리지 서비스가 중단 없이 실행되고 있지만 데이터 손실 위험을 최소화하기 위해 클러스터 쿼럼이 손실되기 전에 영향을 받는 디바이스를 복구하거나 교체해야 합니다. 이 설명서의 [클러스터링 개요](#)를 참조하세요.
- INACTIVE - 서비스가 실행되고 있지 않으므로 사용할 수 없습니다.

#### 사용량



```
snowballEdge describe-service --service-id service-id --profile profile-name
```

### Example **describe-service** 명령의 출력

```
{
  "ServiceId": "s3",
  "Status": {
    "State": "ACTIVE"
  },
  "Storage": {
    "TotalSpaceBytes": 99608745492480,
    "FreeSpaceBytes": 99608744468480
  },
  "Endpoints": [
    {
      "Protocol": "http",
      "Port": 8080,
      "Host": "192.0.2.0"
    },
    {
      "Protocol": "https",
      "Port": 8443,
      "Host": "192.0.2.0",
      "CertificateAssociation": {
        "CertificateArn": "arn:aws:snowball-
device::certificate/6d955EXAMPLEdb71798146EXAMPLE3f0"
      }
    }
  ]
}
```

### Example Snowball Edge의 Amazon S3 호환 스토리지 서비스 출력

이 `describe-service` 명령은 `service-id` 파라미터의 `s3-snow` 값에 대해 다음과 같은 출력을 제공합니다.

```
{
  "ServiceId" : "s3-snow",
  "Autostart" : false,
```

```
"Status" : {
  "State" : "ACTIVE"
},
"ServiceCapacities" : [ {
  "Name" : "S3 Storage",
  "Unit" : "Byte",
  "Used" : 640303104,
  "Available" : 219571981512
} ],
"Endpoints" : [ {
  "Protocol" : "https",
  "Port" : 443,
  "Host" : "10.0.2.123",
  "CertificateAssociation" : {
    "CertificateArn" : "arn:aws:snowball-device::certificate/
a65ba817f2c5ac9683fc3bc1ae123456"
  },
  "Description" : "s3-snow bucket API endpoint",
  "DeviceId" : "JID6ebd4c50-c3a1-4b16-b32c-b254f9b7f2dc",
  "Status" : {
    "State" : "ACTIVE"
  }
}, {
  "Protocol" : "https",
  "Port" : 443,
  "Host" : "10.0.3.202",
  "CertificateAssociation" : {
    "CertificateArn" : "arn:aws:snowball-device::certificate/
a65ba817f2c5ac9683fc3bc1ae123456"
  },
  "Description" : "s3-snow object API endpoint",
  "DeviceId" : "JID6ebd4c50-c3a1-4b16-b32c-b254f9b7f2dc",
  "Status" : {
    "State" : "ACTIVE"
  }
}, {
  "Protocol" : "https",
  "Port" : 443,
  "Host" : "10.0.3.63",
  "CertificateAssociation" : {
    "CertificateArn" : "arn:aws:snowball-device::certificate/
a65ba817f2c5ac9683fc3bc1ae123456"
  },
  "Description" : "s3-snow bucket API endpoint",
```

```
"DeviceId" : "JID2a1e0deb-38b1-41f8-b904-a396c62da70d",
"Status" : {
  "State" : "ACTIVE"
}
}, {
  "Protocol" : "https",
  "Port" : 443,
  "Host" : "10.0.2.243",
  "CertificateAssociation" : {
    "CertificateArn" : "arn:aws:snowball-device::certificate/
a65ba817f2c5ac9683fc3bc1ae123456"
  },
  "Description" : "s3-snow object API endpoint",
  "DeviceId" : "JID2a1e0deb-38b1-41f8-b904-a396c62da70d",
  "Status" : {
    "State" : "ACTIVE"
  }
}, {
  "Protocol" : "https",
  "Port" : 443,
  "Host" : "10.0.2.220",
  "CertificateAssociation" : {
    "CertificateArn" : "arn:aws:snowball-device::certificate/
a65ba817f2c5ac9683fc3bc1ae123456"
  },
  "Description" : "s3-snow bucket API endpoint",
  "DeviceId" : "JIDcc45fa8f-b994-4ada-a821-581bc35d8645",
  "Status" : {
    "State" : "ACTIVE"
  }
}, {
  "Protocol" : "https",
  "Port" : 443,
  "Host" : "10.0.2.55",
  "CertificateAssociation" : {
    "CertificateArn" : "arn:aws:snowball-device::certificate/
a65ba817f2c5ac9683fc3bc1ae123456"
  },
  "Description" : "s3-snow object API endpoint",
  "DeviceId" : "JIDcc45fa8f-b994-4ada-a821-581bc35d8645",
  "Status" : {
    "State" : "ACTIVE"
  }
}, {
```

```
"Protocol" : "https",
"Port" : 443,
"Host" : "10.0.3.213",
"CertificateAssociation" : {
  "CertificateArn" : "arn:aws:snowball-device::certificate/
a65ba817f2c5ac9683fc3bc1ae123456"
},
"Description" : "s3-snow bucket API endpoint",
"DeviceId" : "JID4ec68543-d974-465f-b81d-89832dd502db",
"Status" : {
  "State" : "ACTIVE"
}
}, {
  "Protocol" : "https",
  "Port" : 443,
  "Host" : "10.0.3.144",
  "CertificateAssociation" : {
    "CertificateArn" : "arn:aws:snowball-device::certificate/
a65ba817f2c5ac9683fc3bc1ae123456"
  },
  "Description" : "s3-snow object API endpoint",
  "DeviceId" : "JID4ec68543-d974-465f-b81d-89832dd502db",
  "Status" : {
    "State" : "ACTIVE"
  }
}, {
  "Protocol" : "https",
  "Port" : 443,
  "Host" : "10.0.2.143",
  "CertificateAssociation" : {
    "CertificateArn" : "arn:aws:snowball-device::certificate/
a65ba817f2c5ac9683fc3bc1ae123456"
  },
  "Description" : "s3-snow bucket API endpoint",
  "DeviceId" : "JID6331b8b5-6c63-4e01-b3ca-eab48b5628d2",
  "Status" : {
    "State" : "ACTIVE"
  }
}, {
  "Protocol" : "https",
  "Port" : 443,
  "Host" : "10.0.3.224",
  "CertificateAssociation" : {
```

```

    "CertificateArn" : "arn:aws:snowball-device:::certificate/
a65ba817f2c5ac9683fc3bc1ae123456"
  },
  "Description" : "s3-snow object API endpoint",
  "DeviceId" : "JID6331b8b5-6c63-4e01-b3ca-eab48b5628d2",
  "Status" : {
    "State" : "ACTIVE"
  }
} ]
}

```

## Snowball Edge의 기능 상태 보기

Snowball Edge에서 사용할 수 있는 기능의 상태를 나열하려면 `describe-features` 명령을 사용합니다.

`RemoteManagementState`는 Snowball Edge 디바이스 관리의 상태를 나타내며 다음 상태 중 하나를 반환합니다.

- `INSTALLED_ONLY` - 기능이 설치되었지만 활성화되지 않았습니다.
- `INSTALLED_AUTOSTART` - 기능이 활성화되고 디바이스가 켜져 있을 AWS 리전 때 디바이스가에 연결을 시도합니다.
- `NOT_INSTALLED` - 디바이스가 기능을 지원하지 않거나 출시 전에 이미 현장에 있었습니다.

### 사용량

```
snowballEdge describe-features --profile profile-name
```

### Example `describe-features` 명령의 출력

```

{
  "RemoteManagementState" : String
}

```

## Snowball Edge에 대한 시간 서버 설정

Snowball Edge 클라이언트 명령을 사용하여 현재 NTP(Network Time Protocol) 구성을 보고 시간을 제공할 서버 또는 피어를 선택할 수 있습니다. 디바이스가 잠긴 상태와 잠금 해제된 상태 모두에 있을 때 Snowball Edge 클라이언트 명령을 사용할 수 있습니다.

안전한 NTP 시간 서버를 제공하는 것은 사용자의 책임입니다. 디바이스가 연결되는 NTP 타임 서버를 설정하려면 `update-time-servers` CLI 명령을 사용합니다.

### Snowball Edge의 시간 소스 확인

디바이스가 현재 연결되어 있는 NTP 시간 소스를 확인하려면 `describe-time-sources` 명령을 사용하세요.

```
snowballEdge describe-time-sources --profile profile-name
```

#### Example `describe-time-sources` 명령의 출력

```
{
  "Sources" : [ {
    "Address" : "172.31.2.71",
    "State" : "LOST",
    "Type" : "PEER",
    "Stratum" : 10
  }, {
    "Address" : "172.31.3.203",
    "State" : "LOST",
    "Type" : "PEER",
    "Stratum" : 10
  }, {
    "Address" : "172.31.0.178",
    "State" : "LOST",
    "Type" : "PEER",
    "Stratum" : 10
  }, {
    "Address" : "172.31.3.178",
    "State" : "LOST",
    "Type" : "PEER",
    "Stratum" : 10
  }, {
    "Address" : "216.239.35.12",
    "State" : "CURRENT",
```

```

    "Type" : "SERVER",
    "Stratum" : 1
  } ]
}

```

이 `describe-time-sources` 명령은 시간 소스 상태 목록을 반환합니다. 각 시간 소스 상태에는 Address, State, Type, Stratum 필드가 포함됩니다. 이러한 필드의 의미는 다음과 같습니다.

- Address - 시간 소스의 DNS 이름/IP 주소입니다.
- State - 디바이스와 해당 시간 소스 간의 현재 연결 상태입니다. 다음과 같은 다섯 가지 상태가 있을 수 있습니다.
  - CURRENT - 시간 소스가 현재 시간을 동기화하는 데 사용되고 있습니다.
  - COMBINED - 시간 소스가 현재 소스와 결합됩니다.
  - EXCLUDED - 시간 소스가 결합 알고리즘에 의해 제외됩니다.
  - LOST - 시간 소스와의 연결이 끊어졌습니다.
  - UNACCEPTABLE - 결합 알고리즘이 잘못된 티커로 간주되거나 변동성이 너무 큰 잘못된 시간 소스입니다.
- Type - NTP 시간 소스는 서버 또는 피어일 수 있습니다. `update-time-servers` 명령으로 서버를 설정할 수 있습니다. 피어는 클러스터의 다른 Snowball Edge 디바이스만 될 수 있으며 클러스터가 연결될 때 자동으로 설정됩니다.
- Stratum - 이 필드에는 소스의 계층이 표시됩니다. 계층 1은 로컬로 연결된 기준 클럭이 있는 소스를 나타냅니다. 계층 1 소스와 동기화된 소스는 계층 2에 있습니다. 계층 2 소스와 동기화된 소스는 계층 3에 있는 방식입니다.

NTP 시간 소스는 서버 또는 피어일 수 있습니다. 사용자는 `update-time-servers` 명령을 사용하여 서버를 설정할 수 있지만 피어는 클러스터의 다른 Snowball Edge 디바이스일 수 있습니다. 예제 출력에서 `describe-time-sources`는 클러스터 5에 있는 Snowball Edge에서 호출됩니다. 출력에는 피어 4개와 서버 1개가 포함됩니다. 피어의 계층은 10이고 서버의 계층은 1이므로 서버가 현재 시간 소스로 선택됩니다.

## 타임 서버 업데이트

`update-time-servers` 명령과 시간 서버 주소를 사용하여 NTP용 NTP 서버 또는 피어를 사용하도록 Snowball Edge를 구성합니다.

```
snowballEdge update-time-servers time-server-address --profile profile-name
```

**Note**

이 `update-time-servers` 명령은 이전 NTP 시간 서버 설정을 재정의합니다.

**Example `update-time-servers` 명령의 출력**

```
Updating time servers now.
```

## Snowball Edge NFC 태그를 검증하기 위한 QR 코드 가져오기

이 명령을 사용하여 AWS Snowball Edge Verification 앱에 사용할 디바이스 고유의 QR 코드를 생성할 수 있습니다. NFC 검증에 대한 자세한 내용은 [NFC 태그 검증](#) 섹션을 참조하세요.

**사용량**

```
snowballEdge get-app-qr-code --output-file ~/downloads/snowball-qr-code.png --  
profile profile-name
```

**Example 출력**

```
QR code is saved to ~/downloads/snowball-qr-code.png
```

## MTU 크기 업데이트

`update-mtu-size` 명령을 사용하여 Snowball Edge 디바이스의 물리적 인터페이스의 최대 전송 단위(MTU) 크기를 바이트 단위로 수정합니다. 이 물리적 네트워크 인터페이스와 연결된 모든 가상 네트워크 인터페이스와 직접 네트워크 인터페이스는 동일한 MTU 크기로 구성됩니다.

**Note**

최소 MTU 크기는 1,500바이트이고 최대 크기는 9,216바이트입니다.

`describe-device` 명령을 사용하여 물리적 네트워크 인터페이스 ID와 해당 인터페이스의 현재 MTU 크기를 검색할 수 있습니다. 자세한 내용은 [Snowball Edge 상태 보기](#) 단원을 참조하십시오.



`describe-direct-network-interface` 및 `describe-virtual-network-interface` 명령을 사용하여 해당 인터페이스의 현재 MTU 크기를 검색할 수 있습니다.

## 사용량

```
snowballEdge update-mtu-size --physical-network-interface-id physical-network-interface-id --mtu-size size-in-bytes --profile profile-name
```

## Example `update-mtu-size` 출력

```
{
  "PhysicalNetworkInterface": {
    "PhysicalNetworkInterfaceId": "s.ni-8c1f891d7f5b87cfe",
    "PhysicalConnectorType": "SFP_PLUS",
    "IpAddressAssignment": "DHCP",
    "IpAddress": "192.0.2.0",
    "Netmask": "255.255.255.0",
    "DefaultGateway": "192.0.2.255",
    "MacAddress": "8A:2r:5G:9p:6Q:4s",
    "MtuSize": "5743"
  }
}
```

# Snowball Edge와의 데이터 마이그레이션을 위해 Amazon S3 어댑터를 사용하여 파일 전송

다음은 Amazon S3 REST API 작업을 사용하여 AWS Snowball Edge 디바이스에 이미 있는 S3 버킷과 프로그래밍 방식으로 데이터를 주고받는 데 사용할 수 있는 Amazon S3 어댑터의 개요입니다. 이 Amazon S3 REST API 지원은 일부 작업으로 제한됩니다. 이 작업 하위 집합을 AWS SDKs 중 하나와 함께 사용하여 프로그래밍 방식으로 데이터를 전송할 수 있습니다. 또한 지원되는 AWS Command Line Interface (AWS CLI) 명령의 하위 집합을 Amazon S3에 사용하여 프로그래밍 방식으로 데이터를 전송할 수도 있습니다.

솔루션에서 AWS SDK for Java 버전 1.11.0 이상을 사용하는 경우 `S3ClientOptions` 다음을 사용해야 합니다.

- `disableChunkedEncoding()` - chunked 인코딩이 어댑터에서 지원되지 않음을 나타냅니다.
- `setPathStyleAccess(true)` - 모든 요청에 대해 경로 방식 액세스를 사용하도록 어댑터를 구성합니다.

자세한 내용은 Java용 Amazon AppStream SDK의 [클래스 S3ClientOptions.Builder](#)를 참조하세요.

## Important

한 번에 하나의 메서드만 사용하여 AWS Snowball Edge 디바이스의 로컬 버킷에 데이터를 읽고 쓰는 것이 좋습니다. 동일한 버킷에서 NFS 인터페이스와 Amazon S3 어댑터를 동시에 사용하면 읽기/쓰기 충돌이 발생할 수 있습니다.

[AWS Snowball Edge 할당량](#)에서 한도에 대해 자세히 설명합니다.

AWS 서비스가 Snowball Edge에서 제대로 작동하려면 서비스의 포트를 허용해야 합니다. 자세한 내용은 [Snowball Edge의 AWS 서비스에 대한 포트 요구 사항](#)을 참조하세요.

## 주제

- [Amazon S3 어댑터와 함께 사용할 AWS CLI 버전 1.16.14 다운로드 및 설치](#)
- [Snowball Edge 디바이스에서 AWS CLI 및 API 작업 사용](#)
- [Snowball Edge에서 로컬 Amazon S3 자격 증명 가져오기 및 사용](#)
- [Snowball Edge의 Amazon S3 어댑터에 지원되지 않는 Amazon S3 기능](#)

- [Snowball Edge로의 데이터 전송 성능을 개선하기 위해 작은 파일 일괄 처리](#)
- [Snowball Edge와의 데이터 전송에 지원되는 AWS CLI 명령](#)
- [데이터 전송을 위해 Snowball Edge에서 지원되는 Amazon S3 REST API 작업](#)

## Amazon S3 어댑터와 함께 사용할 AWS CLI 버전 1.16.14 다운로드 및 설치

현재 AWS CLI 의 버전 1.16.14 및 이전 버전만 Snowball Edge 디바이스 사용이 지원됩니다. 최신 버전의 AWS CLI 는 Amazon S3 S3 어댑터와 호환되지 않습니다.

### Note

Snowball Edge에서 Amazon S3 호환 스토리지를 사용하는 경우 최신 버전의를 사용할 수 있습니다 AWS CLI. 최신 버전을 다운로드하여 사용하려면 [AWS Command Line Interface 사용 설명서](#)를 참조하세요.

## Linux 운영 체제 AWS CLI 에 설치

다음 체인 명령을 실행합니다.

```
curl "https://s3.amazonaws.com/aws-cli/awscli-bundle-1.16.14.zip" -o "awscli-bundle.zip";unzip awscli-bundle.zip;sudo ./awscli-bundle/install -i /usr/local/aws -b /usr/local/bin/aws;/usr/local/bin/aws --version;
```

## Windows 운영 체제 AWS CLI 에 설치

운영 체제에 맞는 설치 프로그램 파일을 다운로드하고 실행합니다.

- [Python 2와 번들로 제공되는 32비트 설치 관리자](#)
- [Python 3과 번들로 제공되는 32비트 설치 관리자](#)
- [Python 2와 번들로 제공되는 64비트 설치 관리자](#)
- [Python 3과 번들로 제공되는 64비트 설치 관리자](#)

- [올바른 버전을 자동으로 설치하는 32비트 및 64비트 설치 관리자를 포함하는 설정 파일](#)

## Snowball Edge 디바이스에서 AWS CLI 및 API 작업 사용

AWS CLI 또는 API 작업을 사용하여 Snowball Edge에서 IAM, Amazon S3 및 Amazon EC2 명령을 실행할 때는 리전을 ""로 지정해야 합니다snow. 다음 예제와 같이 명령 자체 내에서 AWS configure 또는 snow를 사용하여 작업을 수행할 수 있습니다.

```
aws configure --profile abc
AWS Access Key ID [None]: AKIAIOSFODNN7EXAMPLE
AWS Secret Access Key [None]: 1234567
Default region name [None]: snow
Default output format [None]: json
```

Or

```
aws s3 ls --endpoint http://192.0.2.0:8080 --region snow --profile snowballEdge
```

## AWS Snowball Edge에 대한 Amazon S3 API 인터페이스를 사용한 권한 부여

Amazon S3 어댑터를 사용하면 기본적으로 모든 상호 작용이 AWS 서명 버전 4 알고리즘으로 서명됩니다. 이 권한 부여는 소스에서 인터페이스로 이동하는 데이터만 확인하는 데에만 사용됩니다. 모든 암호화 및 암호화 해제는 디바이스에서 이루어집니다. 암호화되지 않은 데이터는 디바이스에 저장되지 않습니다.

인터페이스를 사용할 때는 다음 사항에 유의하세요.

- 로컬 Amazon S3 자격 증명을 가져와 AWS Snowball Edge 디바이스에 대한 요청에 서명하려면 snowballEdge list-access-keys 및 snowballEdge get-secret-access-keys Snowball Edge 클라이언트 명령을 실행합니다. 자세한 내용은 [Snowball Edge 클라이언트 구성 및 사용](#) 단원을 참조하십시오. 이러한 로컬 Amazon S3 자격 증명은 키 페어(액세스 키 및 비밀 키)를 포함합니다. 이러한 키는 작업과 연결된 디바이스에 대해서만 유효합니다. (AWS Identity and Access Management IAM)이 없기 AWS 클라우드 때문에에서 사용할 수 없습니다.
- 암호화 키는 사용하는 AWS 자격 증명에 따라 변경되지 않습니다. 서명 버전 4 알고리즘을 통한 서명은 소스에서 인터페이스로 이동하는 데이터를 확인하는 데에만 사용됩니다. 따라서 이러한 서명은 Snowball에서 데이터를 암호화하는 데 사용되는 암호화 키의 요소가 아닙니다.

## Snowball Edge에서 로컬 Amazon S3 자격 증명 가져오기 및 사용

Snowball Edge와의 모든 상호 작용은 AWS 서명 버전 4 알고리즘으로 서명됩니다. 알고리즘에 대한 자세한 정보는 AWS 일반 참조의 [서명 버전 4 서명 프로세스](#)를 참조하세요.

`snowballEdge list-access-keys` 및 `snowballEdge get-secret-access-key` Snowball Edge 클라이언트 정보를 실행하여 Snowball Edge 클라이언트 Edge 디바이스에 대한 요청에 서명하기 위한 로컬 Amazon S3 자격 증명을 얻을 수 있습니다([Snowball Edge에 대한 자격 증명 가져오기](#) 참조). 이러한 로컬 Amazon S3 자격 증명은 키 페어(액세스 키 ID 및 비밀 키)를 포함합니다. 이러한 자격 증명은 작업과 연결된 디바이스에 대해서만 유효합니다. IAM이 없기 AWS 클라우드 때문에에서 사용할 수 없습니다.

이러한 자격 증명을 서버의 AWS 자격 증명 파일에 추가할 수 있습니다. 기본 자격 증명 프로필 파일은 일반적으로 `~/.aws/credentials`에 있지만, 위치는 플랫폼에 따라 다를 수 있습니다. 이 파일은 많은 AWS SDKs 및에서 공유됩니다 AWS CLI. 다음 예시와 같이 프로필 이름을 사용하여 로컬 자격 증명을 저장할 수 있습니다.

```
[snowballEdge]
aws_access_key_id = AKIAIOSFODNN7EXAMPLE
aws_secret_access_key = wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
```

## Snowball Edge에서 S3 어댑터를 엔드포인트로 AWS CLI 사용하도록 구성

AWS CLI 를 사용하여 AWS Snowball Edge 디바이스에 명령을 실행할 때 엔드포인트를 Amazon S3 어댑터로 지정합니다. 다음과 같이 HTTPS 엔드포인트 또는 보안되지 않은 HTTP 엔드포인트를 사용할 수 있습니다.

### HTTPS 보안 엔드포인트

```
aws s3 ls --endpoint https://192.0.2.0:8443 --ca-bundle path/to/certificate --profile snowballEdge
```

### HTTP 보안되지 않는 엔드포인트

```
aws s3 ls --endpoint http://192.0.2.0:8080 --profile snowballEdge
```

8443의 HTTPS 엔드포인트를 사용하는 경우 데이터가 서버에서 Snowball Edge로 안전하게 전송됩니다. 이 암호화는 Snowball Edge에서 새 IP 주소를 가져올 때마다 생성하는 인증서로 보장됩니다. 인증

서가 생기면 이를 로컬 `ca-bundle.pem` 파일에 저장할 수 있습니다. 그런 다음 다음 설명과 같이 인증서 경로를 포함하도록 AWS CLI 프로필을 구성할 수 있습니다.

인증서를 인터페이스 엔드포인트에 연결하려면

1. Snowball Edge를 전원과 네트워크에 연결한 후 켭니다.
2. 디바이스가 부팅을 마치면 로컬 네트워크상의 IP 주소를 적어 둡니다.
3. 네트워크의 터미널에서 Snowball Edge를 ping할 수 있는지 확인합니다.
4. 터미널에서 `snowballEdge get-certificate` 명령을 실행합니다. 이 명령에 대한 자세한 내용은 [Snowball Edge에서 퍼블릭 키 인증서 관리](#) 섹션을 참조하세요.
5. `snowballEdge get-certificate` 명령의 출력을 파일(예: `ca-bundle.pem`)에 저장합니다.
6. 터미널에서 다음 명령을 실행합니다.

```
aws configure set profile.snowballEdge.ca_bundle /path/to/ca-bundle.pem
```

절차를 완료하면 다음 예시와 같이 이러한 로컬 자격 증명, 인증서 및 지정된 엔드포인트로 CLI 명령을 실행할 수 있습니다.

```
aws s3 ls --endpoint https://192.0.2.0:8443 --profile snowballEdge
```

## Snowball Edge의 Amazon S3 어댑터에 지원되지 않는 Amazon S3 기능

Amazon S3 어댑터를 사용하여 Amazon S3 API 작업을 통해 프로그래밍 방식으로 Snowball Edge에서 데이터를 전송하고 수신할 수 있습니다. 하지만 일부 Amazon S3 전송 기능 및 API 작업은 Amazon S3 어댑터를 사용하여 Snowball Edge 디바이스에서 사용할 수 있도록 지원되지 않습니다. 예를 들어, 다음 기능 및 작업은 Snowball Edge에서 사용할 수 있도록 지원되지 않습니다.

- [TransferManager](#) - 이 유틸리티는 Java용 SDK를 사용하여 파일을 로컬 환경에서 Amazon S3로 전송합니다. 대신 지원되는 API 작업 또는 AWS CLI 명령을 인터페이스에서 사용하는 것을 고려해 보세요.
- [GET Bucket\(List Objects\) 버전 2](#) - 이 GET 작업의 구현은 버킷 내 객체를 일부 또는 전부(최대 1,000개) 반환합니다. [GET Bucket\(List Objects\) 버전 1](#) 작업 또는 `ls` AWS CLI 명령을 사용하는 것을 고려해 보세요.

- [ListBucket](#) - 객체 엔드포인트가 있는 ListBuckets는 지원되지 않습니다. 다음 명령은 Snowball Edge의 Amazon S3 호환 스토리지에서 작동하지 않습니다.

```
aws s3 ls --endpoint https://192.0.2.0 --profile profile
```

## Snowball Edge로의 데이터 전송 성능을 개선하기 위해 작은 파일 일괄 처리

암호화 때문에 각 복사 작업에는 일정한 오버헤드가 있어야 합니다. 작은 파일을 AWS Snowball Edge 디바이스로 전송하는 프로세스의 속도를 높이려면 단일 아카이브에서 함께 배치 처리할 수 있습니다. 파일을 함께 배치 처리할 때 지원되는 아카이브 형식 중 하나로 배치 처리가 된 경우 Amazon S3로 파일을 가져올 때 자동으로 압축이 풀릴 수 있습니다.

일반적으로 크기가 1MB 이하인 파일이 배치에 포함됩니다. 배치 안에 보유할 수 있는 파일의 수에 대한 하드 제한은 없습니다. 하지만 약 10,000개의 파일로 배치를 제한하는 것이 좋습니다. 배치에 100,000개 이상의 파일이 있는 경우 디바이스를 반환한 이후 파일을 Amazon S3로 가져오는 속도에 영향을 줄 수 있습니다. 각 배치의 총 크기는 100GB를 넘지 않는 것이 좋습니다.

파일 배치 처리는 직접 관리하는 수동 프로세스입니다. 파일을 일괄 처리한 후 `--metadata snowball-auto-extract=true` 옵션과 AWS CLI `cp` 함께 명령을 사용하여 Snowball Edge 디바이스로 전송합니다. `snowball-auto-extract=true`를 지정하면 배치 처리된 파일의 크기가 100GB를 넘지 않는 한, 데이터를 Amazon S3로 가져올 때 아카이브된 파일의 내용 압축이 자동으로 풀립니다.

### Note

100GB를 넘는 모든 배치는 Amazon S3로 가져올 때 압축이 풀리지 않습니다.

### 작은 파일을 배치 처리하려면

1. 작은 파일을 어떤 형식으로 배치 처리할지 결정합니다. 자동 압축 풀기 기능은 TAR, ZIP 및 `tar.gz` 형식을 지원합니다.
2. 함께 배치 처리하고자 하는 파일의 크기 및 전체 파일 수를 포함하여 어떤 작은 파일을 함께 배치 처리하고자 하는지 식별합니다.

### 3. 명령줄의 파일을 아래 예시와 같이 일괄 처리합니다.

- Linux의 경우 디바이스로 파일을 전송하는 데 사용되는 것과 동일한 명령줄로 파일을 배치 처리합니다.

```
tar -cf - /Logs/April | aws s3 cp - s3://amzn-s3-demo-bucket/batch01.tar --
metadata snowball-auto-extract=true --endpoint http://192.0.2.0:8080
```

#### Note

또는 원하는 아카이브 유틸리티를 사용하여 파일을 하나 이상의 큰 아카이브로 배치 처리할 수 있습니다. 그러나 이 접근 방식을 사용하려면 아카이브를 Snowball Edge로 전송하기 전에 아카이브를 저장하기 위한 추가 로컬 스토리지가 필요합니다.

- Windows의 경우 모든 파일이 명령이 실행되는 동일한 디렉터리에 있는 경우 다음 예시 명령을 사용하여 파일을 일괄 처리합니다.

```
7z a -tzip -so "test" | aws s3 cp - s3://amzn-s3-demo-bucket/batch01.zip --
metadata snowball-auto-extract=true --endpoint http://192.0.2.0:8080
```

명령이 실행되는 다른 디렉터리의 파일을 일괄 처리하려면 다음 예시 명령을 사용합니다.

```
7z a -tzip -so "test" "c:\temp" | aws s3 cp - s3://amzn-s3-demo-bucket/
batch01.zip --metadata snowball-auto-extract=true --endpoint http://10.x.x.x:8080
```

#### Note

Microsoft Windows 2016의 경우 tar를 사용할 수 없지만 Windows용 Tar 웹 사이트에서 다운로드할 수 있습니다.

7ZIP 웹 사이트에서 7 ZIP을 다운로드할 수 있습니다.

- Snowball Edge를 사용하여 Amazon S3로 전송하고자 하는 모든 작은 파일을 아카이브할 때까지 반복합니다.
- 아카이브된 파일을 Snowball로 전송합니다. 데이터를 자동으로 추출하고 1단계에서 앞서 언급한 지원되는 아카이브 형식 중 하나를 사용한 경우 --metadata snowball-auto-extract=true 옵션과 함께 명령을 사용합니다 AWS CLI cp.



**Note**

보관되지 않은 파일이 있는 경우 이 명령을 사용하면 안 됩니다.

아카이브 파일을 만들 때 추출 시 현재 데이터 구조가 유지됩니다. 즉, 파일 및 폴더가 포함된 아카이브 파일을 생성하면 Snowball Edge는 Amazon S3로의 수집 프로세스 중에 이 파일을 다시 생성합니다.

아카이브 파일은 파일이 저장되어 있는 동일한 디렉터리에 추출되며 그에 따라 폴더 구조가 구축됩니다. 아카이브 파일을 복사할 때는 `--metadata snowball-auto-extract=true` 플래그를 설정하는 것이 중요하다는 점을 명심하세요. 그렇지 않으면 Amazon S3로 데이터를 가져올 때 Snowball Edge는 데이터를 추출하지 않습니다.

3단계의 예시를 사용하여, `/Logs/April/`의 폴더 구조에 파일 `a.txt`, `b.txt`, `c.txt` 등이 들어 있는 경우 이 아카이브 파일을 `/amzn-s3-demo-bucket/`의 루트에 배치한 경우 추출 후 데이터는 다음과 같이 표시됩니다.

```
/amzn-s3-demo-bucket/Logs/April/a.txt
/amzn-s3-demo-bucket/Logs/April/b.txt
/amzn-s3-demo-bucket/Logs/April/c.txt
```

아카이브 파일을 `/amzn-s3-demo-bucket/Test/`에 배치한 경우 추출은 다음과 같이 표시됩니다.

```
/amzn-s3-demo-bucket/Test/Logs/April/a.txt
/amzn-s3-demo-bucket/Test/Logs/April/b.txt
/amzn-s3-demo-bucket/Test/Logs/April/c.txt
```

## Snowball Edge와의 데이터 전송에 지원되는 AWS CLI 명령

아래에서 Snowball Edge의 Amazon S3 어댑터 또는 Amazon S3 호환 스토리지를 해당 AWS Command Line Interface (AWS CLI) 명령의 엔드포인트로 지정하는 방법에 대한 정보를 찾을 수 있습니다. 어댑터 또는 Snowball Edge의 Amazon S3 호환 스토리지를 사용하여 AWS Snowball Edge 디바이스로 데이터를 전송하는 데 지원되는 Amazon S3에 대한 AWS CLI 명령 목록을 찾을 수도 있습니다.

**Note**

AWS CLI 호출할 리전 지정 AWS CLI을 포함하여 설치 및 설정에 대한 자세한 내용은 [AWS Command Line Interface 사용 설명서](#)를 참조하세요.

현재 Snowball Edge 디바이스는 Amazon S3 어댑터를 사용할 경우 버전 1.16.14 이하 버전만 지원합니다. [Snowball Edge 클라이언트 버전 찾기](#)을(를) 참조하세요. Snowball Edge에서 Amazon S3 호환 스토리지를 사용하는 경우 최신 버전의를 사용할 수 있습니다 AWS CLI. 최신 버전을 다운로드하여 사용하려면 [AWS Command Line Interface 사용 설명서](#)를 참조하세요.

**Note**

AWS CLI버전 1.16.14를 설치하기 전에 Python 2.6.5+ 또는 3.4+ 버전을 설치해야 합니다.

## Amazon S3 및 Snowball Edge를 사용한 데이터 전송에 지원되는 AWS CLI 명령

다음은 AWS Snowball Edge 디바이스가 지원하는 Amazon S3에 대한 AWS CLI 명령 및 옵션의 하위 집합에 대한 설명입니다. 나열되지 않은 명령이나 옵션은 지원되지 않습니다. `--sse`, `--storage-class` 등과 같은 일부 지원되지 않는 옵션을 명령과 함께 선언할 수 있습니다. 하지만 이러한 옵션은 무시되며 데이터를 가져오는 방식에 영향을 미치지 않습니다.

- `cp` - AWS Snowball Edge 디바이스에서 파일 또는 객체를 복사합니다. 다음은 이 명령에 사용되는 옵션입니다.
  - `--dryrun(부울)` - 지정된 명령을 사용하여 수행될 작업이 실행되지 않고 표시됩니다.
  - `--quiet(부울)` - 지정된 명령에 의해 수행되는 작업이 표시되지 않습니다.
  - `--include(문자열)` - 지정한 패턴과 일치하는 파일 또는 객체를 명령에서 제외하지 않습니다. 자세한 내용은 AWS CLI 명령 참조의 [제외 및 포함 필터 사용](#)을 참조하세요.
  - `--exclude(문자열)` - 지정한 패턴과 일치하는 모든 파일 또는 객체를 명령에서 제외합니다.
  - `--follow-symlinks` | `--no-follow-symlinks(부울)` - 로컬 파일 시스템에서 Amazon S3로 업로드하는 경우에만 심볼 링크(symlinks)를 따릅니다. Amazon S3는 심볼 링크를 지원하지 않으므로 링크 대상의 콘텐츠는 링크 이름 아래에 업로드됩니다. 두 가지 옵션을 모두 지정하지 않는 경우 기본값은 symlink를 따르는 것입니다.
  - `--only-show-errors(부울)` - 오류와 경고만 표시됩니다. 다른 모든 출력은 억제됩니다.

- `--recursive`(부울) - 지정된 디렉터리 또는 접두사 아래의 모든 파일 또는 객체에 대해 명령이 수행됩니다.
- `--page-size`(정수) - 나열 작업에 대한 각 응답으로 반환할 결과 수입니다. 기본값은 1,000(허용되는 최대값)입니다. 작업 시간이 초과된 경우 더 낮은 값을 사용하면 도움이 될 수 있습니다.
- `--metadata`(맵) - Amazon S3에 객체와 함께 저장할 메타데이터 맵입니다. 이 맵은 이 요청에 속한 모든 객체에 적용됩니다. 동기화에서 이 기능은 변경되지 않은 파일은 새 메타데이터를 받지 않음을 뜻합니다. 두 Amazon S3 위치 간에 복사할 때 별도로 지정하지 않은 경우 `metadata-directive` 인수의 기본값은 REPLACE입니다.
- `ls` - AWS Snowball Edge 디바이스의 객체를 나열합니다. 다음은 이 명령에 사용되는 옵션입니다.
  - `--human-readable`(부울) - 파일 크기가 사람이 읽을 수 있는 형식으로 표시됩니다.
  - `--summarize`(부울) - 요약 정보가 표시됩니다. 이 정보는 객체의 수와 총 크기입니다.
  - `--recursive`(부울) - 지정된 디렉터리 또는 접두사 아래의 모든 파일 또는 객체에 대해 명령이 수행됩니다.
  - `--page-size`(정수) - 나열 작업에 대한 각 응답으로 반환할 결과 수입니다. 기본값은 1,000(허용되는 최대값)입니다. 작업 시간이 초과된 경우 더 낮은 값을 사용하면 도움이 될 수 있습니다.
- `rm` - AWS Snowball Edge 디바이스에서 객체를 삭제합니다. 다음은 이 명령에 사용되는 옵션입니다.
  - `--dryrun`(부울) - 지정된 명령을 사용하여 수행될 작업이 실행되지 않고 표시됩니다.
  - `--include`(문자열) - 지정한 패턴과 일치하는 파일 또는 객체를 명령에서 제외하지 않습니다. 자세한 내용은 AWS CLI 명령 참조의 [제외 및 포함 필터 사용](#)을 참조하세요.
  - `--exclude`(문자열) - 지정한 패턴과 일치하는 모든 파일 또는 객체를 명령에서 제외합니다.
  - `--recursive`(부울) - 지정된 디렉터리 또는 접두사 아래의 모든 파일 또는 객체에 대해 명령이 수행됩니다.
  - `--page-size`(정수) - 나열 작업에 대한 각 응답으로 반환할 결과 수입니다. 기본값은 1,000(허용되는 최대값)입니다. 작업 시간이 초과된 경우 더 낮은 값을 사용하면 도움이 될 수 있습니다.
  - `--only-show-errors`(부울) - 오류와 경고만 표시됩니다. 다른 모든 출력은 억제됩니다.
  - `--quiet`(부울) - 지정된 명령에 의해 수행되는 작업이 표시되지 않습니다.
- `sync` - 디렉터리와 접두사를 동기화합니다. 이 명령은 소스 디렉터리의 새 파일 및 업데이트된 파일을 대상에 복사합니다. 이 명령은 하나 이상의 파일이 포함된 경우에만 대상에 디렉터를 만듭니다.

### Important

동일한 Snowball Edge에 있는 디렉터리끼리 동기화하는 것은 지원되지 않습니다.

한 AWS Snowball Edge 디바이스에서 다른 AWS Snowball Edge 디바이스로의 동기화는 지원되지 않습니다.

이 옵션은 온프레미스 데이터 스토리지와 Snowball Edge 사이의 콘텐츠를 동기화하는 데만 사용할 수 있습니다.

- `--dryrun(부울)` - 지정된 명령을 사용하여 수행될 작업이 실행되지 않고 표시됩니다.
- `--quiet(부울)` - 지정된 명령에 의해 수행되는 작업이 표시되지 않습니다.
- `--include(문자열)` - 지정한 패턴과 일치하는 파일 또는 객체를 명령에서 제외하지 않습니다. 자세한 내용은 AWS CLI 명령 참조의 [제외 및 포함 필터 사용](#)을 참조하세요.
- `--exclude(문자열)` - 지정한 패턴과 일치하는 모든 파일 또는 객체를 명령에서 제외합니다.
- `--follow-symlinks` 또는 `--no-follow-symlinks(부울)` - 로컬 파일 시스템에서 Amazon S3로 업로드하는 경우에만 심볼 링크(symlinks)를 따릅니다. Amazon S3는 심볼 링크를 지원하지 않으므로 링크 대상의 콘텐츠는 링크 이름 아래에 업로드됩니다. 두 가지 옵션을 모두 지정하지 않는 경우 기본값은 symlink를 따르는 것입니다.
- `--only-show-errors(부울)` - 오류와 경고만 표시됩니다. 다른 모든 출력은 억제됩니다.
- `--no-progress(부울)` - 파일 전송 진행 상황이 표시되지 않습니다. 이 옵션은 `--quiet` 및 `--only-show-errors` 옵션을 입력하지 않는 경우에 한해 적용됩니다.
- `--page-size(정수)` - 나열 작업에 대한 각 응답으로 반환할 결과 수입니다. 기본값은 1,000(허용되는 최대값)입니다. 작업 시간이 초과된 경우 더 낮은 값을 사용하면 도움이 될 수 있습니다.
- `--metadata(맵)` - Amazon S3에 객체와 함께 저장할 메타데이터 맵입니다. 이 맵은 이 요청에 속한 모든 객체에 적용됩니다. 동기화에서 이 기능은 변경되지 않은 파일은 새 메타데이터를 받지 않음을 뜻합니다. 두 Amazon S3 위치 간에 복사할 때 별도로 지정하지 않은 경우 `metadata-directive` 인수의 기본값은 REPLACE입니다.

#### Important

동일한 Snowball Edge에 있는 디렉터리끼리 동기화하는 것은 지원되지 않습니다.

한 AWS Snowball Edge 디바이스에서 다른 AWS Snowball Edge 디바이스로의 동기화는 지원되지 않습니다.

이 옵션은 온프레미스 데이터 스토리지와 Snowball Edge 사이의 콘텐츠를 동기화하는 데만 사용할 수 있습니다.

- `--size-only(부울)` - 이 옵션에서 각 키의 크기는 원본에서 대상으로 동기화할지 여부를 결정하는 데 사용되는 유일한 조건입니다.

- `--exact-timestamps`(부울) - Amazon S3에서 로컬 스토리지로 동기화할 때 타임스탬프가 정확히 일치하는 경우에만 같은 크기의 항목이 무시됩니다. 로컬 버전이 Amazon S3 버전보다 최신이 아닐 경우 기본 동작은 같은 크기의 항목을 무시하는 것입니다.
- `--delete`(부울) - 대상에는 있지만 원본에는 없는 파일이 동기화 중에 삭제됩니다.

이름에 공백이 있는 파일 또는 폴더(예: my photo.jpg 또는 My Documents)로 작업할 수 있습니다. 그러나 AWS CLI 명령에서 공백을 올바르게 처리해야 합니다. 자세한 내용은 AWS Command Line Interface 사용 설명서의 [AWS CLI의 파라미터 값 지정](#)을 참조하세요.

## 데이터 전송을 위해 Snowball Edge에서 지원되는 Amazon S3 REST API 작업

다음에는 Amazon S3 어댑터 사용 시 지원되는 Amazon S3 REST API 작업 목록이 나와 있습니다. 목록에는 API 작업이 Amazon S3와 연동하는 방식에 대한 정보를 제공하는 링크가 포함되어 있습니다. 또한 Amazon S3 API 작업과 AWS Snowball Edge 디바이스 작업 간의 동작 차이도 다릅니다. 다음 예시와 같이 AWS Snowball Edge 에 디바이스에서 반환되는 모든 응답에서는 Server를 AWSSnowball로 선언합니다.

```
HTTP/1.1 201 OK
x-amz-id-2: JuKZqmXuiwFeDQxhD7M8KtsKobSzwA1QEjLbTMTagkKdBX2z7I1/jGhDeJ3j6s80
x-amz-request-id: 32FE2CEB32F5EE25
Date: Fri, 08 2016 21:34:56 GMT
Server: AWSSnowball
```

Amazon S3 REST API 호출에는 SigV4 서명이 필요합니다. AWS CLI 또는 AWS SDK를 사용하여 이러한 API 호출을 수행하는 경우 SigV4 서명이 처리됩니다. 그렇지 않은 경우 자체 SigV4 서명 솔루션을 구현해야 합니다. 자세한 내용은 Amazon Simple Storage Service 사용 설명서의 [요청 인증\(AWS 서명 버전 4\)](#)을 참조하세요.

- [GET Bucket\(List Objects\)버전 1](#) - 지원됩니다. 하지만 이 GET 작업 구현에서는 다음이 지원되지 않습니다.
  - 페이지 매김
  - 마커
  - 구분 기호
  - 목록이 반환될 때 목록이 정렬되지 않습니다.

버전 1만 지원됩니다. GET Bucket(List Objects) 버전 2는 지원되지 않습니다.

- [GET Service](#)
- [HEAD Bucket](#)
- [HEAD Object](#)
- [GET Object](#) - Snow 디바이스의 S3 버킷에서 객체를 다운로드하는 것입니다.
- [PUT Object](#) -를 사용하여 객체를 AWS Snowball Edge 디바이스에 업로드하면 PUT Object ETag가 생성됩니다.

ETag는 객체의 해시입니다. ETag는 객체의 콘텐츠에 대한 변경 사항만 반영하고 메타데이터에 대한 변경을 반영하지 않습니다. ETag는 객체 데이터의 MD5 다이제스트일 수도 아닐 수도 있습니다. 객체 복사에 대한 자세한 내용은 Amazon Simple Storage Service API 참조의 [공통 응답 헤더](#)를 참조하세요.

- [DELETE Object](#)
- [멀티파트 업로드 시작](#) -이 구현에서는 AWS Snowball Edge 디바이스에 이미 있는 객체에 대한 멀티파트 업로드 요청을 시작하면 먼저 해당 객체가 삭제됩니다. 그런 다음 AWS Snowball Edge 디바이스에 여러 부분으로 복사합니다.
- [멀티파트 업로드 목록 조회](#)
- [부분 업로드](#)
- [멀티파트 업로드 완료](#)
- [멀티파트 업로드 중단](#)

#### Note

이 목록에 없는 Amazon S3 어댑터 REST API 작업은 지원되지 않습니다. 지원되지 않는 REST API 작업을 Snowball Edge에 사용하면 해당 작업이 지원되지 않는다는 오류 메시지가 반환됩니다.

## Snowball Edge에서 NFS 인터페이스 관리

NFS(Network File System) 인터페이스를 사용하여 디바이스가 운영 체제에 로컬 스토리지인 것처럼 Snowball Edge에 파일을 업로드합니다. 이렇게 하면 파일 복사, 드래그 앤 드롭 또는 기타 그래픽 사용자 인터페이스 기능과 같은 운영 체제의 기능을 사용할 수 있으므로 데이터를 보다 사용자 친화적으로 전송할 수 있습니다. 디바이스의 각 S3 버킷은 NFS 인터페이스 엔드포인트로 사용할 수 있으며 데이터를 복사하도록 탑재할 수 있습니다. NFS 인터페이스는 가져오기 작업에 사용할 수 있습니다.

디바이스 주문 작업이 생성될 때 Snowball Edge 디바이스가 포함하도록 구성된 경우 NFS 인터페이스를 사용할 수 있습니다. 디바이스가 NFS 인터페이스를 포함하도록 구성되지 않은 경우 S3 Edge에서 S3 어댑터 또는 Amazon S3 호환 스토리지를 사용하여 데이터를 전송합니다. S3 어댑터에 대한 자세한 내용은 [를 사용하여 Amazon S3 어댑터 스토리지 관리 AWS OpsHub](#) 섹션을 참조하세요. Snowball Edge의 Amazon S3 호환 스토리지에 대한 자세한 내용은 섹션을 참조하세요 [를 사용하여 Snowball Edge에서 Amazon S3 호환 스토리지 설정 AWS OpsHub](#).

시작되면 NFS 인터페이스가 1GB의 메모리와 1개의 CPU를 사용합니다. 이로 인해 Snowball Edge에서 실행되는 다른 서비스의 수 또는 실행할 수 있는 EC2-compatible 인스턴스의 수가 제한될 수 있습니다.

NFS 인터페이스를 통해 전송되는 데이터는 전송 중에 암호화되지 않습니다. NFS 인터페이스를 구성할 때 CIDR 블록을 제공할 수 있으며 Snowball Edge는 해당 블록에 주소가 있는 클라이언트 컴퓨터에서 NFS 인터페이스에 대한 액세스를 제한합니다.

디바이스의 파일은 AWS로 반환되면 Amazon S3로 전송됩니다. 자세한 내용은 [Amazon S3로 작업 가져오기 Snowball Edge 작동](#) 참조하세요.

컴퓨터 운영 체제에서 NFS를 사용하는 방법에 대한 자세한 내용은 운영 체제 설명서를 참조하세요.

NFS 인터페이스를 사용할 때는 다음 사항에 유의하세요.

- NFS 인터페이스는 디바이스의 데이터 스토리지를 위한 로컬 버킷을 제공합니다. 가져오기 작업의 경우 로컬 버킷의 데이터를 Amazon S3로 가져오지 않습니다.
- 파일 이름은 Snowball Edge의 로컬 S3 버킷에 있는 객체 키입니다. 키 이름은 최대 1,024바이트 길이와 UTF-8 인코딩을 사용하는 유니코드 문자의 시퀀스입니다. 가능한 경우 NFSv4.1을 사용하고 파일 이름을 유니코드 UTF-8 형식으로 인코딩하여 데이터를 성공적으로 가져오는 것이 좋습니다. UTF-8 인코딩으로 인코딩되지 않은 파일 이름은 사용하는 NFS 인코딩에 따라 S3에 업로드되지 않거나 다른 파일 이름으로 S3에 업로드될 수 있습니다.
- 파일 경로의 최대 길이는 1,024자 미만이어야 합니다. Snowball Edge는 1,024자를 초과하는 파일 경로를 지원하지 않습니다. 이 파일 경로 길이를 초과하면 파일 가져오기 오류가 발생합니다.



- 자세한 내용은 Amazon Simple Storage Service 사용 설명서의 [객체 키](#)를 참조하세요.
- NFS 기반 전송의 경우 Snowball Edge에서 Amazon S3로 가져올 때 표준 POSIX 스타일 메타데이터가 객체에 추가됩니다. 또한 현재 NFS 옵션을 사용하여 Amazon S3 for Snowball Edge로 내부 가져오기 메커니즘의 일부로 사용 AWS DataSync 하므로 메타데이터 "x-amz-meta-user-agent aws-datasync"가 표시됩니다.
- 단일 Snowball Edge 디바이스를 사용하여 최대 4천만 개의 파일을 전송할 수 있습니다. 단일 작업에서 4천만 개 이상의 파일을 전송해야 하는 경우 파일을 일괄 처리하여 전송할 때마다 파일 수를 줄이십시오. 향상된 NFS 인터페이스 또는 S3 인터페이스를 사용하는 Snowball Edge 디바이스의 경우 개별 파일 크기는 최대 5TB까지 원하는 크기일 수 있습니다.

AWS OpsHub GUI 도구를 사용하여 NFS 인터페이스를 구성하고 관리할 수도 있습니다. 자세한 내용은 [NFS 인터페이스 관리](#)을 참조하세요.

## Snowball Edge에 대한 NFS 구성

NFS 인터페이스는 기본적으로 Snowball Edge 디바이스에서 실행되지 않으므로 디바이스로 데이터를 전송하려면 NFS 인터페이스를 시작해야 합니다. Snowball Edge에서 실행되는 가상 네트워크 인터페이스(VNI)의 IP 주소를 제공하고 필요한 경우 파일 공유에 대한 액세스를 제한하여 NFS 인터페이스를 구성할 수 있습니다. NFS 인터페이스를 구성하기 전에 Snowball Edge에서 가상 네트워크 인터페이스(VNI)를 설정합니다. 자세한 내용은 [컴퓨팅 인스턴스를 위한 네트워크 구성](#)을 참조하세요.

### NFS 인터페이스에 대한 Snowball Edge 구성

- `describe-service` 명령을 사용하여 NFS 인터페이스가 활성 상태인지 확인합니다.

```
snowballEdge describe-service --service-id nfs
```

명령은 NFS 서비스, ACTIVE 또는 INACTIVE의 상태를 반환합니다.

```
{
  "ServiceId" : "nfs",
  "Status" : {
    "State" : "ACTIVE"
  }
}
```



State 이름의 값이 IN인 경우 ACTIVENFS 인터페이스 서비스가 활성 상태이며 Snowball Edge NFS 볼륨을 탑재할 수 있습니다. 자세한 내용은

[NFS 인터페이스가 시작된 후 클라이언트 컴퓨터에 엔드포인트를 로컬 스토리지로 탑재합니다.](#)

[다음은 Windows, Linux 및 macOS 운영 체제에 대한 기본 탑재 명령입니다.](#)

- Windows:

```
mount -o nolock rsize=128 wsize=128 mtype=hard nfs-interface-ip-address:/  
buckets/BucketName *
```

- Linux:

```
mount -t nfs nfs-interface-ip-address:/buckets/BucketName mount_point
```

- macOS:

```
mount -t nfs -o vers=3,rsize=131072,wsize=131072,nolocks,hard,retrans=2 nfs-  
interface-ip-address:/buckets/$bucketname mount_point
```

단원을 참조하십시오. 값이 INACTIVE인 경우 서비스를 시작해야 합니다.

## Snowball Edge에서 NFS 서비스 시작

필요한 경우 가상 네트워크 인터페이스(VNI)를 시작한 다음 Snowball Edge에서 NFS 서비스를 시작합니다. 필요한 경우 NFS 서비스를 시작할 때 허용되는 네트워크 주소 블록을 제공합니다. 주소를 제공하지 않으면 NFS 엔드포인트에 대한 액세스가 제한되지 않습니다.

1. `describe-virtual-network-interface` 명령을 사용하여 Snowball Edge에서 사용 가능한 VNIs를 확인합니다.

```
snowballEdge describe-virtual-network-interfaces
```

Snowball Edge에서 하나 이상의 VNIs가 활성 상태인 경우 명령은 다음을 반환합니다.

```
snowballEdge describe-virtual-network-interfaces
[
  {
    "VirtualNetworkInterfaceArn" : "arn:aws:snowball-device::interface/
s.ni-8EXAMPLE8EXAMPLE8",
    "PhysicalNetworkInterfaceId" : "s.ni-8EXAMPLEaEXAMPLEd",
    "IpAddressAssignment" : "DHCP",
    "IpAddress" : "192.0.2.0",
    "Netmask" : "255.255.255.0",
    "DefaultGateway" : "192.0.2.1",
    "MacAddress" : "EX:AM:PL:E1:23:45"
  },{
    "VirtualNetworkInterfaceArn" : "arn:aws:snowball-device::interface/
s.ni-1EXAMPLE1EXAMPLE1",
    "PhysicalNetworkInterfaceId" : "s.ni-8EXAMPLEaEXAMPLEd",
    "IpAddressAssignment" : "DHCP",
    "IpAddress" : "192.0.2.2",
    "Netmask" : "255.255.255.0",
    "DefaultGateway" : "192.0.2.1",
    "MacAddress" : "12:34:5E:XA:MP:LE"
  }
]
```

NFS 인터페이스에 사용할 VNI의 VirtualNetworkInterfaceArn 이름 값을 기록해 둡니다.

2. 사용 가능한 VNI가 없는 경우 create-virtual-network-interface 명령을 사용하여 NFS 인터페이스에 대한 VNI를 생성합니다. 자세한 내용은 [가상 네트워크 인터페이스\(VNI\) 설정](#)을 참조하세요.
3. start-service 명령을 사용하여 NFS 서비스를 시작하고 이를 VNI와 연결합니다. NFS 인터페이스에 대한 액세스를 제한하려면 service-configuration 및 AllowedHosts 파라미터를 명령에 포함합니다.

```
snowballEdge start-service --virtual-network-interface-arns arn-of-vni --service-id
nfs --service-configuration AllowedHosts=CIDR-address-range
```

4. `describe-service` 명령을 사용하여 서비스 상태를 확인합니다. State 이름 값이 ACTIVE인 경우 실행 중입니다.

```
snowballEdge describe-service --service-id nfs
```

이 명령은 서비스 상태와 NFS 엔드포인트의 IP 주소 및 포트 번호, 엔드포인트에 액세스할 수 있는 CIDR 범위를 반환합니다.

```
{
  "ServiceId" : "nfs",
  "Status" : {
    "State" : "ACTIVE"
  },
  "Endpoints" : [ {
    "Protocol" : "nfs",
    "Port" : 2049,
    "Host" : "192.0.2.0"
  } ],
  "ServiceConfiguration" : {
    "AllowedHosts" : [ "10.24.34.0/23", "198.51.100.0/24" ]
  }
}
```

## 클라이언트 컴퓨터에 NFS 엔드포인트 탑재

NFS 인터페이스가 시작된 후 클라이언트 컴퓨터에 엔드포인트를 로컬 스토리지로 탑재합니다.

다음은 Windows, Linux 및 macOS 운영 체제에 대한 기본 탑재 명령입니다.

- Windows:

```
mount -o nolock rsize=128 wsize=128 mtype=hard nfs-interface-ip-address:/  
buckets/BucketName *
```

- Linux:

```
mount -t nfs nfs-interface-ip-address:/buckets/BucketName mount_point
```

- macOS:

```
mount -t nfs -o vers=3,rsize=131072,wsize=131072,nolocks,hard,retrans=2 nfs-  
interface-ip-address:/buckets/$bucketname mount_point
```

## Snowball Edge에서 NFS 인터페이스 중지

NFS 인터페이스를 통해 파일 전송을 마치고 Snowball Edge의 전원을 끄기 전에 `stop-service` 명령을 사용하여 NFS 서비스를 중지합니다.

```
snowballEdge stop-service --service-id nfs
```

# Snowball Edge에서 Amazon EC2-compatible 컴퓨팅 인스턴스 사용

Snowball Edge에 호스팅된 Amazon EC2 호환 컴퓨팅 인스턴스를 sbe1, sbe-c 및 sbe-g 인스턴스 유형으로 실행할 수 있습니다. sbe1 인스턴스 유형은 Snowball Edge 스토리지 최적화 옵션으로 디바이스에서 작동합니다. sbe-c 인스턴스 유형은 Snowball Edge 컴퓨팅 최적화 옵션으로 디바이스에서 작동합니다. 지원되는 인스턴스 유형의 목록은 [Snowball Edge 디바이스에서 컴퓨팅 인스턴스 사용 시 할당량](#) 섹션을 참조하세요.

Snowball Edge 디바이스 옵션용으로 지원되는 세 가지 컴퓨팅 인스턴스 유형은 모두 Snowball Edge 디바이스에 대해 고유합니다. 이에 상응하는 클라우드 기반 인스턴스와 마찬가지로 이들 인스턴스 역시 시작하려면 Amazon Machine Image(AMI)가 필요합니다. Snowball Edge 작업을 생성하기 전에 AMI를 클라우드에서 인스턴스의 기본 이미지로 선택합니다.

Snowball Edge에서 컴퓨팅 인스턴스를 사용하려면 작업을 생성하여 Snowball Edge 디바이스를 주문하고 AMIs를 지정합니다. [AWS Snow 패밀리 관리 콘솔](#), AWS CLI 또는 AWS SDKs. 일반적으로, 인스턴스를 사용하기 위해 작업을 생성하기 전에 수행해야 하는 정리 작업 사전 조건이 있습니다.

디바이스가 도착한 후 AMI 및 인스턴스 관리를 시작할 수 있습니다. Amazon EC2 호환 엔드포인트를 통해 Snowball Edge에서 컴퓨팅 인스턴스를 관리할 수 있습니다. 이러한 유형의 엔드포인트는 AWS SDK에 관련된 많은 Amazon EC2 호환 CLI 명령 및 작업을 지원합니다. Snowball Edge AWS Management Console 의를 사용하여 AMIs 및 컴퓨팅 인스턴스를 관리할 수 없습니다.

디바이스 사용을 마치면 로 반환합니다 AWS. 디바이스가 가져오기 작업에 사용되었다면 Amazon S3 어댑터 또는 NFS 인터페이스를 사용해 전송된 데이터는 Amazon S3으로 가져옵니다. 그렇지 않으면 디바이스가 반환될 때 디바이스를 완전히 삭제합니다 AWS. 이 삭제는 NIST(National Institute of Standards and Technology) 800-88 표준에 따른 것입니다.

## Important

- Snowball Edge 디바이스에서 암호화된 AMIs 사용은 지원되지 않습니다.
- Snowball Edge에서 실행되는 컴퓨팅 인스턴스의 데이터는 로 가져오지 않습니다 AWS.

## 주제

- [Snowball Edge에서 Amazon EC2와 Amazon EC2-compatible 인스턴스의 차이점](#)
- [Snowball Edge의 컴퓨팅 인스턴스 요금](#)

- [Snowball Edge에서 Amazon EC2-compatible AMI 사용](#)
- [Snowball Edge 디바이스로 가상 머신 이미지 가져오기](#)
- [Snowball Edge 디바이스에서 AWS CLI 및 API 작업 사용](#)
- [Snowball Edge의 컴퓨팅 인스턴스에 대한 네트워크 구성](#)
- [SSH를 사용하여 Snowball Edge의 컴퓨팅 인스턴스에 연결](#)
- [동일한 Snowball Edge의 EC2-compatible 컴퓨팅 인스턴스에서 S3 버킷으로 데이터 전송](#)
- [EC2-compatible 인스턴스 자동 시작](#)
- [Snowball Edge에서 Amazon EC2-compatible 엔드포인트 사용](#)
- [Snowball Edge에서 시작 템플릿을 사용하여 EC2-compatible 인스턴스 자동 시작](#)
- [Snowball Edge에서 Amazon EC2-compatible 인스턴스와 함께 Snow용 인스턴스 메타데이터 서비스 사용](#)
- [Snowball Edge에서 Amazon EC2-compatible 인스턴스와 함께 블록 스토리지 사용](#)
- [Snowball Edge의 보안 그룹을 사용하여 네트워크 트래픽 제어](#)
- [Snowball Edge에서 지원되는 EC2-compatible 인스턴스 메타데이터 및 사용자 데이터](#)
- [Snowball Edge에서 실행 중인 EC2-compatible 인스턴스 중지](#)

## Snowball Edge에서 Amazon EC2와 Amazon EC2-compatible 인스턴스의 차이점

AWS Snowball Edge EC2-compatible 인스턴스를 사용하면 고객이 EC2-compatible AMIs, EC2 APIs

## Snowball Edge의 컴퓨팅 인스턴스 요금

컴퓨팅 인스턴스 사용 시에는 추가 비용이 발생합니다. 자세한 내용은 [AWS Snowball Edge 요금](#)을 참조하세요.

## Snowball Edge에서 Amazon EC2-compatible AMI 사용

AWS Snowball Edge 디바이스에서 Amazon Machine Image(AMI)를 사용하려면 먼저 디바이스에 추가해야 합니다. AMI 추가 방법은 다음과 같습니다.

- 디바이스를 주문할 때 AMI를 업로드합니다.
- 디바이스가 현장에 도착하면 AMI를 추가합니다.

Snowball Edge와 함께 제공되는 Amazon EC2 컴퓨팅 인스턴스는 디바이스에 추가하는 Amazon EC2 AMIs를 기반으로 시작됩니다. Amazon EC2 호환 AMI는 Linux와 Microsoft Windows 운영 체제를 모두 지원합니다.

## Linux

지원되는 Linux 운영 체제는 다음과 같습니다.

- [Amazon Linux 2 for Snowball Edge](#)

### Note

이 AMI의 최신 버전은 Snowball Edge가 배송을 준비할 때 제공됩니다 AWS. 디바이스에서 이 AMI의 버전을 수신할 때 확인하려면 [Snowball Edge용 Amazon Linux 2 AMI 버전 확인](#) 섹션을 참조하세요.

- [CentOS 7 \(x86\\_64\) - HVM 업데이트 포함](#)
- Ubuntu 16.04 LTS - Xenial (HVM)

### Note

Ubuntu 16.04 LTS - Xenial(HVM) 이미지는에서 더 이상 지원되지 AWS Marketplace않지만 Amazon EC2 VM Import/Export를 통해 Snowball Edge 디바이스에서 사용할 수 있으며 AMIs.

- [Ubuntu 20.04 LTS - Focal](#)
- [Ubuntu 22.04 LTS - Jammy](#)

보안을 위한 모범 사례로 새로운 Amazon Linux 2 AMIs가 출시되면 Snowball Edge에서 Amazon Linux 2 AMIs up-to-date 유지하세요. [Snowball Edge에서 Amazon Linux 2 AMIs 업데이트](#)을(를) 참조하세요.

## Windows

지원되는 Windows 운영 체제는 다음과 같습니다.

- Windows Server 2012 R2
- Windows Server 2016
- Windows Server 2019

VM Import/Export를 AWS 사용하여 로 Windows AMIs를 추가할 수 있습니다. 아니면 디바이스를 현장에 배포한 후 바로 이미지를 디바이스로 가져올 수 있습니다. 자세한 내용은 [Snowball Edge에 Microsoft Windows AMI 추가](#) 단원을 참조하십시오.

#### Note

에서 시작된 Windows AMIs는 디바이스에 추가할 AWS 수 없습니다.  
UEFI는 지원되지 않으므로 로컬에서 가져온 AMI는 BIOS 부팅 모드여야 합니다.

Snowball Edge는 기존 보유 라이선스 사용(BYOL) 모델을 지원합니다. 자세한 내용은 [Snowball Edge에 Microsoft Windows AMI 추가](#) 단원을 참조하십시오.

#### Note

AWS Snowball Edge EC2-compatible 인스턴스를 사용하면 고객이 EC2-compatible AMIs.  
EC2 APIs

## 주제

- [Snowball Edge 주문 작업을 생성할 때 AMI 추가](#)
- [에서 Snowball Edge AWS Marketplace 에 AMI 추가](#)
- [디바이스 수신 후 Snowball Edge에 AMI 추가](#)
- [Snowball Edge에 Microsoft Windows AMI 추가](#)
- [Snowball Edge로 VM 이미지 가져오기](#)
- [Snowball Edge용 최신 Amazon Linux 2 AMI 내보내기](#)

## Snowball Edge 주문 작업을 생성할 때 AMI 추가

디바이스를 주문하면 AWS Snow 패밀리 관리 콘솔의 Compute using EC2 instances - optional 섹션에서 AMI를 선택하여 디바이스에 AMI를 추가할 수 있습니다. Compute using EC2 instances - optional에는 디바이스에 로드할 수 있는 모든 AMI가 나열되어 있습니다. 이러한 AMI는 다음 범주로 분류됩니다.

- AWS Marketplace의 AMIs- 지원되는 AMIs 목록에서 생성된 AMIs. AWS Marketplace에서 지원되는 AMI에서 AMIs [에서 Snowball Edge AWS Marketplace 에 AMI 추가](#).



- VM Import를 사용하여 업로드한 AMI - 디바이스를 주문하면 VM Import를 사용하여 업로드한 AMI가 콘솔에 나열됩니다. 자세한 내용은 VM Import/Export 사용 설명서에서 [VM Import/Export를 사용하여 VM을 이미지로 가져오기](#)를 참조하세요. 지원되는 가상화 환경에 대한 자세한 내용은 [VM Import/Export Requirements](#) 섹션을 참조하세요.

## 에서 Snowball Edge AWS Marketplace 에 AMI 추가

AWS Marketplace 인스턴스 AWS Marketplace 를 시작하고, 인스턴스에서 AMIs를 생성하고, Snow 디바이스를 주문할 리전과 동일한 리전에서 AMI를 구성하여 여러 AMI를 Snowball Edge 디바이스에 추가할 수 있습니다. 그런 다음 디바이스 주문 작업을 생성할 때 디바이스에 AMI를 포함하도록 선택할 수 있습니다. Marketplace에서 AMI를 선택할 때 지원되는 제품 코드와 플랫폼이 있는지 확인합니다.

### 주제

- [Snowball Edge용 AWS Marketplace AMIs의 제품 코드 및 플랫폼 세부 정보 확인](#)
- [Snowball Edge용 Amazon Linux 2 AMI 버전 확인](#)
- [Snowball Edge 디바이스의 AMI 구성](#)

## Snowball Edge용 AWS Marketplace AMIs의 제품 코드 및 플랫폼 세부 정보 확인

에서 Snowball Edge 디바이스 AWS Marketplace 에 AMI를 추가하는 프로세스를 시작하기 전에 AMI의 제품 코드 및 플랫폼 세부 정보가에서 지원되는지 확인합니다 AWS 리전.

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 모음에서 인스턴스를 시작할 리전과 Snowball Edge 디바이스 주문 작업을 생성할 리전을 선택합니다. 현재 위치와 관계없이 사용자가 고를 수 있는 리전을 임의로 선택합니다.
3. 탐색 창에서 AMI를 선택합니다.
4. 필터 및 검색 옵션을 사용하여 기준과 일치하는 AMI만 볼 수 있도록 표시되는 AMI 목록의 범위를 지정합니다. 예를 들어에서 제공하는 AMIs는 퍼블릭 이미지를 AWS Marketplace선택합니다. 그런 다음 검색 옵션을 사용하여 표시되는 AMI 목록의 범위를 추가로 지정할 수 있습니다.
  - (새 콘솔) 검색 창을 선택하고 메뉴에서 소유자 별칭, = 연산자, 값 amazon을 차례로 선택합니다.
  - (기존 콘솔) 검색(Search) 창을 선택하고 메뉴에서 소유자(Owner), 값 Amazon 이미지(Amazon images)를 차례로 선택합니다.

**Note**

AMIs에는 소스 열에 aws-marketplace가 AWS Marketplace 포함됩니다.

5. AMI ID 열에서 AMI의 AMI ID를 선택합니다.
6. AMI의 이미지 요약에서 해당 리전에서 제품 코드를 지원하는지 확인합니다. 자세한 내용은 아래 표를 참조하세요.

지원되는 AWS Marketplace AMI 제품 코드

운영 체제	제품 코드
Ubuntu Server 14.04 LTS	b3dl4415quatdnd14qa6kcu45
CentOS 7(x86_64)	aw0evgkw8e5c1q413zgy5pjce
Ubuntu 16.04 LTS	csv6h7oyg29b7epjzg7qdr7no
Amazon Linux 2	avyfzznywekzgl5qv5f57ska
Ubuntu 20.04 LTS	a8jyyfnf4hjutohctm41o2z18m
Ubuntu 22.04 LTS	47xbqns9xujfkkjt189a13aqe

7. 그런 다음 플랫폼 세부 정보에 아래 목록의 항목 중 하나가 포함되어 있는지도 확인합니다.
  - Amazon Linux, Ubuntu 또는 Debian
  - Red Hat Linux bring-your-own-license
  - Amazon RDS for Oracle bring-your-own-license
  - Windows bring-your-own-license

## Snowball Edge용 Amazon Linux 2 AMI 버전 확인

다음 절차에 따라 Snowball Edge에서 Snowball Edge용 Amazon Linux 2 AMI의 버전을 확인합니다. 계속하기 AWS CLI 전에 최신 버전을 설치합니다. 자세한 내용은 AWS Command Line Interface 사용 설명서의 [의 최신 버전 설치 또는 업데이트를 AWS CLI](#) 참조하세요.

- `describe-images` AWS CLI 명령을 사용하여 AMI에 대한 설명을 확인합니다. 버전은 설명에 포함되어 있습니다. 이전 단계의 퍼블릭 키 인증서를 제공합니다. 자세한 내용은 AWS CLI 명령 참조의 [describe-images](#)를 참조하세요.

```
aws ec2 describe-images --endpoint http://snow-device-ip:8008 --region snow
```

### Example/ describe-images 명령의 출력

```
{
  "Images": [
    {
      "CreationDate": "2024-02-12T23:24:45.705Z",
      "ImageId": "s.ami-02ba84cb87224e16e",
      "Public": false,
      "ProductCodes": [
        {
          "ProductCodeId": "avyfzzywekctkg15qv5f57ska",
          "ProductCodeType": "marketplace"
        }
      ],
      "State": "AVAILABLE",
      "BlockDeviceMappings": [
        {
          "DeviceName": "/dev/xvda",
          "Ebs": {
            "DeleteOnTermination": true,
            "Iops": 0,
            "SnapshotId": "s.snap-0efb49f2f726fde63",
            "VolumeSize": 8,
            "VolumeType": "sbp1"
          }
        }
      ],
      "Description": "Snow Family Amazon Linux 2 AMI 2.0.20240131.0 x86_64
HVM gp2",
      "EnaSupport": false,
      "Name": "amzn2-ami-snow-family-hvm-2.0.20240131.0-x86_64-gp2-
b7e7f8d2-1b9e-4774-a374-120e0cd85d5a",
```

```

        "RootDeviceName": "/dev/xvda"
    }
]
}

```

이 예제에서 Snowball Edge용 Amazon Linux 2 AMI의 버전은 **2.0.20240131.0**. Description 이름 값에서 찾을 수 있습니다.

## Snowball Edge 디바이스의 AMI 구성

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 에서 지원되는 AMI의 새 인스턴스를 시작합니다 AWS Marketplace.

### Note

인스턴스 시작 시 인스턴스에 할당하는 스토리지 크기가 해당 사용 사례에 적합한지 확인합니다. Amazon EC2 콘솔에서는 스토리지 추가에서 이 작업을 수행합니다.

3. Snowball Edge에서 실행하고 싶은 애플리케이션을 설치 및 구성한 후 예상대로 작동하는지 테스트합니다.

### Important

- 단일 볼륨 AMI만 지원됩니다.
- AMI의 EBS 볼륨은 10TB 이하여야 합니다. AMI 데이터에 필요한 EBS 볼륨 크기를 프로비저닝하는 것이 좋습니다. 이렇게 하면 AMI를 내보내고 디바이스로 로드하는 데 걸리는 시간을 줄일 수 있습니다. 디바이스를 배포한 후 인스턴스 크기를 조정하거나 인스턴스에 볼륨을 추가할 수 있습니다.
- AMI의 EBS 스냅샷은 암호화되지 않아야 합니다.

4. 이 인스턴스를 생성하기 위해 SSH 키 페어에 사용한 PEM 또는 PPK 파일의 사본을 만듭니다. 이 파일을 Snowball Edge 디바이스와 통신하기 위해 사용하려는 서버에 저장합니다. SSH를 사용하여 디바이스의 EC2 호환 인스턴스에 연결할 때 필요하므로 이 파일의 경로를 기록해 둡니다.

**⚠ Important**

이 절차를 따르지 않으면 Snowball Edge 디바이스의 수령 시 인스턴스를 SSH와 연결할 수 없습니다.

5. 인스턴스를 AMI로 저장합니다. 자세한 정보는 Amazon EC2 사용 설명서의 [Linux 인스턴스용 Amazon EC2 사용 설명서](#)를 참조하세요.
6. SSH를 사용하여 연결하려는 각 인스턴스에 대해 1~4단계를 반복합니다. 각 SSH 키 페어의 사본을 만들고 연결된 AMI를 추적해야 합니다.
7. 이제 디바이스를 주문하면 이러한 AMI를 디바이스에 추가할 수 있습니다.

## 디바이스 수신 후 Snowball Edge에 AMI 추가

디바이스가 현장에 도착하면 새 AMI를 추가할 수 있습니다. 자세한 내용은 [Snowball Edge 디바이스로 가상 머신 이미지 가져오기](#) 섹션을 참조하세요. 모든 VM이 지원되기는 하지만, 지원되는 AMI만 전체 기능을 테스트했다는 점을 기억하세요.

**ℹ Note**

VM Import를 사용하여 AMI를 디바이스에 추가하거나 디바이스가 배포된 후 VM을 가져오는 경우 운영 체제를 사용하는 VM을 추가할 수 있습니다. 그러나 지원되는 운영 체제만 Snowball Edge에서 테스트 및 검증되었습니다. 디바이스로 가져오는 가상 이미지에 있는 운영 체제 또는 소프트웨어의 이용 약관을 준수할 책임은 사용자에게 있습니다.

**⚠ Important**

AWS 서비스가 Snowball Edge에서 제대로 작동하려면 서비스의 포트를 허용해야 합니다. 자세한 내용은 [Snowball Edge의 AWS 서비스에 대한 포트 요구 사항](#)을 참조하세요.

## Snowball Edge에 Microsoft Windows AMI 추가

지원되는 Windows 운영 체제를 사용하는 가상 머신(VMs)의 경우 VM Import/Export를 AWS 사용하여 Windows VM 이미지를 가져오거나 사이트에 배포된 직후 디바이스로 가져와 AMI를 추가할 수 있습니다.

## 기존 보유 라이선스 사용(BYOL)

Snowball Edge는 자체 라이선스가 있는 사용자의 디바이스로 Microsoft Windows AMI를 가져올 수 있도록 지원합니다. 기존 보유 라이선스 사용(BYOL)은 온프레미스 라이선스를 사용하여 소유한 AMI를 가져오는 프로세스입니다. AWS는 BYOL 옵션에 대한 공유 배포 옵션과 전용 배포 옵션을 모두 AWS 제공합니다.

VM Import/Export를 AWS 사용하여 로 가져오거나 사이트에 배포된 후 바로 디바이스로 가져와서 디바이스에 Windows VM 이미지를 추가할 수 있습니다. 에서 시작된 Windows AMIs 추가할 수 없습니다. 따라서 Snowball Edge 디바이스에서 AMI를 사용하려면 자체 Windows VM 이미지를 생성 및 가져오고 자체 라이선스를 가져와야 합니다. Windows 라이선스 및 BYOL에 대한 자세한 내용은 [Amazon Web Services](#) 및 [Microsoft: FAQ](#) 섹션을 참조하세요.

## Snowball Edge로 가져올 Windows VM 이미지 생성

Windows VM 이미지를 생성하려면 Windows 및 macOS 운영 체제에서 지원되는 VirtualBox와 같은 가상화 환경이 필요합니다. Snow 디바이스용 VM을 생성할 때는 RAM이 4GB 이상인 코어를 2개 이상 할당하는 것이 좋습니다. VM이 가동되어 실행되면 운영 체제(Windows Server 2012, 2016 또는 2019)를 설치해야 합니다. Snowball Edge 디바이스에 필요한 드라이버를 설치하려면 이 섹션의 지침을 따르세요.

Windows AMI를 Snow 디바이스에서 실행하려면 VirtIO, FLR, NetVCM, Vioinput, Viorng, Vioscsi, Vioserial 및 VioStor 드라이버를 추가해야 합니다. GitHub의 virtio-win-pkg-scripts 리포지토리에서 Windows 이미지에 이러한 드라이버를 설치하기 위한 [Microsoft 소프트웨어 설치 관리자\(virtio-win-guest-tools-installer\)](#)를 [다운로드](#)할 수 있습니다.

### Note

배포된 Snow 디바이스로 VM 이미지를 직접 가져오려는 경우 VM 이미지 파일은 RAW 형식이어야 합니다.

## Windows 이미지 생성

1. Microsoft Windows 컴퓨터에서 시작을 선택하고 **devmgmt.msc**를 선택하여 장치 관리자를 엽니다.
2. 기본 메뉴에서 작업을 선택한 다음 레거시 하드웨어 추가를 선택합니다.

**Note**

Windows 11의 경우 작업 메뉴에 레거시 하드웨어 추가 옵션이 나타나기 전에 먼저 아래 패널에서 디바이스를 선택합니다.

3. 마법사에서 다음을 선택합니다.
4. 목록에서 수동으로 선택한 하드웨어 설치(고급)를 선택하고 다음을 선택합니다.
5. 모든 장치 보기를 선택하고 다음을 선택합니다.
6. 디스크 있음을 선택하고 목록에서 제조업체 파일 복사를 연 다음 ISO 파일을 탐색합니다.
7. ISO 파일에서 Driver\W2K8R2\amd64 디렉터리를 탐색한 다음 해당 .INF 파일을 찾습니다.
8. .INF 파일을 선택하고 열기를 차례로 선택한 다음 확인을 선택합니다.
9. 드라이버 이름이 표시되면 다음을 선택하고 다음을 두 번 더 선택합니다. 그런 다음 Finish(완료)를 선택합니다.

그러면 새 드라이버를 사용하여 디바이스가 설치됩니다. 실제 하드웨어는 존재하지 않으므로 디바이스에 문제가 있음을 나타내는 노란색 느낌표가 표시됩니다. 이 문제를 해결해야 합니다.

**하드웨어 문제 해결**

1. 느낌표가 있는 디바이스의 컨텍스트(마우스 오른쪽 버튼 클릭) 메뉴를 엽니다.
2. 제거를 선택하고 이 장치의 드라이버 소프트웨어 삭제를 선택 취소한 다음 확인을 선택합니다.

드라이버가 설치되고 디바이스에서 AMI를 시작할 준비가 되었습니다.

**Snowball Edge로 VM 이미지 가져오기**

VM 이미지를 준비한 후 옵션 중 하나를 사용하여 이미지를 디바이스로 가져올 수 있습니다.

- VM Import/Export를 사용하는 클라우드에서 로 VM 이미지를 가져 AWS 와 AMI로 등록할 때에서 주문할 때 디바이스에 추가할 수 있습니다 AWS Snow 패밀리 관리 콘솔. 자세한 내용은 VM Import/Export 사용 설명서에서 [VM Import/Export를 사용하여 VM을 이미지로 가져오기](#)를 참조하세요.
- 사이트에 배포된 디바이스에서 로컬로 - AWS OpsHub 또는 AWS Command Line Interface ()를 사용하여 VM 이미지를 디바이스로 직접 가져올 수 있습니다 AWS CLI.

사용에 대한 자세한 내용은 로컬에서 Amazon EC2 호환 컴퓨팅 인스턴스 사용을 AWS OpsHub참조하세요. [EC2-compatible](#)

사용에 대한 자세한 내용은 단원을 AWS CLI참조하십시오 [Snowball Edge 디바이스로 가상 머신 이미지 가져오기](#).

## Snowball Edge용 최신 Amazon Linux 2 AMI 내보내기

Amazon Linux 2 AMIs 최신 버전으로 업데이트하려면 먼저 최신 Amazon Linux 2 VM 이미지를에서 내보낸 AWS Marketplace다음 해당 VM 이미지를 Snow 디바이스로 가져옵니다.

1. `ssm get-parameters` AWS CLI 명령을 사용하여에서 Amazon Linux 2 AMI의 최신 이미지 ID를 찾습니다 AWS Marketplace.

```
aws ssm get-parameters --names /aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2 --query 'Parameters[0].[Value]' --region your-region
```

이 명령은 AMI의 최신 이미지 ID를 반환합니다. 예를 들어 `ami-0ccb473bada910e74`입니다.

2. 최신 Amazon Linux 2 이미지를 내보냅니다. Amazon EC2 사용 설명서의 [Amazon Machine Image\(AMI\)에서 직접 VM 내보내기](#)를 참조하세요. `ec2 export-image` 명령의 `image-id` 파라미터 값으로 Amazon Linux 2 AMI의 최신 이미지 ID를 사용합니다.
3. AWS CLI 또는를 사용하여 VM 이미지를 Snow 디바이스로 가져옵니다 AWS OpsHub.
  - 사용에 대한 자세한 내용은 단원을 AWS CLI참조하십시오 [Snowball Edge 디바이스로 가상 머신 이미지 가져오기](#).
  - 사용에 대한 자세한 내용은 단원을 AWS OpsHub참조하십시오 [를 사용하여 Amazon EC2-compatible AMI로 이미지 가져오기 AWS OpsHub](#).

## Snowball Edge 디바이스로 가상 머신 이미지 가져오기

AWS CLI 및 VM Import/Export 서비스를 사용하여 가상 머신(VM) 이미지를 Snowball Edge 디바이스로 Amazon Machine Image(AMI)로 가져올 수 있습니다. VM 이미지를 가져온 후 AMI로 등록하고 Amazon EC2-compatible 인스턴스로 시작합니다.



Snowball Edge 디바이스 주문 작업을 생성할 때 Amazon EC2의 AMIs를 디바이스에 추가할 수 있습니다. Snowball Edge 디바이스를 받은 후이 절차를 사용합니다. 자세한 내용은 [컴퓨팅 및 스토리지 옵션 선택](#) 단원을 참조하십시오.

를 사용하여 VM 이미지 파일을 업로드 AWS OpsHub 할 수도 있습니다. 자세한 내용은 [이미지를 Amazon EC2-compatible AMI로 디바이스로 가져오기](#)를 참조하세요.

## 주제

- [1단계: VM 이미지를 준비하여 Snowball Edge 디바이스에 업로드](#)
- [2단계: Snowball Edge에 필요한 권한 설정](#)
- [3단계: Snowball Edge에서 VM 이미지를 스냅샷으로 가져오기](#)
- [4단계: Snowball Edge에서 스냅샷을 AMI로 등록](#)
- [5단계: Snowball Edge의 AMI에서 인스턴스 시작](#)
- [Snowball Edge에 대한 추가 AMI 작업](#)

## 1단계: VM 이미지를 준비하여 Snowball Edge 디바이스에 업로드

VM Import/Export를 AWS 클라우드 사용하여의 Amazon EC2 AMI 또는 인스턴스에서 VM 이미지를 내보내거나 선택한 가상화 플랫폼을 사용하여 로컬에서 VM 이미지를 생성하여 VM 이미지를 준비합니다.

VM Import/Export를 사용하여 Amazon EC2 인스턴스를 VM 이미지로 내보내려면 VM Import/Export 사용 설명서의 [VM Import/Export를 사용하여 인스턴스를 VM으로 내보내기](#)를 참조하세요. VM Import/Export를 사용하여 Amazon EC2 AMI를 VM 이미지로 내보내려면 VM Import/Export 사용 설명서의 [Amazon Machine Image\(AMI\)에서 직접 VM 내보내기](#)를 참조하세요.

로컬 환경에서 VM 이미지를 생성하는 경우 이미지가 Snowball Edge 디바이스에서 AMI로 사용하도록 구성되어 있는지 확인합니다. 환경에 따라 다음 항목을 구성해야 할 수 있습니다.

- 운영 체제 구성 및 업데이트
- 호스트 이름을 설정합니다.
- 네트워크 시간 프로토콜(NTP)이 구성되어 있는지 확인합니다.
- 필요한 경우 SSH 퍼블릭 키를 포함합니다. 키 페어의 로컬 복사본을 만듭니다. 자세한 내용은 [SSH를 사용하여 Snowball Edge의 컴퓨팅 인스턴스에 연결](#)을 참조하세요.
- Snowball Edge 디바이스에서 사용할 소프트웨어를 설치하고 구성합니다.

**Note**

Snowball Edge 디바이스의 디스크 스냅샷을 준비할 때 다음 제한 사항에 유의하세요.

- Snowball Edge는 현재 RAW 이미지 형식의 스냅샷 가져오기만 지원합니다.
- Snowball Edge는 현재 크기가 1GB~1TB인 스냅샷 가져오기만 지원합니다.

## Snowball Edge 디바이스의 Amazon S3 버킷에 VM 이미지 업로드

VM 이미지를 준비한 후 Snowball Edge 디바이스 또는 클러스터의 S3 버킷에 업로드합니다. Snowball Edge에서 S3 어댑터 또는 Amazon S3 호환 스토리지를 사용하여 스냅샷을 업로드할 수 있습니다.

S3 어댑터를 사용하여 가상 머신 이미지를 업로드하려면

- `cp` 명령을 사용하여 VM 이미지 파일을 디바이스의 버킷에 복사합니다.

```
aws s3 cp image-path s3://S3-bucket-name --endpoint http://S3-object-API-endpoint:443 --profile profile-name
```

자세한 내용은 이 설명서의 [지원되는 AWS CLI 명령을 참조하세요](#).

Snowball Edge에서 Amazon S3 호환 스토리지를 사용하여 VM 이미지를 업로드하려면

- `put-object` 명령을 사용하여 스냅샷 파일을 디바이스의 버킷에 복사합니다.

```
aws s3api put-object --bucket bucket-name --key path-to-snapshot-file --body snapshot-file --endpoint-url s3api-endpoint-ip --profile your-profile
```

자세한 내용은 [Snowball Edge 디바이스에서 S3 객체 작업을 참조하세요](#).

## 2단계: Snowball Edge에 필요한 권한 설정

가져오기에 성공하려면 Snowball Edge 디바이스, Amazon EC2 및 사용자에서 VM Import/Export에 대한 권한을 설정해야 합니다.

**Note**

이러한 권한을 제공하는 서비스 역할 및 정책은 Snowball Edge 디바이스에 있습니다.

## Snowball Edge에서 VM Import/Export에 필요한 권한

가져오기 프로세스를 시작하려면 먼저 Snowball Edge 디바이스에서 VM Import/Export가 역할을 수입하도록 허용하는 신뢰 정책을 사용하여 IAM 역할을 생성해야 합니다. 디바이스의 VM Import/Export가 디바이스의 S3 버킷에 저장된 이미지에 액세스할 수 있도록 허용하는 추가 권한이 역할에 부여됩니다.

### 신뢰 정책 json 파일 생성

다음은 S3 버킷에서 가져와야 하는 스냅샷에 VM Import가 액세스할 수 있도록 역할에 연결하는 데 필요한 신뢰 정책의 예시입니다.

### JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "vmie.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

### 신뢰 정책 json 파일로 역할 생성

역할 이름은 vmimport일 수 있습니다. 명령에서 --role-name 옵션을 사용하여 변경할 수 있습니다.

```
aws iam create-role --role-name role-name --assume-role-policy-document file:///trust-policy-json-path --endpoint http://snowball-ip:6078 --region snow --profile profile-name
```

다음은 create-role 명령의 출력 예시입니다.

```
{
  "Role":{
    "AssumeRolePolicyDocument":{
      "Version":"2012-10-17",
      "Statement":[
        {
          "Action":"sts:AssumeRole",
          "Effect":"Allow",
          "Principal":{
            "Service":"vmie.amazonaws.com"
          }
        }
      ]
    },
    "MaxSessionDuration":3600,
    "RoleId":"AROACEMGEZDGNBVG3TQ0JQGEZAAAABQBB6NSGNAAAABPSVLTREPY3FPAFOLKJ3",
    "CreateDate":"2022-04-19T22:17:19.823Z",
    "RoleName":"vmimport",
    "Path":"/",
    "Arn":"arn:aws:iam::123456789012:role/vmimport"
  }
}
```

## 역할에 대한 정책 생성

다음 예시 정책에는 Amazon S3에 액세스할 수 있는 최소한의 필수 권한이 있습니다. Amazon S3 버킷 이름을 이미지가 있는 이름으로 변경합니다. 독립형 Snowball Edge 디바이스의 경우 *snow-id*를 작업 ID로 변경합니다. 디바이스 클러스터의 경우 *snow-id*를 클러스터 ID로 변경합니다. 또한 접두사를 사용하여 VM Import가 스냅샷을 가져올 수 있는 위치를 더 좁힐 수 있습니다. 다음과 같은 정책 json 파일을 생성합니다.

## JSON

```
{
  "Version":"2012-10-17",
  "Statement":[
    {
      "Effect":"Allow",
      "Action":[
        "s3:GetBucketLocation",
```

```

        "s3:GetObject",
        "s3:ListBucket",
        "s3:GetMetadata"
    ],
    "Resource": [
        "arn:aws:s3:snow:account-id:snow/snow-id/bucket/import-snapshot-
bucket-name",
        "arn:aws:s3:snow:account-id:snow/snow-id/bucket/import-snapshot-
bucket-name/*"
    ]
  }
]
}

```

정책 파일을 사용하여 정책을 생성합니다.

```
aws iam create-policy --policy-name policy-name --policy-document file:///policy-json-
file-path --endpoint http://snowball-ip:6078 --region snow --profile profile-name
```

다음은 create-policy 명령의 출력 예시입니다.

```

{
  "Policy":{
    "PolicyName":"vmimport-resource-policy",
    "PolicyId":"ANPACEMGEZDGNBVG3TQ0JQGEZAAAAB00EE3IIHAAAABWZJPI2VW4UUTFEDBC2R",
    "Arn":"arn:aws:iam:123456789012:policy/vmimport-resource-policy",
    "Path":"/",
    "DefaultVersionId":"v1",
    "AttachmentCount":0,
    "IsAttachable":true,
    "CreateDate":"2020-07-25T23:27:35.690000+00:00",
    "UpdateDate":"2020-07-25T23:27:35.690000+00:00"
  }
}

```

## 정책에 역할 연결

이전 역할에 정책을 연결하고 필요한 리소스에 액세스할 수 있는 권한을 부여합니다. 이렇게 하면 로컬 VM Import 서비스가 Amazon S3의 스냅샷을 디바이스에 다운로드할 수 있습니다.

```
aws iam attach-role-policy --role-name role-name --policy-arn
arn:aws:iam::123456789012:policy/policy-name --endpoint http://snowball-ip:6078 --
region snow --profile profile-name
```

## Snowball Edge에서 호출자에게 필요한 권한

Snowball Edge VM Import가 담당할 역할 외에도 사용자가 VMIE에 역할을 넘길 수 있는 권한을 가지고 있는지 확인해야 합니다. 기본 루트 사용자를 사용하여 가져오기를 수행하는 경우 루트 사용자는 이미 필요한 모든 권한을 가지고 있으므로 이 단계를 건너뛰고 3단계로 이동해도 됩니다.

가져오기를 수행하는 사용자에게 다음 두 IAM 권한을 연결합니다.

- pass-role
- get-role

### 역할에 대한 정책 생성

다음은 사용자가 IAM 역할에 대해 get-role 및 pass-role 작업을 수행할 수 있도록 허용하는 정책 예시입니다.

### JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "iam:GetRole",
      "Resource": "*"
    },
    {
      "Sid": "iamPassRole",
      "Effect": "Allow",
      "Action": "iam:PassRole",
      "Resource": "arn:aws:iam::*:role/snowball*",
      "Condition": {
        "StringEquals": {
          "iam:PassedToService": "importexport.amazonaws.com"
        }
      }
    }
  ]
}
```

```

    }
  ]
}

```

정책 파일을 사용하여 정책을 생성합니다.

```
aws iam create-policy --policy-name policy-name --policy-document file:///policy-json-  
file-path --endpoint http://snowball-ip:6078 --region snow --profile profile-name
```

다음은 create-policy 명령의 출력 예시입니다.

```
{
  "Policy":{
    "PolicyName":"caller-policy",
    "PolicyId":"ANPACEMGEZDGNBVG3TQ0JQGEZAAAAB000TUOE3AAAAAPPBEUM7Q7ARPUE53C6R",
    "Arn":"arn:aws:iam::123456789012:policy/caller-policy",
    "Path":"/",
    "DefaultVersionId":"v1",
    "AttachmentCount":0,
    "IsAttachable":true,
    "CreateDate":"2020-07-30T00:58:25.309000+00:00",
    "UpdateDate":"2020-07-30T00:58:25.309000+00:00"
  }
}
```

정책이 생성되면 Amazon EC2 API 또는 CLI 작업을 직접 호출하여 스냅샷을 가져올 IAM 사용자에게 정책을 연결합니다.

```
aws iam attach-user-policy --user-name your-user-name --policy-arn  
arn:aws:iam::123456789012:policy/policy-name --endpoint http://snowball-ip:6078 --  
region snow --profile profile-name
```

## Snowball Edge에서 Amazon EC2 APIs 호출하는 데 필요한 권한

스냅샷을 가져오려면 IAM 사용자에게 `ec2:ImportSnapshot` 권한이 있어야 합니다. 사용자 액세스를 제한할 필요가 없는 경우 `ec2:*` 권한을 사용하여 전체 Amazon EC2 액세스 권한을 부여할 수 있습니다. 다음은 디바이스의 Amazon EC2에 대해 부여하거나 제한할 수 있는 권한입니다. 표시된 내용에 따라 정책 파일을 생성합니다.

## JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:ImportSnapshot",
        "ec2:DescribeImportSnapshotTasks",
        "ec2:CancelImportTask",
        "ec2:DescribeSnapshots",
        "ec2>DeleteSnapshot",
        "ec2:RegisterImage",
        "ec2:DescribeImages",
        "ec2:DeregisterImage"
      ],
      "Resource": "*"
    }
  ]
}
```

정책 파일을 사용하여 정책을 생성합니다.

```
aws iam create-policy --policy-name policy-name --policy-document file:///policy-json-  
file-path --endpoint http://snowball-ip:6078 --region snow --profile profile-name
```

다음은 create-policy 명령의 출력 예시입니다.

```
{
  "Policy": {
    "PolicyName": "ec2-import.json",
    "PolicyId": "ANPACEMGEZDGNBVG3TQ0JQGEZAAAAABQBGPDQC5AAAAATYN62UNBFYTF5WVCSCZS",
    "Arn": "arn:aws:iam::123456789012:policy/ec2-import.json",
    "Path": "/",
    "DefaultVersionId": "v1",
    "AttachmentCount": 0,
    "IsAttachable": true,
    "CreateDate": "2022-04-21T16:25:53.504000+00:00",
  }
}
```



```

      "UpdateDate": "2022-04-21T16:25:53.504000+00:00"
    }
  }
}

```

정책이 생성되면 Amazon EC2 API 또는 CLI 작업을 직접 호출하여 스냅샷을 가져올 IAM 사용자에게 정책을 연결합니다.

```

aws iam attach-user-policy --user-name your-user-name --policy-arn
arn:aws:iam::123456789012:policy/policy-name --endpoint http://snowball-ip:6078 --
region snow --profile profile-name

```

### 3단계: Snowball Edge에서 VM 이미지를 스냅샷으로 가져오기

다음 단계는 VM 이미지를 디바이스의 스냅샷으로 가져오는 것입니다. S3Bucket 파라미터의 값은 VM 이미지가 포함된 버킷의 이름입니다. S3Key 파라미터의 값은 이 버킷의 VM 이미지 파일에 대한 경로입니다.

```

aws ec2 import-snapshot --disk-container "Format=RAW,UserBucket={S3Bucket=bucket-
name,S3Key=image-file}" --endpoint http://snowball-ip:8008 --region snow --
profile profile-name

```

자세한 내용은 AWS CLI 명령 참조의 [import-snapshot](#)을 참조하세요.

이 명령은 다음과 같은 스위치를 지원하지 않습니다.

- [--client-data value]
- [--client-token value]
- [--dry-run]
- [--no-dry-run]
- [--encrypted]
- [--no-encrypted]
- [--kms-key-id value]
- [--tag-specifications value]

#### Example **import-snapshot** 명령의 출력

```
{
```

```

"ImportTaskId": "s.import-snap-1234567890abc",
"SnapshotTaskDetail": {
  "DiskImageSize": 2.0,
  "Encrypted": false,
  "Format": "RAW",
  "Progress": "3",
  "Status": "active",
  "StatusMessage": "pending",
  "UserBucket": {
    "S3Bucket": "bucket",
    "S3Key": "vmimport/image01"
  }
}
}
}

```

### Note

Snowball Edge는 현재 디바이스당 한 번에 하나의 활성화 가져오기 작업만 실행하도록 허용합니다. 새 가져오기 작업을 시작하려면 현재 작업이 완료될 때까지 기다리거나 클러스터에서 사용 가능한 다른 노드를 선택합니다. 원하는 경우 현재 가져오기를 취소할 수도 있습니다. 지연을 방지하려면 가져오기가 진행 중인 동안에는 Snowball Edge 디바이스를 재부팅하지 않습니다. 디바이스를 재부팅하면 가져오기가 실패하고 디바이스에 액세스할 수 있게 될 때 진행 상황이 삭제됩니다. 스냅샷 가져오기 태스크 상태를 확인하려면 다음 명령을 사용합니다.

```

aws ec2 describe-import-snapshot-tasks --import-task-ids id --endpoint
http://snowball-ip:8008 --region snow --profile profile-name

```

## 4단계: Snowball Edge에서 스냅샷을 AMI로 등록

디바이스로 스냅샷을 성공적으로 가져오면 `register-image` 명령을 사용하여 스냅샷을 등록할 수 있습니다.

### Note

모든 스냅샷을 사용할 수 있는 경우에만 AMI를 등록할 수 있습니다.

자세한 내용은 AWS CLI 명령 참조의 [register-image](#)를 참조하세요.

## Example/ register-image 명령

```
aws ec2 register-image \
--name ami-01 \
--description my-ami-01 \
--block-device-mappings "[{\"DeviceName\": \"/dev/sda1\", \"Ebs\": {\"Encrypted\": false, \"DeleteOnTermination\": true, \"SnapshotId\": \"snapshot-id\", \"VolumeSize\": 30}}]" \
--root-device-name /dev/sda1 \
--endpoint http://snowball-ip:8008 \
--region snow \
--profile profile-name
```

다음은 블록 디바이스 매핑 JSON의 예시입니다. 자세한 내용은 AWS CLI 명령 참조의 [register-image](#)의 [block-device-mapping](#) 파라미터를 참조하세요.

```
[
  {
    "DeviceName": "/dev/sda",
    "Ebs": {
      "Encrypted": false,
      "DeleteOnTermination": true,
      "SnapshotId": "snapshot-id",
      "VolumeSize": 30
    }
  }
]
```

## Example/ register-image 명령

```
{
  "ImageId": "s.ami-8de47d2e397937318"
}
```

## 5단계: Snowball Edge의 AMI에서 인스턴스 시작

인스턴스를 시작하려면 AWS CLI 명령 참조의 [run-instances](#)를 참조하세요.

image-id 파라미터의 값은 register-image 명령의 출력으로서 ImageId 이름의 값입니다.

```
aws ec2 run-instances --image-id image-id --instance-type instance-type --endpoint
http://snowball-ip:8008 --region snow --profile profile-name
```

```
{
  "Instances":[
    {
      "SourceDestCheck":false,
      "CpuOptions":{
        "CoreCount":1,
        "ThreadsPerCore":2
      },
      "InstanceId":"s.i-12345a73123456d1",
      "EnaSupport":false,
      "ImageId":"s.ami-1234567890abcdefg",
      "State":{
        "Code":0,
        "Name":"pending"
      },
      "EbsOptimized":false,
      "SecurityGroups":[
        {
          "GroupName":"default",
          "GroupId":"s.sg-1234567890abc"
        }
      ],
      "RootDeviceName":"/dev/sda1",
      "AmiLaunchIndex":0,
      "InstanceType":"sbe-c.large"
    }
  ],
  "ReservationId":"s.r-1234567890abc"
}
```

#### Note

AWS OpsHub 를 사용하여 인스턴스를 시작할 수도 있습니다. 자세한 내용은 이 설명서의 [Amazon EC2-compatible 인스턴스 시작하기](#)를 참조하세요.

## Snowball Edge에 대한 추가 AMI 작업

추가 AWS CLI 명령을 사용하여 스냅샷 가져오기 상태를 모니터링하고, 가져온 스냅샷에 대한 세부 정보를 가져오고, 스냅샷 가져오기를 취소하고, 스냅샷을 가져온 후 스냅샷을 삭제하거나 등록 취소할 수 있습니다.

### Snowball Edge에서 스냅샷 가져오기 상태 모니터링

Amazon EC2 `describe-import-snapshot-tasks` 명령을 실행하여 현재 가져오기 진행 상태를 확인할 수 있습니다. 이 명령은 `task-state`의 페이지 매김 및 필터링을 지원합니다.

#### Example/ describe-import-snapshot-tasks 명령

```
aws ec2 describe-import-snapshot-tasks --import-task-ids id --endpoint http://snowball-ip:8008 --region snow --profile profile-name
```

#### Example/ describe-import-snapshot-tasks 명령 출력

```
{
  "ImportSnapshotTasks": [
    {
      "ImportTaskId": "s.import-snap-8f6bfd7fc9ead9aca",
      "SnapshotTaskDetail": {
        "Description": "Created by AWS-Snowball-VMImport service for s.import-snap-8f6bfd7fc9ead9aca",
        "DiskImageSize": 8.0,
        "Encrypted": false,
        "Format": "RAW",
        "Progress": "3",
        "SnapshotId": "s.snap-848a22d7518ad442b",
        "Status": "active",
        "StatusMessage": "pending",
        "UserBucket": {
          "S3Bucket": "bucket1",
          "S3Key": "image1"
        }
      }
    }
  ]
}
```

**Note**

이 명령은 지난 7일 이내에 성공적으로 완료되었거나 삭제된 것으로 표시된 작업에 대한 출력만 표시합니다. 필터링은 Name=task-state, Values=active | deleting | deleted | completed만 지원합니다.

이 명령은 다음과 같은 파라미터를 지원하지 않습니다.

- [--dry-run]
- [--no-dry-run]

## Snowball Edge에서 가져오기 작업 취소

가져오기 태스크를 취소하려면 `cancel-import-task` 명령을 실행합니다.

Example/ `cancel-import-task` 명령

```
aws ec2 cancel-import-task --import-task-id import-task-id --endpoint http://snowball-ip:8008 --region snow --profile profile-name
```

Example/ `cancel-import-task` 명령 출력

```
{
  "ImportTaskId": "s.import-snap-8234ef2a01cc3b0c6",
  "PreviousState": "active",
  "State": "deleting"
}
```

**Note**

완료 상태가 아닌 태스크만 취소할 수 있습니다.

이 명령은 다음과 같은 파라미터를 지원하지 않습니다.

- [--dry-run]
- [--no-dry-run]

## Snowball Edge에서 스냅샷 설명

스냅샷을 가져온 후 이 명령을 사용하여 스냅샷을 설명할 수 있습니다. 스냅샷을 필터링하려면 이전 가져오기 태스크 응답의 스냅샷 ID를 사용하여 `snapshot-ids`에서 전달할 수 있습니다. 이 명령은 `volume-id`, `status` 및 `start-time`의 페이지 매김 및 필터링을 지원합니다.

### Example/ describe-snapshots 명령

```
aws ec2 describe-snapshots --snapshot-ids snapshot-id --endpoint http://snowball-  
ip:8008 --region snow --profile profile-name
```

### Example/ describe-snapshots 명령 출력

```
{  
  "Snapshots": [  
    {  
      "Description": "Created by AWS-Snowball-VMImport service for s.import-  
snap-8f6bfd7fc9ead9aca",  
      "Encrypted": false,  
      "OwnerId": "123456789012",  
      "SnapshotId": "s.snap-848a22d7518ad442b",  
      "StartTime": "2020-07-30T04:31:05.032000+00:00",  
      "State": "completed",  
      "VolumeSize": 8  
    }  
  ]  
}
```

이 명령은 다음과 같은 파라미터를 지원하지 않습니다.

- `[--restorable-by-user-ids value]`
- `[--dry-run]`
- `[--no-dry-run]`

## Snowball Edge 디바이스에서 스냅샷 삭제

소유하고 있어 더 이상 필요하지 않은 스냅샷을 제거하려면 `delete-snapshot` 명령을 사용할 수 있습니다.

## Example/ delete-snapshot 명령

```
aws ec2 delete-snapshot --snapshot-id snapshot-id --endpoint http://snowball-ip:8008 --region snow --profile profile-name
```

### Note

Snowball Edge는 보류 중 상태이거나 AMI의 루트 디바이스로 지정된 스냅샷의 삭제를 지원하지 않습니다.

이 명령은 다음과 같은 파라미터를 지원하지 않습니다.

- [--dry-run]
- [--no-dry-run]

## Snowball Edge에서 AMI 등록 취소

더 이상 필요 없는 AMI의 등록을 취소하려면 deregister-image 명령을 사용할 수 있습니다. 보류 중 상태인 AMI의 등록 취소는 현재 지원되지 않습니다.

## Example/ deregister-image 명령

```
aws ec2 deregister-image --image-id image-id --endpoint http://snowball-ip:8008 --region snow --profile profile-name
```

이 명령은 다음과 같은 파라미터를 지원하지 않습니다.

- [--dry-run]
- [--no-dry-run]

## Snowball Edge 디바이스에서 AWS CLI 및 API 작업 사용

AWS Command Line Interface (AWS CLI) 또는 API 작업을 사용하여 Snowball Edge에서 IAM, Amazon S3 및 Amazon EC2 명령을 실행할 때는 ""region로 지정해야 합니다snow. 다음 예제와 같이 명령 자체 내에서 AWS configure 또는를 사용하여이 작업을 수행할 수 있습니다.

```
aws configure --profile ProfileName
```



```
AWS Access Key ID [None]: defgh
AWS Secret Access Key [None]: 1234567
Default region name [None]: snow
Default output format [None]: json
```

Or

```
aws s3 ls --endpoint http://192.0.2.0:8080 --region snow --profile ProfileName
```

## Snowball Edge의 컴퓨팅 인스턴스에 대한 네트워크 구성

Snowball Edge에서 컴퓨팅 인스턴스를 시작한 후에는 네트워크 인터페이스를 생성하여 IP 주소를 제공해야 합니다. Snowball Edges는 가상 네트워크 인터페이스와 직접 네트워크 인터페이스라는 두 가지 종류의 네트워크 인터페이스를 지원합니다.

가상 네트워크 인터페이스(VNI) - 가상 네트워크 인터페이스는 Snowball Edge의 EC2-compatible 인스턴스에 연결하기 위한 표준 네트워크 인터페이스입니다. 다이렉트 네트워크 인터페이스 사용 여부에 관계없이 EC2 호환 인스턴스 각각에 대해 VNI를 생성해야 합니다. VNI를 통과하는 트래픽은 설정한 보안 그룹에 의해 보호됩니다. Snowball Edge를 제어하는 데 사용하는 물리적 네트워크 포트에만 VNIs를 연결할 수 있습니다.

### Note

VNI는 Snowball Edge를 관리하는 데 사용되는 것과 동일한 물리적 인터페이스(RJ45, SFP+ 또는 QSFP)를 사용합니다. 디바이스 관리에 사용되는 인터페이스와 다른 물리적 인터페이스에 VNI를 만들면 예상치 못한 결과가 발생할 수 있습니다.

다이렉트 네트워크 인터페이스(DNI) - 멀티캐스트 스트림, 전환 라우팅, 로드 밸런싱과 같은 사용 사례를 지원하는 고급 네트워크 기능입니다. 중간 번역 또는 필터링 없이 인스턴스에 계층 2 네트워크 액세스를 제공하여 Snowball Edge의 네트워크 구성에 대한 유연성을 높이고 네트워크 성능을 개선할 수 있습니다. DNI는 VLAN 태그를 지원하고 MAC 주소를 사용자 지정합니다. DNI의 트래픽은 보안 그룹에 의해 보호되지 않습니다.

Snowball Edge 디바이스에서는 DNI를 RJ45, SFP 또는 QSFP 포트와 연결할 수 있습니다. 각 물리적 포트는 최대 63개의 DNI를 지원합니다. DNIs Snowball Edge를 관리하는 데 사용하는 것과 동일한 물리적 네트워크 포트에 연결할 필요가 없습니다.

**Note**

Snowball Edge 스토리지 최적화(EC2 컴퓨팅 기능 포함) 디바이스는 DNI를 지원하지 않습니다.

## 주제

- [Snowball Edge의 DNIs 또는 VNIs에 대한 사전 조건](#)
- [Snowball Edge에서 가상 네트워크 인터페이스\(VNI\) 설정](#)
- [Snowball Edge에서 직접 네트워크 인터페이스\(DNI\) 설정](#)

## Snowball Edge의 DNIs 또는 VNIs에 대한 사전 조건

VNI 또는 DNI를 구성하려면 먼저 다음 사전 조건을 충족해야 합니다.

1. 디바이스에 전원이 연결되어 있고 RJ45 포트와 같은 물리적 네트워크 인터페이스 중 하나가 IP 주소와 연결되어 있는지 확인합니다.
2. Snowball Edge에서 사용 중인 물리적 네트워크 인터페이스와 연결된 IP 주소를 가져옵니다.
3. Snowball Edge 클라이언트를 구성합니다. 자세한 내용은 [Snowball Edge 클라이언트의 프로필 구성](#) 단원을 참조하십시오.
4. AWS CLI 구성. 자세한 내용은 AWS Command Line Interface 사용 설명서 [시작하기 AWS CLI](#)를 참조하세요.
5. 디바이스의 잠금을 해제합니다.
  - AWS OpsHub 를 사용하여 디바이스를 잠금 해제합니다. 자세한 내용은 Snowball Edge 참조하세요. [AWS OpsHub](#)
  - Snowball Edge 클라이언트를 사용하여 디바이스를 잠금 해제합니다. 자세한 내용은 [Snowball Edge 잠금 해제](#) 단원을 참조하십시오.
6. 디바이스에서 EC2 호환 인스턴스를 시작합니다. VNI를 이 인스턴스와 연결합니다.
7. Snowball Edge 클라이언트를 사용하여 describe-device 명령을 실행합니다. 명령의 출력에서 물리적 네트워크 인터페이스 ID 목록을 제공합니다. 자세한 내용은 [Snowball Edge 상태 보기](#) 단원을 참조하십시오.
8. 사용하려는 물리적 네트워크 인터페이스의 ID를 알아내 이를 기록해 둡니다.

## Snowball Edge에서 가상 네트워크 인터페이스(VNI) 설정

물리적 네트워크 인터페이스의 ID를 식별한 후 해당하는 물리적 인터페이스를 사용하여 가상 네트워크 인터페이스(VNI)를 설정할 수 있습니다. 다음 절차에 따라 VNI를 설정합니다. VNI를 생성하기 전에 필수 작업을 수행해야 합니다.

### VNI 생성 및 IP 주소 연결

1. Snowball Edge 클라이언트를 사용하여 `create-virtual-network-interface` 명령을 실행합니다. 다음 예시에서는 두 가지 서로 다른 IP 주소 배정 방법, 즉 DHCP 또는 STATIC을 사용하여 이 명령을 실행하는 방법을 보여줍니다. DHCP 메서드는 Dynamic Host Configuration Protocol(DHCP)를 사용합니다.

```
snowballEdge create-virtual-network-interface \
--physical-network-interface-id s.ni-abcd1234 \
--ip-address-assignment DHCP \
--profile profile-name
//OR//

snowballEdge create-virtual-network-interface \
--physical-network-interface-id s.ni-abcd1234 \
--ip-address-assignment STATIC \
--static-ip-address-configuration IpAddress=192.0.2.0,Netmask=255.255.255.0 \
--profile profile-name
```

이 명령은 IP 주소가 포함된 JSON 구조를 반환합니다. 프로세스 후반부에서 `ec2 associate-address` AWS CLI 명령과 함께 사용할 IP 주소를 기록해 둡니다.

이 IP 주소가 필요할 때마다 Snowball Edge 클라이언트 명령 `describe-virtual-network-interfaces` Snowball Edge 클라이언트 명령 또는 AWS CLI 명령을 사용하여 `aws ec2 describe-addresses` 가져올 수 있습니다.

2. AWS CLI 를 사용하여 IP 주소를 EC2-compatible 인스턴스와 연결하고 빨간색 텍스트를 값으로 바꿉니다.

```
aws ec2 associate-address --public-ip 192.0.2.0 --instance-id s.i-01234567890123456
--endpoint http://Snowball Edge physical IP address:8008
```

## Snowball Edge에서 직접 네트워크 인터페이스(DNI) 설정

### Note

직접 네트워크 인터페이스 기능은 2021년 1월 12일 이후에 사용할 수 있으며 Snowball Edges를 사용할 수 있는 모든 AWS 리전에서 사용할 수 있습니다.

### Snowball Edge의 DNI에 대한 사전 조건

직접 네트워크 인터페이스(DNI)를 설정하기 전에 사전 조건 섹션의 작업을 수행해야 합니다.

1. DNI를 설정하기 전에 필수 작업을 수행합니다. 지침은 [Snowball Edge의 DNIs 또는 VNIs에 대한 사전 조건](#) 섹션을 참조하세요.
2. 또한 디바이스에서 인스턴스를 시작하고, VNI를 생성한 다음, 인스턴스와 연결해야 합니다. 지침은 [Snowball Edge에서 가상 네트워크 인터페이스\(VNI\) 설정](#) 섹션을 참조하세요.

### Note

현장 소프트웨어 업데이트를 수행하여 기존 디바이스에 직접 네트워킹을 추가한 경우, 기능을 완전히 활성화하려면 디바이스를 두 번 재시작해야 합니다.

DNI를 생성하고 IP 주소를 연결합니다.

1. 다음 명령을 실행하여 직접 네트워크 인터페이스를 생성하고 Amazon EC2 호환 인스턴스에 연결합니다. 다음 단계에서는 디바이스의 MAC 주소가 필요합니다.

```
create-direct-network-interface [--endpoint endpoint] [--instance-id instanceId]
  [--mac macAddress]
  [--physical-network-interface-id physicalNetworkInterfaceId]
  [--unlock-code unlockCode] [--vlan vlanId]
```

### OPTIONS

**--endpoint <endpoint>** 이 요청을 보낼 엔드포인트입니다. 디바이스의 엔드포인트는 https 스키마 뒤에 IP 주소를 사용하는 URL이 됩니다. 예를 들어 디바이스의 IP 주소가 123.0.1.2인 경우 디바이스의 엔드포인트는 https://123.0.1.2가 됩니다.

**--instance-id <instanceId>** 인터페이스를 연결할 EC2 호환 인스턴스 ID입니다(선택 사항).

**--mac <macAddress>** 네트워크 인터페이스의 MAC 주소를 설정합니다(선택 사항).

**--physical-network-interface-id <physicalNetworkInterfaceId>** 새 가상 네트워크 인터페이스를 생성할 물리적 네트워크 인터페이스의 ID입니다. describe-device 명령을 사용하여 Snowball Edge에서 사용할 수 있는 물리적 네트워크 인터페이스를 결정할 수 있습니다.

**--vlan <vlanId>** 인터페이스에 할당된 VLAN을 설정합니다(선택 사항). 지정된 경우 인터페이스에서 전송되는 모든 트래픽에는 지정된 VLAN ID의 태그가 지정됩니다. 수신 트래픽은 지정된 VLAN ID로 필터링되며, 인스턴스로 전달되기 전에 모든 VLAN 태그가 제거됩니다.

2. DNI를 생성하여 EC2 호환 인스턴스에 연결한 후에는 Amazon EC2 호환 인스턴스 내에서 구성을 두 번 변경해야 합니다.

- 첫 번째는 EC2 호환 인스턴스와 연결된 VNI를 위한 패킷이 eth0을 통해 전송되도록 변경하는 것입니다.
- 두 번째는 부팅 시 DHCP 또는 고정 IP를 사용하도록 직접 네트워크 인터페이스를 구성하는 것입니다.

다음은 이러한 구성을 변경하는 Amazon Linux 2 및 CentOS Linux용 셸 스크립트의 예시입니다.

## Amazon Linux 2

```
# Mac address of the direct network interface.
# You got this when you created the direct network interface.
DNI_MAC=[MAC ADDRESS FROM CREATED DNI]

# Configure routing so that packets meant for the VNI always are sent through
eth0.
PRIVATE_IP=$(curl -s http://169.254.169.254/latest/meta-data/local-ipv4)
PRIVATE_GATEWAY=$(ip route show to match 0/0 dev eth0 | awk '{print $3}')
ROUTE_TABLE=10001
echo "from $PRIVATE_IP table $ROUTE_TABLE" > /etc/sysconfig/network-scripts/
rule-eth0
echo "default via $PRIVATE_GATEWAY dev eth0 table $ROUTE_TABLE" > /etc/
sysconfig/network-scripts/route-eth0
echo "169.254.169.254 dev eth0" >> /etc/sysconfig/network-scripts/route-eth0
```

```
# Query the persistent DNI name, assigned by udev via ec2net helper.
# changable in /etc/udev/rules.d/70-persistent-net.rules
DNI=$(ip --oneline link | grep -i $DNI_MAC | awk -F ':' '{ print $2 }')

# Configure DNI to use DHCP on boot.
cat << EOF > /etc/sysconfig/network-scripts/ifcfg-$DNI
DEVICE="$DNI"
NAME="$DNI"
HWADDR=$DNI_MAC
ONBOOT=yes
NOZEROCONF=yes
BOOTPROTO=dhcp
TYPE=Ethernet
MAINROUTETABLE=no
EOF

# Make all changes live.
systemctl restart network
```

## CentOS Linux

```
# Mac address of the direct network interface. You got this when you created the
direct network interface.
DNI_MAC=[MAC ADDRESS FROM CREATED DNI]
# The name to use for the direct network interface. You can pick any name that
isn't already in use.
DNI=eth1

# Configure routing so that packets meant for the VNIC always are sent through
eth0
PRIVATE_IP=$(curl -s http://169.254.169.254/latest/meta-data/local-ipv4)
PRIVATE_GATEWAY=$(ip route show to match 0/0 dev eth0 | awk '{print $3}')
ROUTE_TABLE=10001
echo from $PRIVATE_IP table $ROUTE_TABLE > /etc/sysconfig/network-scripts/rule-
eth0
echo default via $PRIVATE_GATEWAY dev eth0 table $ROUTE_TABLE > /etc/sysconfig/
network-scripts/route-eth0

# Configure your direct network interface to use DHCP on boot.
cat << EOF > /etc/sysconfig/network-scripts/ifcfg-$DNI
DEVICE="$DNI"
NAME="$DNI"
```

```

HWADDR="$DNI_MAC"
ONBOOT=yes
NOZEROCONF=yes
BOOTPROTO=dhcp
TYPE=Ethernet
EOF

# Rename DNI device if needed.
CURRENT_DEVICE_NAME=$(LANG=C ip -o link | awk -F ':' -vIGNORECASE=1 '!/link\/
ieee802\.\.11/ && /'"$DNI_MAC"'/ { print $2 })
ip link set $CURRENT_DEVICE_NAME name $DNI

# Make all changes live.
systemctl restart network

```

## SSH를 사용하여 Snowball Edge의 컴퓨팅 인스턴스에 연결

Secure Shell(SSH)을 사용하여 Snowball Edge의 컴퓨팅 인스턴스에 연결하려면 SSH 키를 제공하거나 생성하는 다음과 같은 옵션이 있습니다.

- 디바이스 주문 작업을 생성할 때 Amazon Machine Image(AMI)에 SSH 키를 제공할 수 있습니다. 자세한 내용은 [Snowball Edge 주문 작업 생성을 참조하세요](#).
- Snowball Edge로 가져올 가상 머신 이미지를 생성할 때 AMI의 SSH 키를 제공할 수 있습니다. 자세한 내용은 [Snowball Edge 디바이스로 가상 머신 이미지 가져오기](#) 단원을 참조하십시오.
- Snowball Edge에서 키 페어를 생성하고 로컬에서 생성된 퍼블릭 키로 인스턴스를 시작하도록 선택할 수 있습니다. 자세한 내용은 Amazon EC2 사용 설명서의 [Amazon EC2를 사용하여 키 페어 생성](#)을 참조하세요.

### SSH를 통해 EC2 인스턴스에 연결하려면

1. 디바이스에 전원이 인가되어 있고 네트워크에 연결되어 있으며 잠금 해제가 되어 있는지 확인합니다. 자세한 내용은 [로컬 네트워크에 Snowball Edge 연결](#) 단원을 참조하십시오.
2. 컴퓨팅 인스턴스에 대해 네트워크 설정이 구성되어 있는지 확인합니다. 자세한 내용은 [Snowball Edge의 컴퓨팅 인스턴스에 대한 네트워크 구성](#) 섹션을 참조하십시오.
3. 이러한 특정 인스턴스에 대해 사용한 PEM 또는 PPK 키 페어를 찾기 위해 메모를 확인합니다. 이 파일을 컴퓨터에 복사합니다. PEM 파일의 경로를 적어 둡니다.

- 다음 예시 명령과 같이 SSH를 통해 인스턴스에 연결합니다. IP 주소는 [Snowball Edge의 컴퓨팅 인스턴스에 대한 네트워크 구성](#)에서 설정한 가상 네트워크 인터페이스(VNIC)의 IP 주소입니다.

```
ssh -i path/to/PEM/key/file instance-user-name@192.0.2.0
```

자세한 내용은 Amazon EC2 사용 설명서의 [SSH를 사용하여 Linux 인스턴스에 연결](#)을 참조하세요.

## 동일한 Snowball Edge의 EC2-compatible 컴퓨팅 인스턴스에서 S3 버킷으로 데이터 전송

동일한 Snowball Edge 디바이스에서 컴퓨팅 인스턴스와 Amazon S3 버킷 간에 데이터를 전송할 수 있습니다. 지원되는 AWS CLI 명령과 적절한 엔드포인트를 사용하여이 작업을 수행할 수 있습니다. 예를 들어, 동일한 디바이스에서 내 sbe1.xlarge 인스턴스의 디렉터리에 있는 데이터를 Amazon S3 버킷인 amzn-s3-demo-bucket으로 이동하고 싶다고 가정합니다. Amazon S3 compatible storage on Snowball Edge 엔드포인트를 사용하고 있다고 가정합니다 `https://S3-object-API-endpoint:443`. 다음 절차를 수행하세요.

동일한 Snowball Edge에서 컴퓨팅 인스턴스와 버킷 간에 데이터 전송

- SSH를 사용하여 컴퓨팅 인스턴스에 연결합니다.
- 를 다운로드하여 설치합니다 AWS CLI. 인스턴스에 AWS CLI이 없는 경우 다운로드하여 설치합니다. 자세한 정보는 [AWS Command Line Interface 설치](#) 섹션을 참조하세요.
- Snowball Edge AWS CLI 의 Amazon S3 엔드포인트와 함께 작동하도록 컴퓨팅 인스턴스에서 구성합니다. 자세한 내용은 [Snowball Edge에서 로컬 Amazon S3 자격 증명 가져오기 및 사용](#) 단원을 참조하십시오.
- 지원되는 Amazon S3 compatible storage on Snowball Edge 명령을 사용하여 데이터를 전송합니다. 예:

```
aws s3 cp ~/june2018/results s3://amzn-s3-demo-bucket/june2018/results --recursive --endpoint https://S3-object-API-endpoint:443
```



## EC2-compatible 인스턴스 자동 시작

Snowball Edge 클라이언트는 환경에서 실행할 수 있는 독립형 명령줄 인터페이스(CLI) 애플리케이션입니다. 이를 통해 Snowball Edge 디바이스 또는 디바이스 클러스터에서 일부 관리 작업을 할 수 있습니다. 서비스를 시작 및 중지하는 방법을 비롯하여 Snowball Edge 클라이언트를 사용하는 방법에 대한 자세한 내용은 [Snowball Edge 클라이언트 구성 및 사용](#) 섹션을 참조하세요.

아래에서 사용 예시를 포함하여 컴퓨팅 인스턴스와 관련된 Snowball Edge 클라이언트 명령에 대한 정보를 찾을 수 있습니다.

AWS Snowball Edge 디바이스에서 사용할 수 있는 Amazon EC2-compatible 명령 목록은 섹션을 참조하세요 [Snowball Edge에서 지원되는 Amazon EC2-compatible AWS CLI 명령](#).

### Snowball Edge에서 EC2-compatible 시작 구성 생성

잠금을 해제한 후 AWS Snowball Edge 디바이스에서 Amazon EC2-compatible 컴퓨팅 인스턴스를 자동으로 시작하려면 시작 구성을 생성할 수 있습니다. 이를 위해서는 다음과 같이 `snowballEdge create-autostart-configuration` 명령을 사용합니다.

#### 사용량

```
snowballEdge create-autostart-configuration --physical-connector-type [SFP_PLUS or RJ45 or QSFP] --ip-address-assignment [DHCP or STATIC] [--static-ip-address-configuration IpAddress=[IP address],NetMask=[Netmask]] --launch-template-id [--launch-template-version]
```

### Snowball Edge에서 EC2-compatible 시작 구성 업데이트

Snowball Edge의 기존 시작 구성을 업데이트하려면 `snowballEdge update-autostart-configuration` 명령을 사용합니다. 다음과 같은 사용법을 찾아볼 수 있습니다. 시작 구성을 활성화 또는 비활성화하려면 `--enabled` 파라미터를 지정합니다.

#### 사용량

```
snowballEdge update-autostart-configuration --autostart-configuration-arn [--physical-connector-type [SFP_PLUS or RJ45 or QSFP]] [--ip-address-assignment [DHCP or STATIC]] [--static-ip-address-configuration IpAddress=[IP address],NetMask=[Netmask]][--launch-template-id] [--launch-template-version] [--enabled]
```

## Snowball Edge에서 EC2-compatible 시작 구성 삭제

더 이상 사용하지 않는 시작 구성을 삭제하려면 다음에 따라 `snowballEdge delete-autostart-configuration` 명령을 사용합니다.

### 사용량

```
snowballEdge delete-autostart-configuration --autostart-configuration-arn
```

## Snowball Edge에서 EC2-compatible 시작 구성 나열

Snowball Edge에서 생성한 시작 구성을 나열하려면 다음에 따라 `describe-autostart-configurations` 명령을 사용합니다.

### 사용량

```
snowballEdge describe-autostart-configurations
```

## Snowball Edge에서 가상 네트워크 인터페이스 생성

Snowball Edge에서 컴퓨팅 인스턴스를 실행하거나 NFS 인터페이스를 시작하려면 먼저 가상 네트워크 인터페이스(VNIC)를 생성해야 합니다. 각 Snowball Edge에는 이 디바이스의 물리적 네트워크 인터페이스 컨트롤러인 네트워크 인터페이스(NIC)가 세 개 있습니다. 디바이스 후면에 있는 RJ45, SFP, QSFP 포트가 바로 이 인터페이스입니다.

각 VNIC는 물리적 인터페이스에 기반을 두고 있으므로 각 NIC에 VNI를 얼마든지 연결할 수 있습니다. 가상 네트워크 인터페이스를 생성하려면 `snowballEdge create-virtual-network-interface` 명령을 사용하십시오.

### Note

`--static-ip-address-configuration` 파라미터는 `--ip-address-assignment` 파라미터에 대해 `STATIC` 옵션을 사용하는 경우에만 유효합니다.

### 사용법

Snowball Edge 클라이언트를 구성하거나 Snowball Edge 클라이언트를 구성하지 않는 두 가지 방법으로 이 명령을 사용할 수 있습니다. 다음 사용 예시는 Snowball Edge 클라이언트가 구성된 방법을 보여줍니다.

```
snowballEdge create-virtual-network-interface --ip-address-assignment [DHCP or STATIC]
--physical-network-interface-id [physical network interface id] --static-ip-address-
configuration IPAddress=[IP address],NetMask=[Netmask]
```

다음 사용 예시는 Snowball Edge 클라이언트가 구성되지 않은 방법을 보여줍니다.

```
snowballEdge create-virtual-network-interface --endpoint https://[ip address]
--manifest-file /path/to/manifest --unlock-code [unlock code] --ip-address-
assignment [DHCP or STATIC] --physical-network-interface-id [physical network interface
id] --static-ip-address-configuration IPAddress=[IP address],NetMask=[Netmask]
```

Example예시: VNIC 생성(DHCP 사용)

```
snowballEdge create-virtual-network-interface --ip-address-assignment dhcp --physical-
network-interface-id s.ni-8EXAMPLEaEXAMPLEd
{
  "VirtualNetworkInterface" : {
    "VirtualNetworkInterfaceArn" : "arn:aws:snowball-device:::interface/
s.ni-8EXAMPLE8EXAMPLEf",
    "PhysicalNetworkInterfaceId" : "s.ni-8EXAMPLEaEXAMPLEd",
    "IpAddressAssignment" : "DHCP",
    "IpAddress" : "192.0.2.0",
    "Netmask" : "255.255.255.0",
    "DefaultGateway" : "192.0.2.1",
    "MacAddress" : "EX:AM:PL:E1:23:45",
    "MtuSize" : "1500"
  }
}
```

## 가상 네트워크 인터페이스 설명

디바이스에서 앞서 생성한 VNIC를 설명하려면 `snowballEdge describe-virtual-network-interfaces` 명령을 사용합니다. 다음과 같은 사용법을 찾아볼 수 있습니다.

### 사용법

Snowball Edge 클라이언트를 구성하거나 Snowball Edge 클라이언트를 구성하지 않는 두 가지 방법으로 이 명령을 사용할 수 있습니다. 다음 사용 예시는 Snowball Edge 클라이언트가 구성된 방법을 보여줍니다.

```
snowballEdge describe-virtual-network-interfaces
```

다음 사용 예시는 Snowball Edge 클라이언트가 구성되지 않은 방법을 보여줍니다.

```
snowballEdge describe-virtual-network-interfaces --endpoint https://[ip address] --
manifest-file /path/to/manifest --unlock-code [unlock code]
```

### Example 예시: VNIC 설명

```
snowballEdge describe-virtual-network-interfaces
[
  {
    "VirtualNetworkInterfaceArn" : "arn:aws:snowball-device:::interface/
s.ni-8EXAMPLE8EXAMPLE8",
    "PhysicalNetworkInterfaceId" : "s.ni-8EXAMPLEaEXAMPLEd",
    "IpAddressAssignment" : "DHCP",
    "IpAddress" : "192.0.2.0",
    "Netmask" : "255.255.255.0",
    "DefaultGateway" : "192.0.2.1",
    "MacAddress" : "EX:AM:PL:E1:23:45",
    "MtuSize" : "1500"
  },{
    "VirtualNetworkInterfaceArn" : "arn:aws:snowball-device:::interface/
s.ni-1EXAMPLE1EXAMPLE1",
    "PhysicalNetworkInterfaceId" : "s.ni-8EXAMPLEaEXAMPLEd",
    "IpAddressAssignment" : "DHCP",
    "IpAddress" : "192.0.2.2",
    "Netmask" : "255.255.255.0",
    "DefaultGateway" : "192.0.2.1",
    "MacAddress" : "12:34:5E:XA:MP:LE",
    "MtuSize" : "1500"
  }
]
```

## Snowball Edge에서 가상 네트워크 인터페이스 업데이트

VNI(가상 네트워크 인터페이스)를 생성한 후에 snowballEdge update-virtual-network-interface 명령을 사용하여 해당 구성을 업데이트할 수 있습니다. 특정 VNI에 Amazon 리소스 이름(ARN)을 제공한 후 업데이트하려는 모든 요소에 대해 이 값만 제공하면 됩니다.

### 사용량

Snowball Edge 클라이언트를 구성하거나 Snowball Edge 클라이언트를 구성하지 않는 두 가지 방법으로 이 명령을 사용할 수 있습니다. 다음 사용 예시는 Snowball Edge 클라이언트가 구성된 방법을 보여줍니다.

```
snowballEdge update-virtual-network-interface --virtual-network-interface-arn [virtual network-interface-arn] --ip-address-assignment [DHCP or STATIC] --physical-network-interface-id [physical network interface id] --static-ip-address-configuration IpAddress=[IP address],NetMask=[Netmask]
```

다음 사용 예시는 Snowball Edge 클라이언트가 구성되지 않은 방법을 보여줍니다.

```
snowballEdge update-virtual-network-interface --endpoint https://[ip address] --manifest-file /path/to/manifest --unlock-code [unlock code] --virtual-network-interface-arn [virtual network-interface-arn] --ip-address-assignment [DHCP or STATIC] --physical-network-interface-id [physical network interface id] --static-ip-address-configuration IpAddress=[IP address],NetMask=[Netmask]
```

Example 예시: VNIC 업데이트(DHCP 사용)

```
snowballEdge update-virtual-network-interface --virtual-network-interface-arn arn:aws:snowball-device:::interface/s.ni-8EXAMPLEbEXAMPLEd --ip-address-assignment dhcp
```

## Snowball Edge에서 가상 네트워크 인터페이스 삭제

가상 네트워크 인터페이스(VNI)를 삭제하려면 `snowballEdge delete-virtual-network-interface` 명령을 사용하면 됩니다.

사용량

Snowball Edge 클라이언트를 구성하거나 Snowball Edge 클라이언트를 구성하지 않는 두 가지 방법으로 이 명령을 사용할 수 있습니다. 다음 사용 예시는 Snowball Edge 클라이언트가 구성된 방법을 보여줍니다.

```
snowballEdge delete-virtual-network-interface --virtual-network-interface-arn [virtual network-interface-arn]
```

다음 사용 예시는 Snowball Edge 클라이언트가 구성되지 않은 방법을 보여줍니다.

```
snowballEdge delete-virtual-network-interface --endpoint https://[ip address] --
manifest-file /path/to/manifest --unlock-code [unlock code] --virtual-network-
interface-arn [virtual network-interface-arn]
```

Example예시: VNIC 삭제

```
snowballEdge delete-virtual-network-interface --virtual-network-interface-arn
arn:aws:snowball-device:::interface/s.ni-8EXAMPLEbEXAMPLEd
```

## Snowball Edge에서 Amazon EC2-compatible 엔드포인트 사용

아래에서 Amazon EC2 호환 엔드포인트의 개요를 찾을 수 있습니다. 엔드포인트를 통해 Amazon Machine Image(AMI)를 관리하고 Amazon EC2 호환 API 작업을 사용하여 인스턴스를 프로그래밍 방식으로 컴퓨팅할 수 있습니다.

### EC2-compatible 엔드포인트를 Snowball Edge의 AWS CLI 엔드포인트로 지정

AWS CLI 를 사용하여 AWS Snowball Edge 디바이스에 명령을 실행할 때 엔드포인트를 Amazon EC2-compatible 엔드포인트로 지정할 수 있습니다. 다음과 같이 HTTPS 엔드포인트 또는 보안되지 않은 HTTP 엔드포인트를 사용할 수 있습니다.

HTTPS 보안 엔드포인트

```
aws ec2 describe-instances --endpoint https://192.0.2.0:8243 --ca-bundle path/to/
certificate
```

HTTP 보안되지 않는 엔드포인트

```
aws ec2 describe-instances --endpoint http://192.0.2.0:8008
```

HTTPS 엔드포인트 8243을 사용하는 경우 전송 중 데이터는 암호화됩니다. 이 암호화는 Snowball Edge가 잠금 해제될 때마다 생성하는 인증서로 보장됩니다. 인증서가 생기면 이를 로컬 `ca-bundle.pem` 파일에 저장할 수 있습니다. 그런 다음, 다음 설명과 같이 인증서 경로를 포함하도록 AWS CLI 프로필을 구성할 수 있습니다.

인증서를 Amazon EC2 호환 엔드포인트에 연결

1. Snowball Edge를 전원과 네트워크에 연결한 후 컵니다.

2. 디바이스의 잠금 해제가 완료되면 로컬 네트워크상의 IP 주소를 적어 둡니다.
3. 네트워크의 터미널에서 Snowball Edge를 ping할 수 있는지 확인합니다.
4. 터미널에서 snowballEdge get-certificate 명령을 실행합니다. 이 명령에 대한 자세한 내용은 [Snowball Edge에서 퍼블릭 키 인증서 관리](#) 섹션을 참조하세요.
5. snowballEdge get-certificate 명령의 출력을 파일(예: ca-bundle.pem)에 저장합니다.
6. 터미널에서 다음 명령을 실행합니다.

```
aws configure set profile.snowballEdge.ca_bundle /path/to/ca-bundle.pem
```

절차를 완료한 후에는 이러한 로컬 자격 증명, 인증서 및 지정된 엔드포인트로 CLI 명령을 실행할 수 있습니다.

## Snowball Edge에서 지원되는 Amazon EC2-compatible AWS CLI 명령

Amazon EC2-compatible 엔드포인트를 통해 Snowball Edge 디바이스에서 컴퓨팅 인스턴스를 관리할 수 있습니다. 이 유형의 엔드포인트는 AWS SDKs의 많은 Amazon EC2 CLI 명령 및 작업을 지원합니다. AWS CLI 호출할 대상 지정을 AWS CLI포함하여 설치 및 설정에 대한 자세한 내용은 [AWS Command Line Interface 사용 설명서](#)를 참조 AWS 리전 하세요.

### Snowball Edge에서 지원되는 EC2-compatible AWS CLI 명령 목록

아래에서 Snowball Edge 디바이스에서 지원되는 Amazon EC2의 AWS CLI 명령 및 옵션 하위 집합에 대한 설명을 확인할 수 있습니다. 아래에 나열되지 않은 명령이나 옵션은 지원되지 않습니다. 명령과 함께 지원되지 않는 옵션을 선언할 수 있습니다. 하지만 이러한 옵션은 무시됩니다.

- [associate-address](#) – 가상 IP 주소를 디바이스의 물리적 네트워크 인터페이스 3개 중 하나에 사용되는 인스턴스와 연결합니다.
  - --instance-id – 단일 sbe 인스턴스의 ID입니다.
  - --public-ip – 인스턴스에 액세스하기 위해 사용할 가상 IP 주소입니다.
- [attach-volume](#) – Amazon EBS 볼륨을 디바이스의 중지되거나 실행 중인 인스턴스에 연결하고 지정된 디바이스 이름이 있는 인스턴스에 공개합니다.
  - --device value - 디바이스의 이름입니다.
  - --instance-id - 대상 Amazon EC2 호환 인스턴스의 ID입니다.
  - --volume-id value - EBS 볼륨의 ID입니다.

- [authorize-security-group-egress](#) - 하나 이상의 송신 규칙을 Snowball Edge 디바이스에 사용할 보안 그룹에 추가합니다. 특히 이 작업을 통해 인스턴스는 하나 이상의 대상 IPv4 CIDR 주소 범위로 트래픽을 보낼 수 있습니다. 자세한 내용은 [Snowball Edge의 보안 그룹을 사용하여 네트워크 트래픽 제어](#) 단원을 참조하십시오.
  - `--group-id value` - 보안 그룹의 ID입니다.
  - `[--ip-permissions value]` - 하나 이상의 IP 권한 집합입니다.
- [authorize-security-group-ingress](#) - 하나 이상의 수신 규칙을 보안 그룹에 추가합니다. `authorize-security-group-ingress` 호출 시 `group-name` 또는 `group-id`에 대한 값을 지정해야 합니다.
  - `[--group-name value]` - 보안 그룹의 이름입니다.
  - `[--group-id value]` - 보안 그룹의 ID입니다.
  - `[--ip-permissions value]` - 하나 이상의 IP 권한 집합입니다.
  - `[--protocol value]` IP 프로토콜. 가능한 값은 `tcp`, `udp` 및 `icmp`입니다. "모든 프로토콜" 값이 지정되지 않는 한(-1) `--port` 인수가 필요합니다.
  - `[--port value]` - TCP 또는 UDP의 경우, 허용할 포트 범위입니다. 이 값은 단일 정수 또는 범위(최소~최대)일 수 있습니다.

ICMP의 경우, 단일 정수 또는 범위(`type~code`). 여기서 `type`은 ICMP 유형 번호를 나타내고 `code`는 ICMP 코드 번호를 나타냅니다. -1의 값은 모든 ICMP 유형에 대한 모든 ICMP 코드를 나타냅니다. `type`에만 해당되는 -1의 값은 지정된 ICMP 유형에 대한 모든 ICMP 코드를 나타냅니다.

- `[--cidr value]` - CIDR IP 범위입니다.
- [create-launch-template](#) - 시작 템플릿을 생성합니다. 시작 템플릿에는 인스턴스를 시작하기 위한 파라미터가 포함되어 있습니다. `RunInstances`를 사용하여 인스턴스를 시작하면 요청에 시작 파라미터를 제공하는 대신 시작 템플릿을 지정할 수 있습니다. 디바이스당 최대 100개의 템플릿을 생성할 수 있습니다.
  - `--launch-template-name string` - 시작 템플릿의 이름입니다.
  - `--launch-template-data structure` - 시작 템플릿에 대한 정보입니다. 다음과 같은 속성이 지원됩니다.
    - `ImageId`
    - `InstanceType`
    - `SecurityGroupIds`
    - `TagSpecifications`
    - `UserData`

JSON 구문:



```
{
  "ImageId":"string",
  "InstanceType":"sbe-c.large",
  "SecurityGroupIds":["string", ...],
  "TagSpecifications":[{"ResourceType":"instance","Tags":
[{"Key":"Name","Value":"Test"},
{"Key":"Stack","Value":"Gamma"}]},
  "UserData":"this is my user data"
}
```

- [--version-description string] - 시작 템플릿의 첫 번째 버전에 대한 설명입니다.
- --endpoint snowballEndpoint - Amazon EC2 호환 API 작업을 사용하여 프로그래밍 방식으로 컴퓨팅 인스턴스를 관리할 수 있는 값입니다. 자세한 내용은 [EC2-compatible 엔드포인트를 Snowball Edge의 AWS CLI 엔드포인트로 지정](#) 단원을 참조하십시오.
- [create-launch-template-version](#) - 시작 템플릿의 새 버전을 생성합니다. 새 버전을 기반으로 하는 시작 템플릿의 기존 버전을 지정할 수 있습니다. 시작 템플릿 버전은 생성한 순서대로 번호가 지정됩니다. 시작 템플릿 버전의 숫자는 지정, 변경 또는 대체할 수 없습니다. 각 시작 템플릿의 버전을 최대 100개까지 생성할 수 있습니다.

요청에서 시작 템플릿 ID 또는 시작 템플릿 이름을 지정합니다.

- --launch-template-id string - 시작 템플릿의 ID입니다.
- --launch-template-name string - 시작 템플릿의 이름입니다.
- --launch-template-data structure - 시작 템플릿에 대한 정보입니다. 다음과 같은 속성이 지원됩니다.
  - ImageId
  - InstanceType
  - SecurityGroupIds
  - TagSpecifications
  - UserData

JSON 구문:

```
{
  "ImageId":"string",
  "InstanceType":"sbe-c.large",
  "SecurityGroupIds":["string", ...],
```

```

"TagSpecifications":[{"ResourceType":"instance","Tags":
[{"Key":"Name","Value":"Test"},
{"Key":"Stack","Value":"Gamma"}]},
"UserData":"this is my user data"
}

```

- `--source-version string` - 새 버전을 기반으로 하는 시작 템플릿의 버전 번호입니다. 새 버전은 `launch-template-data`에 지정된 파라미터를 제외하고 소스 버전과 동일한 시작 파라미터를 상속합니다.
- `--version-description string` - 시작 템플릿의 첫 번째 버전에 대한 설명입니다.
- `--endpoint snowballEndpoint` - Amazon EC2 호환 API 작업을 사용하여 프로그래밍 방식으로 컴퓨팅 인스턴스를 관리할 수 있는 값입니다. 자세한 내용은 [EC2-compatible 엔드포인트를 Snowball Edge의 AWS CLI 엔드포인트로 지정](#) 단원을 참조하십시오.
- [create-tags](#) - 지정된 리소스에 대해 하나 이상의 태그를 추가하거나 덮어씁니다. 각 리소스는 최대 50개의 태그를 보유할 수 있습니다. 각 태그는 키와 값(선택 사항)으로 구성됩니다. 태그 키는 리소스에 대해 고유해야 합니다. 다음 리소스가 지원됩니다.
  - AMI
  - Instance
  - 시작 템플릿
  - 보안 그룹
  - 키 페어
- [create-security-group](#) - Snowball Edge에 보안 그룹을 생성합니다. 최대 50개의 보안 그룹을 생성할 수 있습니다. 보안 그룹을 생성할 때 친근한 이름을 직접 선택하여 지정합니다.
  - `--group-name value` - 보안 그룹의 이름입니다.
  - `--description value` - 보안 그룹의 설명입니다. 이는 참조용일 뿐입니다. 이 값의 최대 길이는 255 자입니다.
- [create-volume](#) - 디바이스의 인스턴스에 연결할 수 있는 EBS 볼륨을 생성합니다.
  - `--size value` - GiB로 볼륨 크기를 표현하며 크기 범위는 1GiB에서 1TB(1,000GiB)까지입니다.
  - `--snapshot-id value` - 볼륨이 생성되는 스냅샷의 ID입니다.
  - `--volume-type value` - 볼륨 유형입니다. 값을 지정하지 않을 경우 기본값은 `sbg1`입니다. 가능한 값은 다음을 포함합니다.
    - 마그네틱 볼륨인 `sbg1`
    - SSD 볼륨인 `sbp1`

- [delete-launch-template](#) - 시작 템플릿을 삭제합니다. 시작 템플릿을 삭제하면 모든 버전이 삭제됩니다.

요청에서 시작 템플릿 ID 또는 시작 템플릿 이름을 지정합니다.

- `--launch-template-id string` - 시작 템플릿의 ID입니다.
- `--launch-template-name string` - 시작 템플릿의 이름입니다.
- `--endpoint snowballEndpoint` - Amazon EC2 호환 API 작업을 사용하여 프로그래밍 방식으로 컴퓨팅 인스턴스를 관리할 수 있는 값입니다. 자세한 내용은 [EC2-compatible 엔드포인트를 Snowball Edge의 AWS CLI 엔드포인트로 지정](#) 단원을 참조하십시오.
- [delete-launch-template-version](#) - 시작 템플릿의 버전을 하나 이상 삭제합니다. 시작 템플릿의 기본 버전은 삭제할 수 없으며, 먼저 다른 버전을 기본 버전으로 할당해야 합니다. 기본 버전이 시작 템플릿의 유일한 버전인 경우 `delete-launch-template` 명령을 사용하여 전체 시작 템플릿을 삭제합니다.

요청에서 시작 템플릿 ID 또는 시작 템플릿 이름을 지정합니다.

- `--launch-template-id string` - 시작 템플릿의 ID입니다.
- `--launch-template-name string` - 시작 템플릿의 이름입니다.
- `--versions (list) "string" "string"` - 삭제할 하나 이상의 시작 템플릿 버전에 대한 버전 번호입니다.
- `--endpoint snowballEndpoint` - Amazon EC2 호환 API 작업을 사용하여 프로그래밍 방식으로 컴퓨팅 인스턴스를 관리할 수 있는 값입니다. 자세한 내용은 [EC2-compatible 엔드포인트를 Snowball Edge의 AWS CLI 엔드포인트로 지정](#) 단원을 참조하십시오.
- [delete-security-group](#) - 보안 그룹을 삭제합니다.

인스턴스와 연결된 보안 그룹이나 다른 보안 그룹에서 참조하는 보안 그룹의 삭제를 시도하는 경우 `DependencyViolation`으로 작업이 실패합니다.

- `--group-name value` - 보안 그룹의 이름입니다.
- `--description value` - 보안 그룹의 설명입니다. 이는 참조용일 뿐입니다. 이 값의 최대 길이는 255 자입니다.
- [delete-tags](#) - 지정된 리소스(AMI, 컴퓨팅 인스턴스, 시작 템플릿 또는 보안 그룹)에서 지정된 태그 세트를 삭제합니다.
- [delete-volume](#) - 지정된 Amazon EBS 볼륨을 삭제합니다. 이때 볼륨은 `available` 상태(인스턴스에 연결되지 않음)여야 합니다.
- `--volume-id value` - 볼륨의 ID입니다.

- [describe-address](#) - 디바이스에서 동일한 수의 sbe 인스턴스와 관련된 하나 이상의 가상 IP 주소를 설명합니다.
  - --public-ips - 인스턴스와 연결된 한 개 이상의 가상 IP 주소입니다.
- [describe-images](#) - 사용 가능한 하나 이상의 이미지(AMI)를 설명합니다. 사용 가능한 이미지는 작업 생성 중에 Snowball Edge 디바이스에 추가됩니다.
  - --image-id - AMI의 Snowball AMI ID입니다.
- [describe-instance-attribute](#) - 지정된 인스턴스의 지정된 속성을 설명합니다. 한 번에 하나의 속성만 지정할 수 있습니다. 다음과 같은 속성이 지원됩니다.
  - instanceInitiatedShutdownBehavior
  - instanceType
  - userData
- [describe-instances](#) - 하나 이상의 인스턴스를 설명합니다. 응답은 인스턴스에 할당된 보안 그룹을 반환합니다.
  - --instance-ids - 디바이스에서 중지된 하나 이상의 sbe 인스턴스 ID입니다.
  - --page-size - 직접 호출을 받아들일 각 페이지의 크기입니다. 이 값은 명령의 출력을 통해 반환되는 항목의 수에는 영향을 미치지 않습니다. 페이지 크기를 더 작게 설정하면 디바이스에 대한 추가 호출이 이루어져 각 호출에서 더 적은 항목을 가져옵니다. 따라서 호출이 시간 초과되는 것을 방지하는 데 도움이 됩니다.
  - --max-items - 명령의 출력을 통해 반환될 항목의 총 수입니다. 사용 가능한 총 항목 수가 지정된 값을 초과하는 경우 명령의 출력에 NextToken이 제공됩니다. 페이지 매김을 재개하려면 후속 명령의 starting-token 인수에 NextToken 값을 제공합니다.
  - --starting-token - 페이지 매김을 시작할 위치를 지정하기 위한 토큰입니다. 이 토큰은 이전에 잘린 응답에서 도출된 NextToken 값입니다.
- [describe-instance-status](#) - 지정된 인스턴스 또는 모든 인스턴스의 상태를 설명합니다. 모든 인스턴스의 상태를 반환하도록 특별히 지정하지 않는 한 기본적으로 실행 중인 인스턴스만 설명됩니다. 인스턴스 상태는 다음 구성 요소를 포함합니다.
  - 상태 확인 - Snow 디바이스는 Amazon EC2 호환 인스턴스에서 상태 확인을 수행하여 하드웨어 및 소프트웨어 문제를 식별합니다.
  - 인스턴스 상태 - 인스턴스를 시작하는 순간부터 종료까지 인스턴스를 관리할 수 있습니다.

이 명령에서는 다음 필터가 지원됩니다.

- [--filters] (list)

## 필터.

- `instance-state-code` - 인스턴스 상태의 코드(16비트 부호 없는 정수). 상위 바이트는 내부 서비스 보고 용도로 사용되므로 무시해야 합니다. 하위 바이트는 표시된 상태를 기반으로 설정됩니다. 유효한 값은 0(보류 중), 16(실행 중), 32(종료 중), 48(종료됨), 64(중지 중), 80(중지됨)입니다.
- `instance-state-name` - 인스턴스의 상태입니다(`pending` | `running` | `shutting-down` | `terminated` | `stopping` | `stopped`).
- `instance-status.reachability` - 이름이 `reachability`인 인스턴스 상태를 기준으로 필터링합니다(`passed` | `failed` | `initializing` | `insufficient-data`).
- `instance-status.status` - 인스턴스의 상태입니다(`ok` | `impaired` | `initializing` | `insufficient-data` | `not-applicable`).
- `system-status.reachability` - 이름이 연결 가능 여부인 시스템 상태를 필터링합니다(`passed` | `failed` | `initializing` | `insufficient-data`).
- `system-status.status` - 인스턴스의 시스템 상태입니다(`ok` | `impaired` | `initializing` | `insufficient-data` | `not-applicable`).
- JSON 구문.

```
[
  {
    "Name": "string",
    "Values": ["string", ...]
  }
  ...
]
```

- `[--instance-ids]` (list)

인스턴스 ID.

기본값: 모든 인스턴스에 대해 설명합니다.

- `[--dry-run | --no-dry-run]`(부울)

실제로 요청하지 않고 작업에 필요한 권한이 있는지 확인하고 오류 응답을 제공합니다. 필요한 권한이 있는 경우 오류 응답은 `DryRunOperation`입니다

그렇지 않으면 `UnauthorizedOperation`입니다.

- `[--include-all-instances | --no-include-all-instances]`(부울)

true 상태일 경우 모든 인스턴스의 상태를 포함합니다. false 상태일 경우 실행 중인 인스턴스의 상태만을 포함합니다.

기본값: false

- `--page-size`(정수) - 직접 호출을 받아들일 각 페이지의 크기입니다. 이 값은 명령의 출력을 통해 반환되는 항목의 수에는 영향을 미치지 않습니다. 페이지 크기를 더 작게 설정하면 디바이스에 대한 추가 호출이 이루어져 각 호출에서 더 적은 항목을 가져옵니다. 따라서 호출이 시간 초과되는 것을 방지하는 데 도움이 됩니다.
- `--max-items`(정수) - 명령의 출력에서 반환되는 항목의 총 수입니다. 사용 가능한 총 항목 수가 지정된 값을 초과하는 경우 명령의 출력에 NextToken이 제공됩니다. 페이지 매김을 재개하려면 후속 명령의 starting-token 인수에 NextToken 값을 제공합니다.
- `--starting-token`(문자열) - 페이지 매김을 시작할 위치를 지정하는 토큰입니다. 이 토큰은 이전에 잘린 응답에서 도출된 NextToken 값입니다.
- [describe-launch-templates](#) - 하나 이상의 시작 템플릿을 설명합니다. describe-launch-templates 명령은 페이지 매김 작업입니다. 다중 호출을 생성하여 결과의 전체 데이터 세트를 검색할 수 있습니다.

요청에서 시작 템플릿 ID 또는 시작 템플릿 이름을 지정합니다.

- `--launch-template-ids` (list) "string" "string" - 시작 템플릿의 ID 목록입니다.
- `--launch-template-names` (list) "string" "string" - 시작 템플릿의 이름 목록입니다.
- `--page-size` - 직접 호출을 받아들일 각 페이지의 크기입니다. 이 값은 명령의 출력을 통해 반환되는 항목의 수에는 영향을 미치지 않습니다. 페이지 크기를 더 작게 설정하면 디바이스에 대한 추가 호출이 이루어져 각 호출에서 더 적은 항목을 가져옵니다. 따라서 호출이 시간 초과되는 것을 방지하는 데 도움이 됩니다.
- `--max-items` - 명령의 출력을 통해 반환될 항목의 총 수입니다. 사용 가능한 총 항목 수가 지정된 값을 초과하는 경우 명령의 출력에 NextToken이 제공됩니다. 페이지 매김을 재개하려면 후속 명령의 starting-token 인수에 NextToken 값을 제공합니다.
- `--starting-token` - 페이지 매김을 시작할 위치를 지정하기 위한 토큰입니다. 이 토큰은 이전에 잘린 응답에서 도출된 NextToken 값입니다.
- `--endpoint snowballEndpoint` - Amazon EC2 호환 API 작업을 사용하여 프로그래밍 방식으로 컴퓨팅 인스턴스를 관리할 수 있는 값입니다. 자세한 내용은 [EC2-compatible 엔드포인트를 Snowball Edge의 AWS CLI 엔드포인트로 지정](#) 단원을 참조하십시오.
- [describe-launch-template-versions](#) - 지정된 시작 템플릿의 버전을 하나 이상 설명합니다. 모든 버전, 개별 버전 또는 특정 버전 범위를 설명할 수 있습니다. describe-launch-template-versions

명령은 페이지 매김 작업입니다. 다중 호출을 생성하여 결과의 전체 데이터 세트를 검색할 수 있습니다.

요청에서 시작 템플릿 ID 또는 시작 템플릿 이름을 지정합니다.

- `--launch-template-id string` - 시작 템플릿의 ID입니다.
- `--launch-template-name string` - 시작 템플릿의 이름입니다.
- `[--versions (list) "string" "string"]` - 삭제할 하나 이상의 시작 템플릿 버전에 대한 버전 번호입니다.
- `[--min-version string]` - 시작 템플릿 버전을 설명하는 최소 버전 번호입니다.
- `[--max-version string]` - 시작 템플릿 버전을 설명하는 최대 버전 번호입니다.
- `--page-size` - 직접 호출을 받아들일 각 페이지의 크기입니다. 이 값은 명령의 출력을 통해 반환되는 항목의 수에는 영향을 미치지 않습니다. 페이지 크기를 더 작게 설정하면 디바이스에 대한 추가 호출이 이루어져 각 호출에서 더 적은 항목을 가져옵니다. 따라서 호출이 시간 초과되는 것을 방지하는 데 도움이 됩니다.
- `--max-items` - 명령의 출력을 통해 반환될 항목의 총 수입니다. 사용 가능한 총 항목 수가 지정된 값을 초과하는 경우 명령의 출력에 NextToken이 제공됩니다. 페이지 매김을 재개하려면 후속 명령의 `starting-token` 인수에 NextToken 값을 제공합니다.
- `--starting-token` - 페이지 매김을 시작할 위치를 지정하기 위한 토큰입니다. 이 토큰은 이전에 잘린 응답에서 도출된 NextToken 값입니다.
- `--endpoint snowballEndpoint` - Amazon EC2 호환 API 작업을 사용하여 프로그래밍 방식으로 컴퓨팅 인스턴스를 관리할 수 있는 값입니다. 자세한 내용은 [EC2-compatible 엔드포인트를 Snowball Edge의 AWS CLI 엔드포인트로 지정](#) 단원을 참조하십시오.
- [describe-security-groups](#) - 하나 이상의 보안 그룹을 설명합니다.

`describe-security-groups` 명령은 페이지 매김 작업입니다. 다중 API 직접 호출을 발행하여 결과의 전체 데이터 세트를 검색할 수 있습니다.

- `[--group-name value]` - 보안 그룹의 이름입니다.
- `[--group-id value]` - 보안 그룹의 ID입니다.
- `[--page-size value]` - AWS 서비스 호출에서 가져올 각 페이지의 크기입니다. 이 크기는 명령의 출력에서 반환된 항목의 수에 영향을 주지 않습니다. 페이지 크기를 작게 설정하면 AWS 서비스에 대한 직접 호출이 많아지고 각 직접 호출의 항목에서 검색 항목이 줄어듭니다. 이 접근 방식은 AWS 서비스 호출이 시간 초과되는 것을 방지하는 데 도움이 될 수 있습니다. 사용 예시는 AWS Command Line Interface 사용 설명서의 [페이지 매김](#) 섹션을 참조하세요.

- [--max-items value] - 명령의 출력을 통해 반환될 항목의 총 수입니다. 사용 가능한 총 항목 수가 지정된 값을 초과하는 경우 명령의 출력에 NextToken이 제공됩니다. 페이지 매김을 재개하려면 후속 명령의 starting-token 인수에 NextToken 값을 제공합니다. NextToken 외부에서 직접 AWS CLI 응답 요소를 사용하면 안 됩니다. 사용 예시는 AWS Command Line Interface 사용 설명서의 [페이지 매김](#) 섹션을 참조하세요.
- [--starting-token value] - 페이지 매김이 시작되는 곳을 지정하는 토큰입니다. 이 토큰은 이전에 잘린 응답에서 도출된 NextToken 값입니다. 사용 예시는 AWS Command Line Interface 사용 설명서의 [페이지 매김](#) 섹션을 참조하세요.
- [describe-tags](#) - 지정된 리소스(image, instance 또는 보안 그룹)에 대한 하나 이상의 태그를 설명합니다. 이 명령에서는 다음 필터가 지원됩니다.
  - launch-template
  - resource-id
  - resource-type - image 또는 instance입니다.
  - 키
  - 값
- [describe-volumes](#) - 지정된 EBS 볼륨을 설명합니다.
  - [--max-items value] - 명령의 출력을 통해 반환될 항목의 총 수입니다. 사용 가능한 총 항목 수가 지정된 값을 초과하는 경우 명령의 출력에 NextToken이 제공됩니다. 페이지 매김을 재개하려면 후속 명령의 starting-token 인수에 NextToken 값을 제공합니다.
  - [--starting-token value] - 페이지 매김이 시작되는 곳을 지정하는 토큰입니다. 이 토큰은 이전에 잘린 응답에서 도출된 NextToken 값입니다.
  - [--volume-ids value] - 하나 이상의 볼륨 ID입니다.
- [detach-volume](#) - 중지되거나 실행 중인 인스턴스에서 Amazon EBS 볼륨을 분리합니다.
  - [--device value] - 디바이스의 이름입니다.
  - [--instance-id] - 대상 Amazon EC2 인스턴스의 ID입니다.
  - --volume-id value - 볼륨의 ID입니다.
- [disassociate-address](#) - 가상 IP 주소를 연결된 인스턴스에서 분리합니다.
  - --public-ip - 인스턴스에서 분리하려고 하는 가상 IP 주소입니다.
- [get-launch-template-data](#) - 지정된 인스턴스의 구성 데이터를 검색합니다. 이 데이터를 사용하여 시작 템플릿을 생성할 수 있습니다.
  - --instance-id - 단일 sbe 인스턴스의 ID입니다.



- `--endpoint snowballEndpoint` - Amazon EC2 호환 API 작업을 사용하여 프로그래밍 방식으로 컴퓨팅 인스턴스를 관리할 수 있는 값입니다. 자세한 내용은 [EC2-compatible 엔드포인트를 Snowball Edge의 AWS CLI 엔드포인트로 지정](#) 단원을 참조하십시오.
- `modify-launch-template` - 시작 템플릿을 수정합니다. 기본 버전으로 설정할 시작 템플릿의 버전을 지정할 수 있습니다. 시작 템플릿 버전을 지정하지 않고 인스턴스를 시작하면 시작 템플릿의 기본 버전이 적용됩니다.

요청에서 시작 템플릿 ID 또는 시작 템플릿 이름을 지정합니다.

- `--launch-template-id string` - 시작 템플릿의 ID입니다.
- `--launch-template-name string` - 시작 템플릿의 이름입니다.
- `--default-version string` - 기본 버전으로 설정할 시작 템플릿의 버전 번호입니다.
- `--endpoint snowballEndpoint` - Amazon EC2 호환 API 작업을 사용하여 프로그래밍 방식으로 컴퓨팅 인스턴스를 관리할 수 있는 값입니다. 자세한 내용은 [EC2-compatible 엔드포인트를 Snowball Edge의 AWS CLI 엔드포인트로 지정](#) 단원을 참조하십시오.
- `modify-instance-attribute` - 지정된 인스턴스의 속성을 수정합니다. 다음과 같은 속성이 지원됩니다.
  - `instanceInitiatedShutdownBehavior`
  - `userData`
- `revoke-security-group-egress` - 하나 이상의 송신 규칙을 보안 그룹에서 제거합니다.
  - `[--group-id value]` - 보안 그룹의 ID입니다.
  - `[--ip-permissions value]` - 하나 이상의 IP 권한 집합입니다.
- `revoke-security-group-ingress` - 보안 그룹에 대한 하나 이상의 수신 규칙을 해지합니다. `revoke-security-group-ingress` 호출 시 `group-name` 또는 `group-id`에 대한 값을 지정해야 합니다.
  - `[--group-name value]` - 보안 그룹의 이름입니다.
  - `[--group-id value]` - 보안 그룹의 ID입니다.
  - `[--ip-permissions value]` - 하나 이상의 IP 권한 집합입니다.
  - `[--protocol value]` IP 프로토콜. 가능한 값은 `tcp`, `udp` 및 `icmp`입니다. "모든 프로토콜" 값이 지정되지 않는 한(-1) `--port` 인수가 필요합니다.
  - `[--port value]` - TCP 또는 UDP의 경우, 허용할 포트 범위입니다. 단일 정수 또는 범위(최소~최대 값)입니다.

ICMP의 경우, 단일 정수 또는 범위(`type~code`). 여기서 `type`은 ICMP 유형 번호를 나타내고 `code`는 ICMP 코드 번호를 나타냅니다. -1의 값은 모든 ICMP 유형에 대한 모든 ICMP 코드를 나타냅니다. `type`에만 해당되는 -1의 값은 지정된 ICMP 유형에 대한 모든 ICMP 코드를 나타냅니다.

- [--cidr value] - CIDR IP 범위입니다.
- [run-instances](#) - AMI에 대해 Snowball AMI ID를 사용하여 많은 컴퓨팅 인스턴스를 시작합니다.

#### Note

컴퓨팅 인스턴스의 크기 및 유형에 따라 Snowball Edge에서 컴퓨팅 인스턴스를 시작하는 데 최대 한 시간 반이 소요될 수 있습니다.

- [--block-device-mappings (list)] - 블록 디바이스 매핑 항목입니다. DeleteOnTermination, VolumeSize, VolumeType 파라미터를 지원합니다. 부트 볼륨은 sbg1 유형이어야 합니다.

이 명령어의 JSON 구문은 다음과 같습니다.

```
{
  "DeviceName": "/dev/sdh",
  "Ebs":
  {
    "DeleteOnTermination": true|false,
    "VolumeSize": 100,
    "VolumeType": "sbp1"|"sbg1"
  }
}
```

- --count - 시작할 인스턴스의 개수입니다. 단일 숫자가 제공된 경우 최소 개수를 시작하는 것으로 가정합니다(1로 기본 설정). 범위가 min:max 형식으로 제공될 경우 첫 번째 수는 시작할 인스턴스의 최소 수로 해석하고 두 번째 수는 시작할 인스턴스의 최대 수로 해석합니다.
- --image-id - describe-images를 직접 호출하여 얻을 수 있는 AMI의 Snowball AMI ID입니다. AMI는 인스턴스를 시작하는 데 필요합니다.
- --InstanceInitiatedShutdownBehavior - 기본적으로 shutdown 또는 poweroff 등의 명령을 사용하여 인스턴스에서 종료를 시작하면 인스턴스가 중지됩니다. 인스턴스가 중지되지 않고 종료되도록 이 동작을 변경할 수 있습니다. stop 및 terminate 파라미터가 지원됩니다. 기본값은 stop입니다. 자세한 정보는 Linux 인스턴스용 Amazon EC2 사용 설명서의 [인스턴스가 시작하는 종료 동작 변경](#) 섹션을 참조하세요.
- --instance-type - sbe 인스턴스 유형입니다.
- --launch-template structure - 인스턴스를 시작하는 데 사용할 시작 템플릿입니다. run-instances 명령에서 파라미터를 지정하면 시작 템플릿에 있는 동일한 파라미터가 재정의됩니다. 시작 템플릿의 이름이나 ID는 지정할 수 있지만, 둘 다 지정할 수는 없습니다.

```
{
  "LaunchTemplateId": "string",
  "LaunchTemplateName": "string",
  "Version": "string"
}
```

- `--security-group-ids` - 하나 이상의 보안 그룹 ID입니다. [CreateSecurityGroup](#)을 사용하여 보안 그룹을 생성할 수 있습니다. 값이 제공되지 않는 경우 기본 보안 그룹의 ID가 생성된 인스턴스에 할당됩니다.
- `--tag-specifications` - 시작 시 리소스에 적용되는 태그입니다. 시작 시에만 인스턴스에 태그를 지정할 수 있습니다. 지정된 태그는 시작 시 생성된 모든 인스턴스에 적용됩니다. 생성된 후에 리소스에 태그를 지정하려면 `create-tags`를 사용합니다.
- `--user-data` - 인스턴스에 사용할 수 있는 사용자 데이터입니다. `r`를 사용하는 경우 AWS CLI base64 인코딩이 자동으로 수행되며 파일에서 텍스트를 로드할 수 있습니다. 그렇지 않으면 base64 인코딩 형식의 텍스트를 입력해야 합니다.
- `--key-name (string)` - 키 페어의 이름입니다. `CreateKeyPair` 또는 `ImportKeyPair`를 사용하여 키 페어를 생성할 수 있습니다.

#### Warning

키 페어를 지정하지 않는 경우에는 사용자가 다른 방법으로 로그인할 수 있도록 구성된 AMI를 선택해야만 인스턴스에 연결할 수 있습니다.

- [start-instances](#) - 이전에 중지했던 sbe 인스턴스를 시작합니다. 인스턴스에 연결된 모든 자원은 시작 및 중지를 통해 지속되지만 인스턴스가 해지되면 삭제됩니다.
  - `--instance-ids` - 디바이스에서 중지된 하나 이상의 sbe 인스턴스 ID입니다.
- [stop-instances](#) - 실행 중인 sbe 인스턴스를 중지합니다. 인스턴스에 연결된 모든 자원은 시작 및 중지를 통해 지속되지만 인스턴스가 해지되면 삭제됩니다.
  - `--instance-ids` - 디바이스에서 중지된 하나 이상의 sbe 인스턴스 ID입니다.
- [terminate-instances](#) - 하나 이상의 인스턴스를 종료합니다. 이 작업은 멍등성을 갖기 때문에 인스턴스를 2회 이상 종료해도 모든 호출이 성공적으로 처리됩니다. 인스턴스에 연결된 모든 리소스는 시작 및 중지 사태가 발생해도 지속되지만 인스턴스가 종료되면 데이터가 삭제됩니다.

**Note**

기본적으로 shutdown 또는 poweroff 등의 명령을 사용하여 인스턴스에서 종료를 시작하면 인스턴스가 중지됩니다. 그러나 이러한 명령이 인스턴스를 종료하도록 InstanceInitiatedShutdownBehavior 속성을 사용하여 이 동작을 변경할 수 있습니다. 자세한 정보는 Linux 인스턴스용 Amazon EC2 사용 설명서의 [인스턴스가 시작하는 종료 동작 변경](#) 섹션을 참조하세요.

- --instance-ids - 디바이스에서 종료된 하나 이상의 sbe 인스턴스 ID입니다. 이러한 인스턴스에 대해 저장된 연결된 데이터는 모두 손실됩니다.
- [create-key-pair](#) - 지정된 이름으로 2048비트 RSA 키 페어를 생성합니다. Amazon EC2는 퍼블릭 키를 저장하며 파일에 저장할 수 있는 프라이빗 키를 표시합니다. 프라이빗 키는 암호화되지 않은 PEM 인코딩 PKCS#1 프라이빗 키로 반환됩니다. 지정된 이름의 키 페어가 이미 있는 경우 Amazon EC2에서 오류가 발생합니다.

- --key-name (string) - 키 페어의 고유 이름입니다.

제약 조건: ASCII 문자 최대 255자.

- [--tag-specifications] (list) - 새 키 페어에 적용할 태그입니다.

```
{
  "ResourceType": "image|"instance|"key-pair|"launch-template|"security-group",
  "Tags": [
    {
      "Key": "string",
      "Value": "string"
    }
    ...
  ]
}
...
```

- [import-key-pair](#) -

- --key-name (string) - 키 페어의 고유 이름입니다.

제약 조건: ASCII 문자 최대 255자.

- --public-key-material (blob) - 퍼블릭 키입니다. API 직접 호출의 경우 텍스트는 base64로 인코딩되어야 합니다. 명령줄 도구의 경우 base64-encoding이 수행됩니다.

- `--tag-specifications` (list) – 새 키 페어에 적용할 태그입니다.

```
{
  "ResourceType": "image|"instance|"key-pair|"launch-template|"security-group",
  "Tags": [
    {
      "Key": "string",
      "Value": "string"
    }
    ...
  ]
}
```

- [describe-key-pairs](#) –

`--filters` (list) – 필터입니다.

- `key-pair-id` – 키 페어의 ID입니다.
- `key-name` – 키 페어의 이름입니다.
- `tag-key` – 리소스에 할당된 태그의 키입니다. 이 필터를 사용하면 태그 값에 상관없이 특정 키가 있는 태그가 할당된 모든 리소스를 찾을 수 있습니다.
- `--tag-specifications` (list) – 새 키 페어에 적용할 태그입니다.
- `tag:key` – 리소스에 할당된 태그의 키/값 조합입니다. 필터 이름에 태그 키를 사용하고 태그 값을 필터 값으로 사용합니다. 예를 들어, `Owner` 키와 `Team A` 값이 있는 태그가 있는 모든 리소스를 찾으려면 필터 이름에 `tag:Owner`를 지정하고 필터 값에 `Team A`를 지정합니다.

```
{
  "Name": "string",
  "Values": ["string", ...]
}
...
```

- `--key-names` (list) – 키 페어 이름입니다.

기본값: 모든 키 페어를 설명합니다.

- `--key-pair-ids` (list) – 키 페어의 ID입니다.
- [delete-key-pair](#) –
  - `--key-name` (string) – 키 페어의 이름입니다.
  - `--key-pair-id` (string) – 키 페어의 ID입니다.

## Snowball Edge에서 지원되는 Amazon EC2-compatible API 작업

이제 Amazon EC2 API Reference의 설명 링크와 함께 Snowball Edge에서 사용할 수 있는 Amazon EC2 호환 API 작업을 찾을 수 있습니다. Amazon EC2 호환 API 직접 호출에는 Signature Version 4(SigV4) 서명이 필요합니다. AWS CLI 또는 AWS SDK를 사용하여 이러한 API 호출을 수행하는 경우 SigV4 서명이 처리됩니다. 그렇지 않은 경우 자체 SigV4 서명 솔루션을 구현해야 합니다. 자세한 내용은 [Snowball Edge에서 로컬 Amazon S3 자격 증명 가져오기 및 사용](#) 단원을 참조하십시오.

- [AssociateAddress](#) - 인스턴스 또는 네트워크 인터페이스에 탄력적 IP 주소를 연결합니다.
- [AttachVolume](#) - 다음 요청 파라미터가 지원됩니다.
  - Device
  - InstanceId
  - VolumeId
- [AuthorizeSecurityGroupEgress](#) - 하나 이상의 송신 규칙을 Snowball Edge 디바이스에 사용할 보안 그룹에 추가합니다. 특히 이 작업을 통해 인스턴스는 하나 이상의 대상 IPv4 CIDR 주소 범위로 트래픽을 보낼 수 있습니다.
- [AuthorizeSecurityGroupIngress](#) - 하나 이상의 수신 규칙을 보안 그룹에 추가합니다. AuthorizeSecurityGroupIngress 호출 시 GroupName 또는 GroupId에 대한 값을 지정해야 합니다.
- [CreateVolume](#) - 다음 요청 파라미터가 지원됩니다.
  - SnapshotId
  - Size
  - VolumeType
  - TagSpecification.N
- [CreateLaunchTemplate](#) - 다음 요청 파라미터가 지원됩니다.
  - ImageId
  - InstanceType
  - SecurityGroupIds
  - TagSpecifications
  - UserData
- [CreateLaunchTemplateVersion](#)
- [CreateTags](#) - 다음 요청 파라미터가 지원됩니다.
  - AMI
  - Instance

- [Launch template](#)
- [Security group](#)
- [CreateSecurityGroup](#) - Snowball Edge에 보안 그룹을 생성합니다. 최대 50개의 보안 그룹을 생성할 수 있습니다. 보안 그룹을 생성할 때 친근한 이름을 직접 선택하여 지정합니다.
- [DeleteLaunchTemplate](#)
- [DeleteLaunchTemplateVersions](#)
- [DeleteSecurityGroup](#) - 보안 그룹을 삭제합니다. 인스턴스와 연결된 보안 그룹이나 다른 보안 그룹에서 참조하는 보안 그룹의 삭제를 시도하는 경우 `DependencyViolation`으로 작업이 실패합니다.
- [DeleteTags](#) - 지정된 리소스 세트에서 지정된 태그 세트를 삭제합니다.
- [DeleteVolume](#) - 다음 요청 파라미터가 지원됩니다.
  - `VolumeId`
- [DescribeAddresses](#)
- [DescribeImages](#)
- [DescribeInstanceAttribute](#) - 다음 속성이 지원됩니다.
  - `instanceType`
  - `userData`
- [DescribeInstanceStatus](#)
- [DescribeLaunchTemplates](#)
- [DescribeLaunchTemplateVersions](#)
- [DescribeInstances](#)
- [DescribeSecurityGroups](#) - 하나 이상의 보안 그룹을 설명합니다. `DescribeSecurityGroups`는 페이지 매김 작업입니다. 다중 API 직접 호출을 발행하여 결과의 전체 데이터 세트를 검색할 수 있습니다.
- [DescribeTags](#) - 이 명령과 함께 다음 필터가 지원됩니다.
  - `resource-id`
  - `resource-type` - AMI 또는 컴퓨팅 인스턴스만 해당합니다.
  - `key`
  - `value`
- [DescribeVolume](#) - 다음 요청 파라미터가 지원됩니다.
  - `MaxResults`
  - `NextToken`

- VolumeId.N
- [DetachVolume](#) - 다음 요청 파라미터가 지원됩니다.
  - Device
  - InstanceId
  - VolumeId
- [DisassociateAddress](#)
- [GetLaunchTemplateData](#)
- [ModifyLaunchTemplate](#)
- [ModifyInstanceAttribute](#) - userData 속성만이 지원됩니다.
- [RevokeSecurityGroupEgress](#) - 하나 이상의 송신 규칙을 보안 그룹에서 제거합니다.
- [RevokeSecurityGroupIngress](#) - 보안 그룹에 대한 하나 이상의 수신 규칙을 해지합니다.  
RevokeSecurityGroupIngress 호출 시 group-name 또는 group-id에 대한 값을 지정해야 합니다.
- [RunInstances](#) -

#### Note

컴퓨팅 인스턴스의 크기 및 유형에 따라 Snowball Edge에서 컴퓨팅 인스턴스를 시작하는 데 최대 한 시간 반이 소요될 수 있습니다.

- [StartInstances](#)
- [StopInstances](#) - 중지된 인스턴스와 연결된 리소스가 지속됩니다. 이러한 리소스를 확보하기 위해 인스턴스를 종료할 수 있습니다. 하지만 연결된 모든 데이터가 삭제됩니다.
- [TerminateInstances](#)

## Snowball Edge에서 시작 템플릿을 사용하여 EC2-compatible 인스턴스 자동 시작

시작 템플릿과 Snowball Edge 클라이언트 시작 구성 명령을 사용하여 AWS Snowball Edge 디바이스에서 Amazon EC2-compatible 인스턴스를 자동으로 시작할 수 있습니다.

시작 템플릿에는 Snowball Edge에 Amazon EC2 호환 인스턴스를 생성하는 데 필요한 구성 정보가 포함되어 있습니다. 시작 템플릿을 사용하여 시작 파라미터를 저장할 수 있으므로 Snowball Edge에서 EC2 호환 인스턴스를 시작할 때마다 시작 파라미터를 지정할 필요가 없습니다.



Snowball Edge에서 자동 시작 구성을 사용하면 Amazon EC2 호환 인스턴스를 시작할 파라미터를 구성할 수 있습니다. Snowball Edge가 구성되면 재부팅하고 잠금을 해제할 때 자동 시작 구성을 사용하여 지정한 파라미터로 인스턴스를 시작합니다. 자동 시작 구성을 사용하여 시작한 인스턴스가 중지되면 디바이스를 잠금 해제할 때 인스턴스가 실행을 시작합니다.

#### Note

자동 시작 구성을 처음 구성한 후에는 디바이스를 다시 시작하십시오. 디바이스가 잠금 해제된 후에는 모든 후속 인스턴스 시작(계획된 재부팅 또는 계획되지 않은 재부팅 이후)이 자동으로 이루어집니다.

시작 템플릿은 해당 인스턴스를 시작할 때 Amazon EC2 호환 인스턴스에 대한 Amazon Machine Image(AMI) ID, 인스턴스 유형, 사용자 데이터, 보안 그룹 및 태그를 지정할 수 있습니다. 지원되는 인스턴스 유형의 목록은 [Snowball Edge 디바이스에서 컴퓨팅 인스턴스 사용 시 할당량](#) 섹션을 참조하세요.

Snowball Edge에서 EC2 호환 인스턴스를 자동으로 시작하려면 다음 단계를 수행합니다.

1. AWS Snowball Edge 디바이스를 주문할 때 컴퓨팅 인스턴스가 있는 Snowball Edge 디바이스를 주문하는 작업을 생성합니다. 자세한 내용은 [Snowball Edge 주문 작업 생성을 참조하세요](#).
2. Snowball Edge를 받은 후 잠금을 해제합니다.
3. EC2 호환 API 명령 `aws ec2 create-launch-template`을 사용하여 시작 템플릿을 생성합니다.
4. Snowball Edge 클라이언트 명령 `snowballEdge create-autostart-configuration`을 사용하여 EC2 호환 인스턴스 시작 템플릿을 네트워크 구성에 바인딩합니다. 자세한 내용은 [Snowball Edge에서 EC2-compatible 시작 구성 생성](#) 단원을 참조하십시오.
5. 재부팅한 다음 디바이스를 잠금 해제합니다. EC2 호환 인스턴스는 시작 템플릿과 Snowball Edge 클라이언트 명령 `create-autostart-configuration`에 지정된 속성을 사용하여 자동으로 시작됩니다.

실행 중인 인스턴스 상태를 확인하려면 EC2 호환 API 명령 `describe-autostart-configurations`를 사용합니다.

**Note**

시작 템플릿을 AWS Snowball Edge 지원하기 위한 콘솔 또는 작업 관리 API는 없습니다. EC2 호환 및 Snowball Edge 클라이언트 CLI 명령을 사용하여 AWS Snowball Edge 디바이스에서 EC2 호환 인스턴스를 자동으로 시작합니다.

## Snowball Edge에서 Amazon EC2-compatible 인스턴스와 함께 Snow용 인스턴스 메타데이터 서비스 사용

Snow용 IMDS는 Snow의 Amazon EC2 호환 인스턴스에 대한 인스턴스 메타데이터 서비스(IMDS)를 제공합니다. 인스턴스 메타데이터는 인스턴스에 대한 정보 카테고리입니다. 여기에는 호스트 이름, 이벤트, 보안 그룹 등의 범주가 포함됩니다. Snow용 IMDS를 사용하면 인스턴스 메타데이터로 Amazon EC2 호환 인스턴스를 시작할 때 지정한 사용자 데이터에 액세스할 수 있습니다. 예를 들어, Snow용 IMDS를 사용하여 인스턴스 구성을 위한 파라미터를 지정하거나 단순 스크립트에 이러한 파라미터를 포함시킬 수 있습니다. 일반 AMI를 빌드하고 사용자 데이터를 사용하여 시작 시간에 제공되는 구성 파일을 수정할 수 있습니다.

인스턴스 메타데이터와 사용자 데이터 및 Snow EC2 호환 인스턴스에 대해 알아보려면 이 설명서의 [지원되는 인스턴스 메타데이터 및 사용자 데이터](#)를 참조하십시오.

**Important**

사용자는 인스턴스 자체 내에서 인스턴스 메타데이터 및 사용자 데이터에만 액세스할 수 있지만, 데이터는 인증 또는 암호화 방법으로 보호되지 않습니다. 인스턴스에 직접 액세스할 수 있는 모든 사람과 인스턴스에서 실행 중인 모든 소프트웨어는 메타데이터를 볼 수 있습니다. 따라서 암호 또는 수명이 긴 암호화 키와 같은 민감한 데이터를 사용자 데이터로 저장해서는 안 됩니다.

**Note**

이 섹션의 예에서는 인스턴스 메타데이터 서비스의 IPv4 주소(169.254.169.254)를 사용합니다. 링크-로컬 IPv6 주소를 사용한 인스턴스 메타데이터 검색은 지원되지 않습니다.

### 주제

- [Snowball Edge의 IMDS 버전](#)
- [Snowball Edge에서 IMDSv1 및 IMDSv2를 사용하여 인스턴스 메타데이터를 검색하는 예제](#)

## Snowball Edge의 IMDS 버전

IMDS 버전 2 또는 IMDS 버전 1을 사용하여 실행 중인 인스턴스에서 인스턴스 메타데이터에 액세스할 수 있습니다.

- 인스턴스 메타데이터 서비스 버전 2(IMDSv2) – 세션 지향 방법
- 인스턴스 메타데이터 서비스 버전 1(IMDSv1) – 요청/응답 방법

Snow 소프트웨어 버전에 따라 IMDSv1 또는 IMDSv2를 사용하거나 둘 다 사용할 수 있습니다. 이는 EC2 호환 인스턴스에서 실행되는 AMI 유형에 따라 달라집니다. Ubuntu 20.04를 실행하는 AMI와 같은 일부 AMI에는 IMDSv2가 필요합니다. 인스턴스 메타데이터 서비스는 PUT 또는 GET 헤더의 존재 여부에 따라 IMDSv1 요청과 IMDSv2 요청을 구별합니다. IMDSv2는 이 두 헤더를 모두 사용합니다. IMDSv1은 GET 헤더만 사용합니다.

AWS IMDSv2에는 더 높은 보안이 포함되어 있으므로 IMDSv1 대신 IMDSv2를 사용하도록 권장합니다. 자세한 내용은 [EC2 인스턴스 메타데이터 서비스의 향상된 기능을 통해 개방형 방화벽, 역방향 프록시 및 SSRF 취약성에 대한 심층적인 방어 기능 추가](#)를 참조하십시오.

## Snowball Edge의 IMDSv2

IMDSv2를 사용하여 인스턴스 메타데이터를 요청하는 경우 요청에는 다음 규칙이 포함되어야 합니다.

1. PUT 요청을 사용하여 인스턴스 메타데이터 서비스의 세션을 초기화합니다. PUT 요청은 후속 GET 요청에 포함되어야 하는 세션 토큰을 인스턴스 메타데이터 서비스에 반환합니다. 세션 기간을 정의하는 세션 토큰입니다. 세션 기간은 최소 1초, 최대 6시간입니다. 지정된 기간 중에는 후속 요청에 동일한 세션 토큰을 사용할 수 있습니다. 지정된 기간이 만료된 후에는 향후 요청에 사용할 새로운 세션 토큰을 생성해야 합니다. 토큰은 IMDSv2를 사용하여 메타데이터에 액세스하는 데 필요합니다.
2. 인스턴스 메타데이터 서비스에 대한 모든 GET 요청에 토큰을 포함시킵니다.
  - a. 토큰은 인스턴스에 특정한 키프입니다. 토큰은 다른 인스턴스에서 유효하지 않으며 해당 토큰이 생성된 인스턴스 외부에서 사용하려고 시도하면 거부됩니다.
  - b. PUT 요청에는 토큰의 TTL(Time to Live)를 최대 6시간(21,600초)까지 초 단위로 지정하는 헤더가 포함되어야 합니다. 토큰은 논리 세션을 나타냅니다. TTL은 토큰이 유효한 시간 길이를 지정하며 따라서 세션 기간을 지정합니다.

- c. 토큰이 만료된 후 인스턴스 메타데이터에 계속 액세스하려면 다른 PUT 요청을 사용하여 새 세션을 생성해야 합니다.
- d. 토큰을 재사용하거나 모든 요청에 새 토큰을 생성하도록 선택할 수 있습니다. 요청 수가 적은 경우 인스턴스 메타데이터 서비스에 액세스해야 할 때마다 토큰을 생성하고 즉시 사용하는 것이 더 간편할 수 있습니다. 하지만 효율성을 향상하려면 인스턴스 메타데이터를 요청해야 할 때마다 PUT 요청을 작성하는 대신 토큰에 더 긴 기간을 지정하고 토큰을 재사용할 수 있습니다. 동시 토큰 수에는 실질적인 제한이 없으며 각각은 자체 세션을 나타냅니다.

IMDSv2 인스턴스 메타데이터 요청에서는 HTTP GET 및 HEAD 메서드가 허용됩니다. PUT 요청은 X-Forwarded-For 헤더가 포함된 경우 거부됩니다.

기본적으로 PUT 요청에 대한 응답에는 IP 프로토콜 수준에서 1의 응답 홉 제한(TTL(Time to Live))이 있습니다. Snow용 IMDS는 PUT 응답에 대한 홉 제한을 수정할 수 없습니다.

다음 예에서는 Linux 셸 스크립트와 IMDSv2를 사용하여 최상위 인스턴스 메타데이터 항목을 가져옵니다. 이 예에서는

1. PUT 요청을 사용하여 6시간(21,600초) 동안 지속되는 세션 토큰을 생성합니다.
2. TOKEN이라는 변수에 세션 토큰 헤더를 저장합니다.
3. 토큰을 사용하여 최상위 메타데이터 항목을 요청합니다.

두 개의 명령을 사용하여 EC2-compatible 토큰을 생성합니다. 명령을 별도로 실행하거나 하나의 명령으로 실행할 수 있습니다.

먼저 다음 명령을 사용하여 토큰을 생성합니다.

#### Note

X-aws-ec2-metadata-token-ttl-seconds는 필수 헤더입니다. 이 헤더가 포함되지 않은 경우 400 - 파라미터 누락 또는 잘못된 파라미터 오류 코드를 받게 됩니다.

```
[ec2-user ~]$ TOKEN=$(curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600")
```

그런 다음 해당 토큰을 사용하여 다음 명령으로 최상위 메타데이터 항목을 생성합니다.

```
[ec2-user ~]$ curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/meta-data/
```

### Note

토큰을 만드는 데 오류가 발생하면 유효한 토큰 대신에 오류 메시지가 변수에 저장되고 명령이 작동하지 않습니다.

토큰을 저장하고 명령을 결합할 수 있습니다. 다음 예는 위의 두 명령을 결합하고 TOKEN이라는 변수에 세션 토큰 헤더를 저장합니다.

### Example결합된 명령

```
[ec2-user ~]$ TOKEN=$(curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600" \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/meta-data/
```

토큰을 생성한 후에는 만료될 때까지 토큰을 재사용할 수 있습니다. 다음 예제 명령에서는 인스턴스를 시작하는 데 사용한 AMI의 ID를 가져오고 이전 예에서 생성한 \$TOKEN에 이를 저장합니다.

### Example토큰 재사용의 경우

```
[ec2-user ~]$ curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/meta-data/ami-id
```

## Snowball Edge의 IMDSv1

IMDSv1은 요청-응답 모델을 사용합니다. 인스턴스 메타데이터를 요청하려면 인스턴스 메타데이터 서비스에 GET 요청을 전송합니다.

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/
```

인스턴스 메타데이터는 실행 중인 인스턴스에서 사용할 수 있으므로 Amazon EC2 콘솔 또는 AWS CLI 를 사용하여 액세스할 필요가 없습니다. 이는 인스턴스에서 실행할 스크립트를 작성할 때 유용합니다. 예를 들어, 사용자는 인스턴스 메타데이터에서 인스턴스의 로컬 IP 주소에 액세스하여 외부 애플리케이션과의 연결을 관리할 수 있습니다. 인스턴스 메타데이터는 몇 가지 범주로 분류될 수 있습니다. 각 인스턴스 메타데이터 범주에 대한 설명은 이 설명서의 [지원되는 인스턴스 메타데이터 및 사용자 데이터](#)를 참조하십시오.

실행 중인 모든 인스턴스 메타데이터 범주를 살펴보려면 다음 IPv4 URI를 사용하십시오.

```
http://169.254.169.254/latest/meta-data/
```

IP 주소는 링크-로컬 주소이며 인스턴스에서만 유효합니다. 자세한 내용은 Wikipedia의 [Link-local address](#)를 참조하십시오.

모든 인스턴스 메타데이터는 텍스트(HTTP 콘텐츠 유형 text/plain)로 반환됩니다.

특정 메타데이터 리소스를 요청하면 적절한 값이 반환되거나 리소스를 이용할 수 없는 경우 404 - 찾을 수 없음 HTTP 오류 코드가 반환됩니다.

일반 메타데이터 리소스(/로 끝나는 URI)를 요청한 경우 이용 가능한 리소스 목록이 반환되거나 해당 리소스가 없는 경우 404 - 찾을 수 없음 HTTP 오류 코드가 반환됩니다. 목록 항목은 개별 라인에 표시되고 줄바꿈(ASCII 문자 코드 10)으로 끝납니다.

IMDSv1을 사용하여 수행한 요청의 경우 다음 HTTP 오류 코드가 반환될 수 있습니다.

- 400 - 파라미터 누락 또는 유효하지 않음 - PUT 요청이 유효하지 않습니다.
- 401 - 권한이 없음 - GET 요청이 유효하지 않은 토큰을 사용합니다. 권장되는 작업은 새 토큰을 생성하는 것입니다.
- 403 - 금지됨 - 요청이 허용되지 않거나 인스턴스 메타데이터 서비스가 꺼져 있습니다.

## Snowball Edge에서 IMDSv1 및 IMDSv2를 사용하여 인스턴스 메타데이터를 검색하는 예제

다음 예에서는 Linux 인스턴스에서 사용할 수 있는 명령을 제공합니다.

## Example 인스턴스 메타데이터의 사용 가능한 버전 가져오기

이 예를 통해 이용 가능한 인스턴스 메타데이터 버전을 가져올 수 있습니다. 각 버전은 새 인스턴스 메타데이터 카테고리가 릴리스될 때 인스턴스 메타데이터 빌드를 참조합니다. 이전 버전의 구조 및 정보를 사용하는 스크립트인 경우 이전 버전을 사용할 수 있습니다.

### IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H
"X-aws-ec2-metadata-token-ttl-seconds: 21600"` && curl -H "X-aws-ec2-metadata-token:
$TOKEN" -v http://169.254.169.254/
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
Dload  Upload  Total    Spent    Left  Speed
 100    56    100    56      0      0    0    3733    0   --:--:--
--:--:-- --:--:-- 3733
*   Trying 169.254.169.254...
* TCP_NODELAY set
* Connected to 169.254.169.254 (169.254.169.254) port 80 (#0)
> GET / HTTP/1.1
> Host: 169.254.169.254
> User-Agent: curl/7.61.1
> Accept: */*
> X-aws-ec2-metadata-token:
MDAXcxNFLbAwJIYx8KzgNckcHTdxT4Tt69TzpKExlXKTULHIQnjEtXvD
>
* HTTP 1.0, assume close after body
< HTTP/1.0 200 OK
< Date: Mon, 12 Sep 2022 21:58:03 GMT
< Content-Length: 274
< Content-Type: text/plain
< Server: EC2ws
<
1.0
2007-01-19
2007-03-01
2007-08-29
2007-10-10
2007-12-15
2008-02-01
2008-09-01
2009-04-04
2011-01-01
```

```
2011-05-01
2012-01-12
2014-02-25
2014-11-05
2015-10-20
2016-04-19
2016-06-30
2016-09-02
2018-03-28
2018-08-17
2018-09-24
2019-10-01
2020-10-27
2021-01-03
2021-03-23
* Closing connection 0
```

## IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/
1.0
2007-01-19
2007-03-01
2007-08-29
2007-10-10
2007-12-15
2008-02-01
2008-09-01
2009-04-04
2011-01-01
2011-05-01
2012-01-12
2014-02-25
2014-11-05
2015-10-20
2016-04-19
2016-06-30
2016-09-02
2018-03-28
2018-08-17
2018-09-24
```



```

2019-10-01
2020-10-27
2021-01-03
2021-03-23
latest

```

## Example 최고 수준 메타데이터 항목 가져오기

이 예는 최고 수준 메타데이터 항목을 가져옵니다. 최상위 메타데이터 항목에 대한 자세한 내용은 이 설명서의 [지원되는 인스턴스 메타데이터 및 사용자](#) 데이터를 참조하십시오.

## IMDSv2

```

[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H
"X-aws-ec2-metadata-token-ttl-seconds: 21600"` && curl -H "X-aws-ec2-metadata-token:
$TOKEN" -v http://169.254.169.254/latest/meta-data/
ami-id
hostname
instance-id
instance-type
local-hostname
local-ipv4
mac
network/
reservation-id
security-groups

```

## IMDSv1

```

[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/
ami-id
hostname
instance-id
instance-type
local-hostname
local-ipv4
mac
network/

```

```
reservation-id
security-groups
```

## Example최상위 메타데이터의 값 가져오기

다음 예에서는 이전 예에서 얻은 몇 가지 최상위 메타데이터 항목의 값을 가져옵니다. IMDSv2 요청에서는 토큰이 만료되지 않았다고 가정하고 이전 예제 명령에서 생성한 저장된 토큰을 사용합니다.

### ami-id IMDSv2

```
curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/meta-data/ami-id ami-0abcdef1234567890
```

### ami-id IMDSv1

```
curl http://169.254.169.254/latest/meta-data/ami-id ami-0abcdef1234567890
```

### reservation-id IMDSv2

```
[ec2-user ~]$ curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/meta-data/reservation-id r-0efghijk987654321
```

### reservation-id IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/reservation-id \
r-0efghijk987654321
```

### local-hostname IMDSv2

```
[ec2-user ~]$ curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/meta-data/local-hostname ip-00-000-00-00
```

## local-hostname IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/local-hostname ip-00-000-00-00
```

## Snowball Edge에서 Amazon EC2-compatible 인스턴스와 함께 블록 스토리지 사용

Snowball Edge의 블록 스토리지는 애플리케이션 요구 사항에 따라 블록 스토리지를 추가하거나 제거할 수 있도록 합니다. Amazon EC2 호환 인스턴스에 연결된 볼륨은 스토리지 볼륨으로 표시되며, 인스턴스 수명과 관계없이 지속됩니다. 익숙한 Amazon EBS API를 사용해 블록 스토리지를 관리할 수 있습니다.

특정 EC2 호환 엔드포인트를 사용하면 일부 Amazon EBS 명령어가 지원됩니다. 지원되는 명령어에는 `attach-volume`, `create-volume`, `delete-volume`, `detach-volume`, `describe-volumes`가 포함됩니다. 이러한 명령에 대한 자세한 내용은 [Snowball Edge에서 지원되는 EC2-compatible AWS CLI 명령 목록](#) 섹션을 참조하세요.

### Important

볼륨을 분리하기 전에 운영 체제 안에서 디바이스의 파일 시스템을 모두 탑재 해제합니다. 해제하지 않으면 잠재적으로 데이터가 손실될 수 있습니다.

다음으로 Amazon EBS 볼륨 할당량 및 디바이스의 Amazon EBS 볼륨과 클라우드의 Amazon EBS 볼륨 차이를 설명합니다.

- Amazon EBS 볼륨은 볼륨을 호스팅하는 디바이스에서 실행되는 EC2 호환 인스턴스에서만 사용할 수 있습니다.
- 볼륨 유형은 용량 최적화 HDD(`sbp1`) 또는 성능 최적화 SSD(`sbp1`)로 제한됩니다. 기본 볼륨 유형은 `sbp1`입니다.

- Snowball Edge에서는 Amazon S3 객체와 Amazon EBS 간에 HDD 메모리를 공유합니다. 에서 HDD 기반 블록 스토리지를 사용하는 경우 Amazon S3 객체에 사용할 수 있는 메모리 양 AWS Snowball Edge이 줄어듭니다. 마찬가지로 Amazon S3 객체는 HDD 볼륨에서 Amazon EBS 볼륨 스토리지에 사용할 수 있는 메모리 양을 줄입니다.
- Amazon EC2 호환 루트 볼륨은 항상 IDE 드라이버를 사용합니다. 추가 Amazon EBS 볼륨은 가능한 경우 Virtio 드라이버를 우선 사용합니다. Virtio 드라이버를 이용할 수 없는 경우 SBE는 IDE 드라이버를 기본으로 사용합니다. 성능이 우수하기 때문에 Virtio 드라이버의 사용을 추천합니다.
- Amazon EBS 볼륨을 생성할 때 encrypted 파라미터는 지원되지 않습니다. 그러나 디바이스의 모든 데이터는 기본적으로 암호화됩니다.
- 볼륨 크기는 1GB에서 10TB까지 가능합니다.
- 하나의 EC2 호환 인스턴스에 최대 10개의 Amazon EBS 볼륨을 연결할 수 있습니다.
- AWS Snowball Edge 디바이스에 추가할 수 있는 Amazon EBS 볼륨 수에 대한 공식적인 제한은 없습니다. 그러나 디바이스의 사용 가능한 공간에 따라 총 Amazon EBS 볼륨 크기가 제한될 수 있습니다.

## Snowball Edge의 보안 그룹을 사용하여 네트워크 트래픽 제어

보안 그룹은 하나 이상의 인스턴스에 대한 트래픽을 제어하는 가상 방화벽 역할을 합니다. 인스턴스를 시작할 때 하나 이상의 보안 그룹을 인스턴스와 연결합니다. 연결된 인스턴스와 트래픽을 주고받을 수 있게 하는 규칙을 각 보안 그룹에 추가할 수 있습니다. 자세한 내용은 Amazon EC2 사용 설명서의 [Linux 인스턴스용 Amazon EC2 보안 그룹](#)을 참조하세요.

Snowball Edge 디바이스의 보안 그룹은 AWS 클라우드의 보안 그룹과 비슷합니다. Virtual Private Cloud(VPC)는 Snowball Edge 디바이스에서 지원되지 않습니다.

다음에서 Snowball Edge 보안 그룹과 EC2-VPC 보안 그룹의 기타 차이를 확인할 수 있습니다.

- 각 Snowball Edge의 보안 그룹 수는 50개로 제한됩니다.
- 기본 보안 그룹은 인바운드와 아웃바운드 트래픽을 모두 허용합니다.
- 로컬 인스턴스 간의 트래픽은 프라이빗 인스턴스 IP 주소 또는 퍼블릭 IP 주소를 사용할 수 있습니다. 예를 들어 SSH를 사용하여 인스턴스 A에서 인스턴스 B로 연결하려고 합니다. 이 경우 보안 그룹이 이 트래픽을 허용하면 대상 IP 주소는 인스턴스 B의 퍼블릭 IP 또는 프라이빗 IP 주소일 수 있습니다.
- AWS CLI 작업 및 API 호출에 대해 나열된 파라미터만 지원됩니다. 일반적으로 이는 EC2-VPC 인스턴스에서 지원되는 항목의 하위 집합입니다.

지원되는 AWS CLI 작업에 대한 자세한 내용은 [섹션을 참조하세요](#) [Snowball Edge에서 지원되는 EC2-compatible AWS CLI 명령 목록](#). 지원되는 API 작업에 대한 자세한 내용은 [Snowball Edge에서 지원되는 Amazon EC2-compatible API 작업](#) 섹션을 참조하세요.

## Snowball Edge에서 지원되는 EC2-compatible 인스턴스 메타데이터 및 사용자 데이터

인스턴스 메타데이터는 실행 중인 인스턴스를 구성 또는 관리하는 데 사용될 수 있는 인스턴스 관련 데이터입니다. Snowball Edge에서는 컴퓨팅 인스턴스에 대해 인스턴스 메타데이터 범주의 하위 집합을 지원합니다. 자세한 내용은 Amazon EC2 사용 설명서의 [인스턴스 메타데이터 및 사용자 데이터](#)를 참조하세요.

다음과 같은 범주가 지원됩니다. 다른 카테고리를 사용하여 404 오류 메시지를 반환합니다.

Snowball Edge 디바이스에서 지원되는 인스턴스 메타데이터 범주

Data	설명
ami-id	인스턴스를 시작하기 위해 사용된 AMI ID.
hostname	인스턴스의 프라이빗 IPv4 DNS 호스트 이름.
instance-id	이 인스턴스의 ID.
instance-type	인스턴스 유형.
local-hostname	인스턴스의 프라이빗 IPv4 DNS 호스트 이름.
local-ipv4	인스턴스의 프라이빗 IPv4 주소.
mac	인스턴스의 미디어 액세스 제어(MAC) 주소.
network/interfaces/macs/ local-hostname	인터페이스의 로컬 호스트이름.
network/interfaces/macs/ local-ipv4s	프라이빗 IPv4 주소는 인터페이스와 연결됩니다.
network/interfaces/macs/ mac	인스턴스의 MAC 주소.

Data	설명
network/interfaces/macs/ <i>mac</i> / public-ipv4s	탄력적 IP 주소는 인터페이스와 연결됩니다.
public-ipv4	퍼블릭 IPv4 주소.
public-keys/0/openssh-key	퍼블릭 키. 시작 시에 인스턴스가 제공된 경우에만 사용할 수 있습니다.
reservation-id	예약 ID:
userData	시작 시 인스턴스에 명령을 전송하기 위한 Shell 스크립트입니다.

## Snowball Edge 디바이스에서 지원되는 인스턴스 동적 데이터 범주

Data	설명
instance-identity/document	인스턴스 속성이 포함된 JSON입니다. instanceId , imageId, privateIp 및 instanceType 에만 값이 있고 반환된 기타 속성은 null입니다. 자세한 내용은 Amazon EC2 사용 설명서의 <a href="#">인스턴스 자격 증명 문서</a> 섹션을 참조하세요.

## Snowball Edge의 컴퓨터 인스턴스 사용자 데이터

셸 스크립트를 사용하여 Snowball Edge 디바이스의 컴퓨팅 인스턴스 사용자 데이터에 액세스합니다. Shell 스크립트를 사용하면 시작 시 인스턴스에 명령을 전송할 수 있습니다. modify-instance-attribute AWS CLI 명령 또는 ModifyInstanceAttribute API 작업을 사용하여 사용자 데이터를 변경할 수 있습니다.

### 사용자 데이터 변경하기

1. stop-instances AWS CLI 명령을 사용하여 컴퓨팅 인스턴스를 중지합니다.
2. modify-instance-attribute AWS CLI 명령을 사용하여 userData 속성을 수정합니다.

3. `start-instances` AWS CLI 명령을 사용하여 컴퓨팅 인스턴스를 다시 시작합니다.

컴퓨팅 인스턴스에서는 Shell 스크립트만 지원됩니다. Snowball Edge 디바이스에서 실행 중인 컴퓨팅 인스턴스에서는 `cloud-init` 패키지 명령이 지원되지 않습니다. AWS CLI 명령 작업에 대한 자세한 내용은 [AWS CLI 명령](#) 참조를 참조하세요.

## Snowball Edge에서 실행 중인 EC2-compatible 인스턴스 중지

디바이스에서 작성한 Amazon EC2-compatible 인스턴스를 실수로 삭제하지 않도록 운영 체제에서 인스턴스를 종료하면 안 됩니다. 예를 들어, `shutdown` 또는 `reboot` 명령을 사용하지 마세요. 운영 체제에서 인스턴스를 종료하면 [terminate-instances](#) 명령을 호출하는 것과 효과가 동일합니다.

대신 [stop-instances](#) 명령을 사용하여 보존하려는 Amazon EC2 호환 인스턴스를 일시 중단합니다.

# AWS IoT Greengrass 를 사용하여 Snowball Edge의 Amazon EC2-compatible 인스턴스에서 사전 설치된 소프트웨어 실행

AWS IoT Greengrass 는 디바이스에서 IoT 애플리케이션을 빌드, 배포 및 관리하는 데 도움이 되는 오픈 소스 사물 인터넷(IoT) 엣지 런타임 및 클라우드 서비스입니다. AWS IoT Greengrass 를 사용하여 디바이스가 생성하는 데이터에 대해 로컬로 작동하고, 기계 학습 모델을 기반으로 예측을 실행하고, 디바이스 데이터를 필터링 및 집계할 수 있는 소프트웨어를 구축할 수 있습니다. 에 대한 자세한 내용은 AWS IoT Greengrass Version 2 개발자 안내서의 [란 무엇입니까 AWS IoT Greengrass?](#)를 AWS IoT Greengrass참조하세요.

Snowball Edge 디바이스 AWS IoT Greengrass 에서를 사용하면 디바이스가 데이터가 생성된 위치와 더 가까운 위치에서 데이터를 수집 및 분석하고, 로컬 이벤트에 자율적으로 대응하고, 로컬 네트워크의 다른 디바이스와 안전하게 통신할 수 있습니다.

## Snowball Edge AWS IoT Greengrass 에서에 대한 Amazon EC2-compatible 인스턴스 설정

### Note

Snowball Edge 디바이스 AWS IoT Greengrass Version 2 에를 설치하려면 디바이스가 인터넷에 연결되어 있는지 확인합니다. 설치 후에는 Snowball Edge 디바이스가 작동하는 데 인터넷이 필요하지 않습니다 AWS IoT Greengrass.

에 대한 EC2-compatible 인스턴스를 설정하려면 AWS IoT Greengrass V2

1. 퍼블릭 IP 주소와 SSH 키를 사용하여 AWS IoT Greengrass 검증된 AMI를 시작합니다.
  - a. 사용 AWS CLI: [run-instances](#).
  - b. 사용 AWS OpsHub: [Amazon EC2-compatible 인스턴스 시작](#).

### Note

인스턴스와 연결된 퍼블릭 IP 주소 및 SSH 키 이름을 기록해 둡니다.



- SSH를 이용해 EC2 호환 인스턴스에 연결합니다. 이를 위해 디바이스에 연결된 컴퓨터에서 다음 명령을 실행합니다. `ssh-key`를 EC2 호환 인스턴스를 시작하는 데 사용한 키로 대체합니다. `public-ip-address`를 EC2 호환 인스턴스의 퍼블릭 IP 주소로 대체합니다.

```
ssh -i ssh-key ec2-user@ public-ip-address
```

### ⚠ Important

컴퓨터에서 이전 버전의 Microsoft Windows를 사용하는 경우 SSH 명령이 없거나 SSH가 있지만 EC2 호환 인스턴스에 연결하지 못할 수 있습니다. EC2 호환 인스턴스에 연결하려면 무료 오픈 소스 SSH 클라이언트인 PuTTY를 설치하고 구성하면 됩니다. SSH 키를 .pem 형식에서 PuTTY 형식으로 변환하고 EC2 인스턴스에 연결해야 합니다. .pem에서 PuTTY 형식으로 변환하는 방법은 Amazon EC2 사용 설명서의 [PuTTYgen을 사용하여 프라이빗 키 변환](#)을 참조하십시오.

## Snowball Edge AWS IoT Greengrass 의 EC2-compatible 인스턴스에 설치

다음으로 EC2-compatible 인스턴스를 로컬 개발에 사용할 수 있는 AWS IoT Greengrass 코어 디바이스로 설정합니다.

를 설치하려면 AWS IoT Greengrass

- 다음 명령을 사용하여 사전 조건 소프트웨어를 설치합니다 AWS IoT Greengrass. 이 명령은 AWS Command Line Interface (AWS CLI) v2, Python 3 및 Java 8을 설치합니다.

```
curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
&& unzip awscliv2.zip && sudo ./aws/install && sudo yum -y install python3
java-1.8.0-openjdk
```

- 루트 사용자에게 AWS IoT Greengrass 소프트웨어를 실행하고 sudoers 구성 파일의 루트 권한을 에서 root ALL=(ALL) ALL로 수정할 수 root ALL=(ALL:ALL) ALL 있는 권한을 부여합니다.

```
sudo sed -in 's/root\tALL=(ALL)/root\tALL=(ALL:ALL)/' /etc/sudoers
```

- 다음 명령을 사용하여 AWS IoT Greengrass 코어 소프트웨어를 다운로드합니다.

```
curl -s https://d2s8p88vqu9w66.cloudfront.net/releases/greengrass-nucleus-latest.zip > greengrass-nucleus-latest.zip && unzip greengrass-nucleus-latest.zip -d GreengrassCore && rm greengrass-nucleus-latest.zip
```

4. 다음 명령을 사용하여 AWS IoT Greengrass 코어 소프트웨어를 설치할 수 있는 자격 증명을 제공합니다. 예시 값을 사용자의 자격 증명으로 바꿉니다.

```
export AWS_ACCESS_KEY_ID=AKIAIOSFODNN7EXAMPLE
export AWS_SECRET_ACCESS_KEY=wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY
```

#### Note

이는 Snowball Edge 디바이스가 아닌 AWS 리전의 IAM 사용자의 자격 증명입니다.

5. 다음 명령을 사용하여 AWS IoT Greengrass 코어 소프트웨어를 설치합니다. 이 명령은 코어 소프트웨어가 작동하는 데 필요한 AWS 리소스를 생성하고 AMI가 부팅될 때 실행되는 시스템 서비스로 코어 소프트웨어를 설정합니다.

명령에서 다음 파라미터를 교체합니다.

- **region**: 리소스를 찾거나 생성할 AWS 리전입니다.
- **MyGreengrassCore**: AWS IoT Greengrass 코어 디바이스의 AWS IoT 사물 이름입니다.
- **MyGreengrassCoreGroup**: AWS IoT Greengrass 코어 디바이스에 대한 AWS IoT 사물 그룹의 이름입니다.

```
sudo -E java -Droot="/greengrass/v2" -Dlog.store=FILE \
  -jar ./GreengrassInstaller/lib/Greengrass.jar \
  --aws-region region \
  --thing-name MyGreengrassCore \
  --thing-group-name MyGreengrassCoreGroup \
  --thing-policy-name GreengrassV2IoTThingPolicy \
  --tes-role-name GreengrassV2TokenExchangeRole \
  --tes-role-alias-name GreengrassCoreTokenExchangeRoleAlias \
  --component-default-user ggc_user:ggc_group \
  --provision true \
```

```
--setup-system-service true \  
--deploy-dev-tools true
```

**Note**

이 명령은 Amazon Linux 2 AMI를 실행하는 Amazon EC2 호환 인스턴스를 위한 명령입니다. Windows AMI의 경우 [AWS IoT Greengrass 코어 소프트웨어 설치를 참조하세요](#).

완료되면 로컬 사용을 위해 Snowball Edge 디바이스에서 AWS IoT Greengrass 코어가 실행됩니다.

## 와 AWS Lambda 함께 사용 AWS Snowball Edge

AWS Lambda 로 구동되는 AWS IoT Greengrass 는 Snowball Edge 디바이스에서 로컬로 서버리스 코드(Lambda 함수)를 실행할 수 있는 컴퓨팅 서비스입니다. Lambda를 사용하여 메시지 대기열 원격 측정 전송(MQTT) 메시지와 함께 Snowball Edge 디바이스에서 Lambda 함수를 호출하고, Lambda 함수에서 Python 코드를 실행하고, 이를 사용하여 클라우드에서 퍼블릭 AWS 서비스 엔드포인트를 호출할 수 있습니다. Snowball Edge 디바이스에서 Lambda 함수를 사용하려면에서 AWS 리전 지원하는에서 Snowball Edge 작업을 생성해야 합니다 AWS IoT Greengrass. 유효한 목록은의 섹션을 AWS 리전 참조 [AWS IoT Greengrass](#) 하세요 AWS 일반 참조. Snowball Edge의 Lambda는 Lambda 및 Snowball Edge 디바이스를 사용할 수 있는 리전에서 사용 가능합니다.

### Note

각 함수에 128MB의 최소 권장 메모리를 할당한 경우 단일 작업에서 최대 7개의 Lambda 함수를 실행할 수 있습니다.

### 주제

- [Snowball Edge에서 Lambda 사용 시작하기](#)
- [Lambda 함수를 Snowball Edge 디바이스에 배포하기](#)

## Snowball Edge에서 Lambda 사용 시작하기

Snowball Edge에서 실행할 Python 언어 Lambda 함수를 생성하기 전에 다음과 같은 서비스, 개념, 관련 주제를 익히는 것이 좋습니다.

### Snowball Edge AWS IoT Greengrass 의에 대한 사전 조건

AWS IoT Greengrass 는 AWS 클라우드 기능을 로컬 디바이스로 확장하는 소프트웨어입니다. 를 AWS IoT Greengrass 사용하면 로컬 디바이스가 정보 소스에 더 가까운 데이터를 수집하고 분석하는 동시에 로컬 네트워크에서 서로 안전하게 통신할 수 있습니다. 보다 구체적으로, 를 사용하는 개발자에서 서버리스 코드(Lambda 함수)를 작성할 AWS IoT Greengrass 수 있습니다 AWS 클라우드. 그러면 애플리케이션의 로컬 실행을 위해 이 코드를 디바이스에 배포할 수 있어 편리합니다.

다음 AWS IoT Greengrass 개념은 Snowball Edge와 AWS IoT Greengrass 함께를 사용할 때 이해하는데 중요합니다.

- AWS IoT Greengrass 요구 사항 - AWS IoT Greengrass 요구 사항의 전체 목록은 AWS IoT Greengrass Version 2 개발자 안내서의 [요구 사항을](#) 참조하세요.
- AWS IoT Greengrass core - AWS IoT Greengrass 코어 소프트웨어를 다운로드하여 디바이스에서 실행되는 EC2 인스턴스에 설치합니다. 이 설명서 [AWS IoT Greengrass 의 Amazon EC2 인스턴스에서 사용을](#) 참조하세요.

Snowball Edge 디바이스에서 Lambda 함수를 사용하려면 먼저 디바이스의 Amazon EC2 인스턴스에 AWS IoT Greengrass 코어 소프트웨어를 설치해야 합니다. Snowball Edge 디바이스에서 사용하려는 Lambda 함수는 Snowball Edge 디바이스 AWS IoT Greengrass 에 설치하는 데 사용할 계정과 동일한 계정으로 생성해야 합니다. Snowball Edge 디바이스 AWS IoT Greengrass 에를 설치하는 방법에 대한 자세한 내용은 섹션을 참조하세요 [AWS IoT Greengrass 를 사용하여 Snowball Edge의 Amazon EC2-compatible 인스턴스에서 사전 설치된 소프트웨어 실행.](#)

- AWS IoT Greengrass 그룹 - Snowball Edge 디바이스는 AWS IoT Greengrass 그룹의 코어 디바이스로서 그룹의 일부입니다. 그룹에 대한 자세한 내용은 AWS IoT Greengrass 개발자 설명서의 [AWS Greengrass IoT 그룹을](#) 참조하세요.
- MQTT - 업계 표준의 경량 MQTT 프로토콜을 AWS IoT Greengrass 사용하여 그룹 내에서 통신합니다. AWS IoT Greengrass 그룹의 MQTT와 호환되는 모든 디바이스 또는 소프트웨어는 MQTT 메시지를 호출할 수 있습니다. 관련 MQTT 메시지를 호출하도록 정의하면 이러한 메시지는 Lambda 함수를 호출할 수 있습니다.

## Snowball Edge AWS Lambda 의에 대한 사전 조건

AWS Lambda 는 서버를 프로비저닝하거나 관리하지 않고도 코드를 실행할 수 있는 컴퓨팅 서비스입니다. Snowball Edge에서 Lambda를 사용하려면 다음 Lambda 개념을 이해하는 것이 중요합니다.

- Lambda 함수 - Lambda에 업로드되고 게시되며 Snowball Edge에서 사용되는 사용자 지정 코드입니다. 자세한 내용은 AWS Lambda 개발자 가이드의 [Lambda 함수](#)를 참조하세요.
- Lambda 콘솔 - Snowball Edge에서 사용하기 위해 Python 언어 Lambda 함수를 업로드, 업데이트, 게시할 콘솔입니다. [Lambda 콘솔](#)에 대한 자세한 내용은 AWS Lambda 개발자 설명서에서 [Lambda란 무엇입니까?](#)를 참조하세요.
- Python - Snowball Edge에서 로 구동되는 Lambda 함수 AWS IoT Greengrass 에 사용되는 상위 수준 프로그래밍 언어입니다.는 Python 버전 3.8.x를 AWS IoT Greengrass 지원합니다.

# Lambda 함수를 Snowball Edge 디바이스에 배포하기

AWS IoT Greengrass 그룹의 Snowball Edge 디바이스에서 Lambda 함수를 실행하려면 함수를 구성 요소로 가져옵니다. AWS IoT Greengrass 콘솔을 사용하여 함수를 구성 요소로 가져오는 방법에 대한 자세한 내용은 AWS IoT Greengrass Version 2 개발자 안내서의 [구성 요소로 Lambda 함수 가져오기 \(콘솔\)](#)를 참조하세요.

1. AWS IoT 콘솔의 Greengrass 구성 요소 페이지에서 구성 요소 생성을 선택합니다.
2. 구성 요소 소스에서 Lambda 함수 가져오기를 선택합니다. Lambda 함수에서 함수의 이름을 선택합니다. Lambda 함수 버전에서 함수의 버전을 선택합니다.
3. 함수를 실행할 수 있는 메시지에 해당 함수를 구독하려면 이벤트 소스 추가를 선택하고 이벤트를 선택하세요. 타임아웃(초)에 타임아웃 기간을 초 단위로 입력합니다.
4. 고정에서 함수를 고정할지 여부를 선택합니다.
5. 구성 요소 생성을 선택합니다.
6. 배포(Deploy)를 선택합니다.
7. 배포에서 기존 배포에 추가를 선택한 다음 Greengrass 그룹을 선택합니다. 다음을 선택합니다.
8. 퍼블릭 구성 요소에서 다음 구성 요소를 선택합니다.
  - aws.greengrass.Cli
  - aws.greengrass.LambdaLauncher
  - aws.greengrass.LambdaManager
  - aws.greengrass.LambdaRuntimes
  - aws.greengrass.Nucleus
9. 배포(Deploy)를 선택합니다.

## Snowball Edge에서 Amazon S3 호환 스토리지 사용

Snowball Edge의 Amazon S3 호환 스토리지는 향상된 복원력, 확장 및 견고한 모바일 엣지 및 연결 해제된 환경으로 확장된 Amazon S3 API 기능 세트를 통해 안전한 객체 스토리지를 제공합니다. Snowball Edge에서 Amazon S3 호환 스토리지를 사용하면 데이터를 저장하고 엣지 컴퓨팅을 위해 Snowball Edge에서 고가용성 애플리케이션을 실행할 수 있습니다.

Snowball Edge 디바이스에 Amazon S3 버킷을 생성하여, 로컬 데이터 액세스, 로컬 데이터 처리, 데이터 레지던시가 필요한 애플리케이션에서 온프레미스로 객체를 쉽게 저장하고 검색할 수 있습니다. Snowball Edge의 Amazon S3 호환 스토리지는 Amazon S3 APIs를 사용하는 새로운 스토리지 클래스 SNOW인를 제공하며 여러 Snowball Edge 디바이스에 데이터를 내구성 있고 중복적으로 저장하도록 설계되었습니다. 버킷 수명 주기 정책, 암호화, 태그 지정을 포함하여 Amazon S3에서와 같이 Snowball Edge 버킷에서 동일한 API 및 기능을 사용할 수 있습니다. 디바이스 또는 디바이스가 반환되면 Snowball Edge의 Amazon S3 호환 스토리지에서 생성되거나 저장된 AWS 모든 데이터가 삭제됩니다. 자세한 내용은 [로컬 컴퓨팅 및 스토리지 전용 작업](#) 섹션을 참조하세요.

독립형 구성 또는 클러스터 구성으로 Snowball Edge에 Amazon S3 호환 스토리지를 배포할 수 있습니다. 독립형 구성에서는 사용 가능한 S3 용량을 디바이스에 프로비저닝할 수 있으며 남은 용량을 블록 스토리지로 사용할 수 있습니다. 클러스터 구성에서는 모든 데이터 디스크 용량이 S3 스토리지에 사용됩니다. 클러스터는 최소 3개의 디바이스에서 최대 16개의 디바이스로 구성될 수 있습니다. 클러스터 크기에 따라 S3 서비스는 1~2개 디바이스의 디바이스 내결합성을 유지하도록 설계되었습니다.

를 사용하면 Snowball Edge의 Amazon S3 compatible storage on Snowball Edge 디바이스 및 AWS 스토리지 서비스 간에 객체를 전송할 AWS DataSync 수 있습니다. 자세한 내용은 AWS DataSync 사용 설명서의 [Snowball Edge에서 S3 호환 스토리지로 전송 구성](#)을 참조하세요.

다음은 Snowball Edge의 Amazon S3 호환 스토리지와 Snowball Edge의 Amazon S3 호환 스토리지를 사용하는 독립 실행형 디바이스의 블록 스토리지 용량입니다. 클러스터의 내결합성 및 스토리지 용량에 대한 내용은 [this table](#) 섹션을 참조하세요.

### Snowball Edge Compute Optimized with NVMe storage

Snowball Edge의 Amazon S3 호환 스토리지 스토리지 및 Snowball Edge 컴퓨팅 최적화(AMD EPYC Gen2 및 NVMe로 컴퓨팅 최적화) 디바이스의 블록 스토리지

Amazon S3 compatible storage on Snowball Edge 스토리지 용량(TB)	블록 스토리지 용량(TB)
3	17.5

Amazon S3 compatible storage on Snowball Edge 스토리지 용량(TB)	블록 스토리지 용량(TB)
5.5	14.5
10.5	8.5
12	6.5
13	5.5
16.5	1.5

### Snowball Edge storage optimized 210 TB

Snowball Edge의 Amazon S3 호환 스토리지 스토리지 용량 및 Snowball Edge 스토리지 최적화 210TB 디바이스의 블록 스토리지

Amazon S3 compatible storage on Snowball Edge 스토리지 용량(TB)	블록 스토리지 용량(TB)
20	206
40	182
60	158
80	134
100	110
120	86
140	62
160	38
180	14
190	2



## Amazon S3 compatible storage on Snowball Edge 사양:

- Snowball Edge 버킷의 최대 수는 디바이스당 또는 클러스터당 100개입니다.
- S3 on Snowball Edge 버킷 소유자 계정은 버킷의 모든 객체를 소유합니다.
- S3 on Snowball Edge 버킷 소유자 계정만 버킷에서 작업을 수행할 수 있습니다.
- 객체 크기 제한은 Amazon S3의 제한과 일치합니다.
- S3 on Snowball Edge에 저장된 모든 객체에는 스토리지 클래스로 SNOW가 있습니다.
- 기본적으로 SNOW 스토리지 클래스에 저장된 모든 객체는 Amazon S3 관리형 암호화 키(SSE-S3)와 함께 서버 측 암호화를 사용하여 저장됩니다. 또한 고객 제공 암호화 키(SSE-C)와 함께 서버 측 암호화를 사용하여 객체를 저장하도록 명시적으로 선택할 수도 있습니다.
- Snowball Edge에 객체를 저장할 공간이 충분하지 않은 경우 API는 용량 부족 예외(ICE)를 반환합니다.

## 주제

- [Snowball Edge에서 Amazon S3 호환 스토리지 주문](#)
- [Snowball Edge에서 Amazon S3 호환 스토리지 설정 및 시작](#)
- [Snowball Edge에서 Amazon S3 호환 스토리지로 S3 버킷 작업](#)
- [Snowball Edge의 Amazon S3 compatible storage on Snowball Edge 버킷에 액세스할 수 있는지 여부 확인](#)
- [Snowball Edge의 Snowball Edge에 있는 Amazon S3 호환 스토리지에서 버킷 또는 리전 버킷 목록 검색](#)
- [Snowball Edge의 Snowball Edge에서 Amazon S3 호환 스토리지로 버킷 가져오기](#)
- [Snowball Edge의 Snowball Edge에서 Amazon S3 호환 스토리지에 S3 버킷 생성](#)
- [Snowball Edge의 Snowball Edge에 있는 Amazon S3 호환 스토리지에서 버킷 삭제](#)
- [를 사용하여 객체 수명 주기 구성 생성 및 관리 AWS CLI](#)
- [Snowball Edge의 Amazon S3 compatible storage on Snowball Edge 버킷에 객체 복사](#)
- [Snowball Edge의 Snowball Edge에서 Amazon S3 호환 스토리지의 버킷에 객체 나열](#)
- [Snowball Edge의 Snowball Edge에 있는 Amazon S3 호환 스토리지의 버킷에서 객체 가져오기](#)
- [Snowball Edge의 Amazon S3 호환 스토리지에 있는 버킷의 객체 삭제](#)
- [Snowball Edge의 Amazon S3 호환 스토리지에 지원되는 REST API 작업](#)
- [Snow 디바이스 클러스터와 함께 Snowball Edge에서 Amazon S3 호환 스토리지 사용](#)
- [Snowball Edge에서 Amazon S3 호환 스토리지 이벤트 알림 구성](#)

- [Snowball Edge에서 로컬 SMTP 알림 구성](#)

## Snowball Edge에서 Amazon S3 호환 스토리지 주문

Snowball Edge에서 Amazon S3 호환 스토리지용 디바이스 주문은 Snowball Edge 주문 프로세스와 매우 유사합니다. 주문하려면 이 설명서의 [Snowball Edge 디바이스 주문 작업 생성](#) 섹션을 참조하고 주문 과정에서 다음 항목을 염두에 두세요.

- 작업 유형 선택에서 로컬 컴퓨팅 및 스토리지 전용을 선택합니다.
- Snow 디바이스에서 Snowball Edge 컴퓨팅 최적화를 선택합니다.
- 스토리지 유형 선택에서 Snowball Edge의 Amazon S3 호환 스토리지를 선택합니다.
- 독립형 디바이스의 경우 스토리지 용량에서 단일 디바이스를 선택한 다음 원하는 스토리지 용량을 선택합니다.
- 클러스터의 경우 스토리지 용량에서 클러스터를 선택한 다음 원하는 스토리지 용량과 내결함성을 선택합니다.

## Snowball Edge에서 Amazon S3 호환 스토리지 설정 및 시작

에서 로컬 환경 AWS 으로 소프트웨어 도구를 설치하고 구성하여 Snowball Edge 디바이스 또는 디바이스 클러스터 및 Snowball Edge의 Amazon S3 호환 스토리지와 상호 작용합니다. 그런 다음 이러한 도구를 사용하여 Snowball Edge 디바이스 또는 클러스터를 설정하고 Snowball Edge에서 Amazon S3 호환 스토리지를 시작합니다.

### 사전 조건

Snowball Edge의 Amazon S3 호환 스토리지를 사용하려면 Snowball Edge 클라이언트와를 로컬 환경에 AWS CLI 설치해야 합니다. 또한 SDK for .NET 및 AWS Tools for Windows PowerShell을 사용하여 다음 버전의 도구를 사용하여 Snowball Edge. AWS recommends에서 Amazon S3 호환 스토리지로 작업할 수 있습니다.

- Snowball Edge 클라이언트 - 최신 버전을 사용합니다. 자세한 내용은 이 설명서의 [Snowball Edge 클라이언트 다운로드 및 설치](#)를 참조하십시오.
- AWS CLI – 버전 2.11.15 이상. 자세한 내용은 AWS Command Line Interface 사용 설명서의 [설치, 업데이트 및 제거 AWS CLI](#)를 참조하세요.
- SDK for .NET – AWSSDK.S3Control 3.7.304.8 이상. 자세한 내용은 [AWS SDK for .NET](#) 단원을 참조하십시오.

- AWS Windows PowerShell용 도구 - 버전 4.1.476 이상. 자세한 내용은 [AWS Tools for PowerShell 사용 설명서](#)를 참조하세요.

## 로컬 환경 설정

이 섹션에서는 Snowball Edge에서 Amazon S3 호환 스토리지와 함께 사용할 수 있도록 Snowball Edge 클라이언트 및 로컬 환경을 설정하고 구성하는 방법을 설명합니다.

1. Snowball Edge 클라이언트를 다운로드 및 설치합니다. 자세한 내용은 [Snowball Edge 클라이언트 다운로드 및 설치](#)를 참조하십시오.
2. Snowball Edge 클라이언트의 프로필을 구성합니다. 자세한 내용은 [Snowball Edge 클라이언트의 프로필 구성](#)을 참조하십시오.
3. 를 사용하는 경우 `clientConfig.AuthenticationRegion` 파라미터 값을 다음과 같이 SDK for .NET 설정합니다.

```
clientConfig.AuthenticationRegion = "snow"
```

## Snowball Edge 디바이스 설정

### Snowball Edge에서 IAM 설정

AWS Identity and Access Management (IAM)를 사용하면 Snowball Edge 디바이스에서 실행되는 AWS 리소스에 대한 세분화된 액세스를 활성화할 수 있습니다. IAM을 사용하여 리소스를 사용하도록 인증(로그인) 및 권한 부여(권한 있음)된 대상을 제어합니다.

IAM은 Snowball Edge에서 로컬로 지원됩니다. 로컬 IAM 서비스를 사용하여 역할을 생성하고 IAM 정책을 연결할 수 있습니다. 이러한 정책을 사용하여 할당된 작업을 수행하는 데 필요한 액세스를 허용할 수 있습니다.

다음 예시는 Amazon S3 API에 대한 모든 액세스를 허용합니다.

### JSON

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

```

```
{
  "Sid": "VisualEditor0",
  "Effect": "Allow",
  "Action": "s3:*",
  "Resource": "*"
}
```

IAM 정책 예시에 대한 자세한 내용은 [AWS Snowball Edge Developer Guide](#) 섹션을 참조하세요.

## Snowball Edge 서비스에서 Amazon S3 호환 스토리지 시작

다음 지침에 따라 Snowball Edge 디바이스 또는 클러스터에서 Snowball Edge의 Amazon S3 호환 스토리지 서비스를 시작합니다.

보다 사용자 친화적인 환경을 원하는 경우를 사용하여 독립 실행형 디바이스 또는 디바이스 클러스터에 대해 Amazon S3 compatible storage on Snowball Edge 서비스를 시작할 수 있습니다 AWS OpsHub. [를 사용하여 Snowball Edge에서 Amazon S3 호환 스토리지 설정 AWS OpsHub](#)을(를) 참조하세요.

1. 다음 명령을 실행하여 Snowball Edge 디바이스 또는 디바이스 클러스터를 잠금 해제합니다.

- 단일 디바이스의 경우:

```
snowballEdge unlock-device --endpoint https://snow-device-ip
```

- 클러스터의 경우:

```
snowballEdge unlock-cluster
```

2. 다음 명령을 실행하고 Snowball Edge 디바이스 또는 디바이스 클러스터가 잠금 해제되었는지 확인합니다.

- 단일 디바이스의 경우:

```
snowballEdge describe-device --endpoint https://snow-device-ip
```

- 클러스터의 경우:

```
snowballEdge describe-cluster --device-ip-addresses [snow-device-1-ip] [snow-
device-2-ip] /
[snow-device-3-ip] [snow-device-4-ip] [snow-device-5-ip] /
[snow-device-6-ip]
```

3. 각 디바이스(하나 또는 클러스터)에 대해 Snowball Edge에서 Amazon S3 호환 스토리지를 시작하려면 다음을 수행합니다.

- a. 다음 describe-device 명령을 실행하여 디바이스의 PhysicalNetworkInterfaceId를 가져옵니다.

```
snowballEdge describe-device --endpoint https://snow-device-ip
```

- b. 다음 create-virtual-network-interface 명령을 두 번 실행하여 (버킷 작업 시) s3control 및 (객체 작업 시) s3api 엔드포인트를 위한 가상 네트워크 인터페이스(VNI)를 생성합니다.

```
snowballEdge create-virtual-network-interface --ip-address-assignment
dhcp --manifest-file manifest --physical-network-interface-id
"PhysicalNetworkInterfaceId" --unlock-code unlockcode --endpoint https://snow-
device-ip
```

이 명령은 IP 주소가 포함된 JSON 구조를 반환합니다. 이 IP 주소를 기록해 둡니다.

이러한 명령에 대한 자세한 내용은 [Snowball Edge에서 가상 네트워크 인터페이스\(VNI\) 설정을 참조하세요.](#)

#### Note

Snowball Edge에서 Amazon S3 호환 스토리지를 시작하면 디바이스 리소스가 사용 됩니다.

4. 다음 start-service 명령을 실행하여 Snowball Edge에서 Amazon S3 호환 스토리지를 시작합니다. 여기에는 디바이스의 IP 주소와 s3control 및 s3api 엔드포인트에 대해 생성한 VNIs의 Amazon 리소스 이름(ARNs)이 포함됩니다.

단일 디바이스에서 서비스를 시작하는 방법은 다음과 같습니다.

```
snowballEdge start-service --service-id s3-snow --device-ip-addresses snow-device-1-ip --virtual-network-interface-arns vni-arn-1 vni-arn-2
```

클러스터에서 서비스를 시작하려면:

```
snowballEdge start-service --service-id s3-snow --device-ip-addresses snow-device-1-ip snow-device-2-ip snow-device-3-ip --virtual-network-interface-arns vni-arn-1 vni-arn-2 vni-arn-3 vni-arn-4 vni-arn-5 vni-arn-6
```

--virtual-network-interface-arns의 경우 이전 단계에서 생성한 모든 VNI에 대한 ARN을 포함합니다. 공백을 사용하여 각 ARN을 구분합니다.

5. 단일 디바이스의 경우 다음 describe-service 명령을 실행합니다.

```
snowballEdge describe-service --service-id s3-snow
```

서비스 상태가 Active가 될 때까지 기다립니다.

클러스터의 경우 다음 describe-service 명령을 실행합니다.

```
snowballEdge describe-service --service-id s3-snow \  
--device-ip-addresses snow-device-1-ip snow-device-2-ip snow-device-3-ip
```

## Snowball Edge 엔드포인트의 Amazon S3 호환 스토리지에 대한 정보 보기

Amazon S3 compatible storage on Snowball Edge service가 실행 중인 경우 describe-service Snowball Edge Client 명령을 사용하여 s3control 및 s3api 엔드포인트와 연결된 IP 주소를 볼 수 있습니다.

```
snowballEdge describe-service --service-id s3-snow --endpoint https://snow-device-ip-address --profile profile-name
```

## Example `describe-service` 명령의 출력

이 예제에서 s3control 엔드포인트의 IP 주소는 192.168.1.222이고 s3api 엔드포인트의 IP 주소는 192.168.1.152입니다.

```
{
  "ServiceId": "s3-snow",
  "Autostart": true,
  "Status": {
    "State": "ACTIVATING",
    "Details": "Attaching storage"
  },
  "ServiceCapacities": [
    {
      "Name": "S3 Storage",
      "Unit": "Byte",
      "Used": 148599705600,
      "Available": 19351400294400
    }
  ],
  "Endpoints": [
    {
      "Protocol": "https",
      "Port": 443,
      "Host": "192.168.1.222",
      "CertificateAssociation": {
        "CertificateArn": "arn:aws:snowball-
device:::certificate/30c563f1124707705117f57f6c3accd42a4528ed6dba1e35c1822a391a717199d8c49973d3",
      },
      "Description": "s3-snow bucket API endpoint (for s3control SDK)",
      "DeviceId": "JID-beta-207429000001-23-12-28-03-51-11",
      "Status": {
        "State": "ACTIVE"
      }
    },
    {
      "Protocol": "https",
      "Port": 443,
      "Host": "192.168.1.152",
      "CertificateAssociation": {
        "CertificateArn": "arn:aws:snowball-
device:::certificate/30c563f1124707705117f57f6c3accd42a4528ed6dba1e35c1822a391a717199d8c49973d3",
      },
    }
  ]
}
```

```

    "Description": "s3-snow object & bucket API endpoint (for s3api SDK)",
    "DeviceId": "JID-beta-207429000001-23-12-28-03-51-11",
    "Status": {
      "State": "ACTIVATING"
    }
  }
]
}

```

## Snowball Edge에서 Amazon S3 호환 스토리지로 S3 버킷 작업

Snowball Edge의 Amazon S3 호환 스토리지를 사용하면 Snowball Edge 디바이스에 Amazon S3 버킷을 생성하여 로컬 데이터 액세스, 로컬 데이터 처리 및 데이터 레지던시가 필요한 애플리케이션을 위해 온프레미스에 객체를 저장하고 검색할 수 있습니다. Snowball Edge의 Amazon S3 호환 스토리지는 Amazon S3 APIs를 사용하는 새로운 스토리지 클래스 SNOW인를 제공하며 여러 Snowball Edge 디바이스에 데이터를 내구성 있고 중복적으로 저장하도록 설계되었습니다. 버킷 수명 주기 정책, 암호화, 태그 지정을 포함하여 Amazon S3에서와 같이 Snowball Edge 버킷에서 동일한 API 및 기능을 사용할 수 있습니다.

AWS Command Line Interface (AWS CLI)를 사용하거나 AWS Java SDK를 통해 프로그래밍 방식으로 Snowball Edge에서 Amazon S3 호환 스토리지를 사용할 수 있습니다. 를 사용하면 s3api 또는 s3control 엔드포인트를 설정하고 명령을 통해 상호 작용 AWS CLI할 수 있습니다. 버킷 및 객체 작업에 동일한 엔드포인트를 사용할 수 있으므로 s3api 엔드포인트를 사용하는 것이 좋습니다.

### Note

s3api 엔드포인트는 Snowball Edge 소프트웨어의 버전 8004 이상에서 사용할 수 있습니다. 디바이스에 설치된 Snowball Edge 소프트웨어 버전을 찾으려면 snowballEdge check-for-updates 명령을 사용합니다. Snowball Edge 디바이스를 업데이트하려면 [Snowball Edge 디바이스의 소프트웨어 업데이트](#)를 참조하세요.

## 사용 AWS CLI

다음 지침에 따라 AWS CLI를 사용하여 디바이스에서 Amazon S3 버킷을 사용합니다.

를 설정하려면 AWS CLI

1. ~/.aws/config에서 객체 엔드포인트의 프로필을 생성합니다.



```
[profile your-profile]
aws_access_key_id = your-access-id
aws_secret_access_key = your-access-key
region = snow
ca_bundle = dev/apps/ca-certs/your-ca_bundle
```

2. 디바이스에서 인증서를 받습니다. 자세한 내용은 [Snowball Edge Developer Guide](#) 섹션을 참조하세요.
3. 가상 환경에 SDK를 설치한 경우 다음 명령을 사용하여 활성화합니다.

```
source your-virtual-environment-name/bin/activate
```

작업을 설정한 후 s3api SDK 또는 s3control SDK를 사용하여 Snowball Edge의 S3 버킷에 액세스할 수 있습니다 AWS CLI.

Example s3api SDK를 사용하여 S3 버킷에 액세스

```
aws s3api list-buckets --endpoint-url https://s3api-endpoint-ip --profile your-profile
```

Example s3control SDK를 사용하여 S3 버킷에 액세스하는 방법

```
aws s3control list-regional-buckets --account-id bucket-owner --endpoint-url
https://s3ctrlapi-endpoint-ip --profile your-profile
```

Example s3api SDK를 사용하여 S3 객체에 액세스

```
aws s3api list-objects-v2 --endpoint-url https://s3api-endpoint-ip --profile your-profile
```

## Java SDK 사용

다음 예시를 사용하면 Java SDK를 통해 Amazon S3 버킷 및 객체에 액세스할 수 있습니다.

```

import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.auth.credentials.AwsBasicCredentials;
import software.amazon.awssdk.auth.credentials.StaticCredentialsProvider;
import software.amazon.awssdk.http.SdkHttpClient;
import software.amazon.awssdk.http.apache.ApacheHttpClient;
import software.amazon.awssdk.regions.Region;

import java.net.URI;

AwsBasicCredentials creds = AwsBasicCredentials.create(accessKey, secretKey); // set
creds by getting Access Key and Secret Key from snowball edge
SdkHttpClient httpClient =
    ApacheHttpClient.builder().tlsTrustManagersProvider(trustManagersProvider).build(); //
    set trust managers provider with client certificate from snowball edge
String s3SnowEndpoint = "10.0.0.0"; // set s3-snow object api endpoint from describe
service

S3Client s3Client =
    S3Client.builder().httpClient(httpClient).region(Region.of("snow")).endpointOverride(new
    URI(s3SnowEndpoint)).credentialsProvider(StaticCredentialsProvider.create(creds)).build();

```

## 버킷 ARN 형식

여기에 나열된 Amazon 리소스 이름(ARN) 형식을 사용하여 Snowball Edge 디바이스의 Amazon S3 버킷을 식별할 수 있습니다.

```
arn:partition:s3:snow:account-id:device/device-id/bucket/bucket-name
```

여기서 *###*은 Snowball Edge 디바이스를 주문한 리전의 파티션입니다. *device-id*는 디바이스가 독립형 Snowball Edge 디바이스인 경우 *job\_id*이고, Snowball Edge 클러스터가 있는 경우 *cluster\_id*입니다.

## 버킷 위치 형식

버킷 위치 형식은 버킷이 생성될 Snowball Edge 디바이스를 지정합니다. 버킷 위치의 형식은 다음과 같습니다.

```
/device-id/bucket/bucket-name
```

자세한 내용은 AWS CLI 명령 참조의 [create-bucket](#)을 참조하세요.

## Snowball Edge의 Amazon S3 compatible storage on Snowball Edge 버킷에 액세스할 수 있는지 여부 확인

다음 예시에서는 head-bucket 명령을 사용하여 버킷이 존재하고 AWS CLI를 사용하여 버킷에 액세스할 수 있는 권한이 있는지를 확인합니다. 이 명령을 사용하려면 각 사용자 입력 자리 표시자를 사용자의 정보로 대체합니다.

```
aws s3api head-bucket --bucket sample-bucket --endpoint-url https://s3api-endpoint-ip
--profile your-profile
```

## Snowball Edge의 Snowball Edge에 있는 Amazon S3 호환 스토리지에서 버킷 또는 리전 버킷 목록 검색

list-regional-buckets 또는를 사용하여 list-buckets를 사용하는 Snowball Edge 버킷의 Amazon S3 호환 스토리지를 나열합니다 AWS CLI.

Example를 사용하여 버킷 또는 리전 버킷 목록 검색 AWS CLI

s3api syntax

```
aws s3api list-buckets --endpoint-url https://s3api-endpoint-ip --profile your-profile
```

list-buckets 명령에 대한 자세한 내용은 AWS CLI 명령 참조의 [list-buckets](#)를 참조하세요.

s3control syntax

```
aws s3control list-regional-buckets --account-id 123456789012 --endpoint-url
https://s3ctrlapi-endpoint-ip --profile your-profiles
```

list-regional-buckets 명령에 대한 자세한 내용은 AWS CLI 명령 참조의 [list-regional-buckets](#)을 참조하세요.

다음 SDK for Java 예시에서는 Snowball Edge 디바이스의 버킷 목록을 가져옵니다. 자세한 내용은 Amazon Simple Storage Service API 참조의 [ListBuckets](#)을 참조하세요.

```
import com.amazonaws.services.s3.model.*;
public void listBuckets() {
    ListBucketsRequest reqListBuckets = new ListBucketsRequest()
        .withAccountId(AccountId)
    ListBucketsResult respListBuckets = s3APIClient.RegionalBuckets(reqListBuckets);
    System.out.printf("ListBuckets Response: %s%n", respListBuckets.toString());
}
```

다음 PowerShell 예시에서는 Snowball Edge 디바이스의 버킷 목록을 가져옵니다.

```
Get-S3CRegionalBucketList -AccountId 012345678910 -Endpoint "https://snowball_ip" -
Region snow
```

다음 .NET 예시에서는 Snowball Edge 디바이스의 버킷 목록을 가져옵니다.

```
using Amazon.S3Control;
using Amazon.S3Control.Model;

namespace SnowTest;

internal class Program
{
    static async Task Main(string[] args)
    {
        var config = new AmazonS3ControlConfig
        {
            ServiceURL = "https://snowball_ip",
            AuthenticationRegion = "snow" // Note that this is not RegionEndpoint
        };

        var client = new AmazonS3ControlClient(config);

        var response = await client.ListRegionalBucketsAsync(new
        ListRegionalBucketsRequest()
        {
            AccountId = "012345678910"
        });
    }
}
```

}

## Snowball Edge의 Snowball Edge에서 Amazon S3 호환 스토리지로 버킷 가져오기

다음 예시에서는 `aws s3control`를 사용하여 Snowball Edge 버킷의 Amazon S3 호환 스토리지를 가져옵니다. AWS CLI. 이 명령을 사용하려면 각 사용자 입력 자리 표시자를 사용자의 정보로 대체합니다.

```
aws s3control get-bucket --account-id 123456789012 --bucket amzn-s3-demo-bucket --
endpoint-url https://s3ctrlapi-endpoint-ip --profile your-profile
```

이 명령에 대한 자세한 내용은 AWS CLI 명령 참조의 [get-bucket](#)을 참조하세요.

다음 Amazon S3 compatible storage on Snowball Edge 예제에서는 Java용 SDK를 사용하여 버킷을 가져옵니다. 자세한 내용은 [Amazon Simple Storage Service API 참조의 GetBucket](#)을 참조하세요.

```
import com.amazonaws.services.s3control.model.*;

public void getBucket(String bucketName) {

    GetBucketRequest reqGetBucket = new GetBucketRequest()
        .withBucket(bucketName)
        .withAccountId(AccountId);

    GetBucketResult respGetBucket = s3ControlClient.getBucket(reqGetBucket);
    System.out.printf("GetBucket Response: %s\n", respGetBucket.toString());
}
```

## Snowball Edge의 Snowball Edge에서 Amazon S3 호환 스토리지에 S3 버킷 생성

Snowball Edge 디바이스에 Amazon S3 버킷을 생성하여, 로컬 데이터 액세스, 로컬 데이터 처리, 데이터 레지던시가 필요한 애플리케이션의 엣지에서 객체를 저장하고 검색할 수 있습니다. Snowball Edge의 Amazon S3 호환 스토리지는 Amazon S3를 SNOW사용하고 여러 디바이스에 데이터를 내구성 있고

중복적으로 저장하도록 설계된 새로운 스토리지 클래스인을 제공합니다. 버킷 수명 주기 정책, 암호화, 태그 지정을 포함하여 Amazon S3 버킷에서와 같이 동일한 API 및 기능을 사용할 수 있습니다.

다음 예시에서는 AWS CLI를 사용하여 Snowball Edge 디바이스용 Amazon S3 버킷을 생성합니다. 이 명령을 실행하려면 사용자 입력 자리 표시자를 사용자의 정보로 대체합니다.

### Example S3 버킷 생성

#### s3api syntax

```
aws s3api create-bucket --bucket your-snow-bucket --endpoint-url https://s3api-endpoint-ip --profile your-profile
```

#### s3control syntax

```
aws s3control create-bucket --bucket your-snow-bucket --endpoint-url https://s3ctrlapi-endpoint-ip --profile your-profile
```

## Snowball Edge의 Snowball Edge에 있는 Amazon S3 호환 스토리지에서 버킷 삭제

s3api SDK 또는 s3control SDK를 사용하여 Snowball Edge의 Amazon S3 호환 스토리지에서 버킷을 삭제할 수 있습니다.

### Important

- 버킷을 AWS 계정 생성하는 버킷을 소유하며 버킷을 삭제할 수 있는 유일한 버킷입니다.
- Snowball Edge 버킷은 비어 있어야 삭제할 수 있습니다.
- 삭제한 버킷은 복구할 수 없습니다.

다음 예제에서는를 사용하여 Amazon S3 compatible storage on Snowball Edge 버킷을 삭제합니다 AWS CLI. 이 명령을 사용하려면 각 사용자 입력 자리 표시자를 사용자의 정보로 대체합니다.

## Example버킷 삭제

### s3api syntax

```
aws s3api delete-bucket --bucket amzn-s3-demo-bucket --endpoint-url https://s3api-endpoint-ip --profile your-profile
```

이 명령에 대한 자세한 내용은 AWS CLI 명령 참조의 [delete-bucket](#)을 참조하세요.

### s3control syntax

```
aws s3control delete-bucket --account-id 123456789012 --bucket amzn-s3-demo-bucket --endpoint-url https://s3ctrlapi-endpoint-ip --profile your-profile
```

이 명령에 대한 자세한 내용은 AWS CLI 명령 참조의 [delete-bucket](#)을 참조하세요.

## 를 사용하여 객체 수명 주기 구성 생성 및 관리 AWS CLI

Amazon S3 수명 주기를 사용하여 Snowball Edge의 Amazon S3 호환 스토리지에 대한 스토리지 용량을 최적화할 수 있습니다. 객체가 오래되거나 더 최신 버전으로 교체되면 객체를 만료시키도록 수명 주기 규칙을 만들 수 있습니다. 수명 주기 규칙을 생성, 사용, 사용 중지 또는 삭제할 수 있습니다. 자세한 내용은 [스토리지 수명 주기 관리](#) 섹션을 참조하세요.

### Note

버킷을 AWS 계정 생성하는 버킷을 소유하며 수명 주기 규칙을 생성, 활성화, 비활성화 또는 삭제할 수 있는 유일한 버킷입니다.

AWS Command Line Interface (AWS CLI)를 사용하여 Amazon S3 compatible storage on Snowball Edge 버킷의 수명 주기 구성을 생성하고 관리하려면 다음 예제를 참조하세요.

## Snowball Edge 버킷에 수명 주기 구성 추가

다음 AWS CLI 예제에서는 Snowball Edge 버킷에 수명 주기 구성 정책을 적용합니다. 이 정책은 플래그가 지정된 접두사(*myprefix*)와 10일 후에 만료되는 태그가 포함된 모든 객체를 지정합니다. 이 예시를 사용하려면 각각의 사용자 입력 자리 표시자를 사용자의 정보로 바꿉니다.

먼저 수명 주기 구성 정책을 JSON 파일로 저장합니다. 이 예시에서 파일의 이름은 **lifecycle-example.json**으로 지정됩니다.

```
{
  "Rules": [{
    "ID": "id-1",
    "Filter": {
      "And": {
        "Prefix": "myprefix",
        "Tags": [{
          "Value": "mytagvalue1",
          "Key": "mytagkey1"
        },
        {
          "Value": "mytagvalue2",
          "Key": "mytagkey2"
        }
      ]
    }
  },
  "Status": "Enabled",
  "Expiration": {
    "Days": 10
  }
}]
}
```

파일을 저장한 후 JSON 파일을 `put-bucket-lifecycle-configuration` 명령의 일부로 제출합니다. 이 명령을 사용하려면 각 사용자 입력 자리 표시자를 사용자의 정보로 대체합니다.

Example/ `put-bucket-lifecycle` 명령

s3api syntax

```
aws s3api put-bucket-lifecycle-configuration --bucket example-snow-bucket \\  
  --lifecycle-configuration file://lifecycle-example.json --endpoint-url  
  https://s3api-endpoint-ip --profile your-profile
```

이 명령에 대한 자세한 내용은 AWS CLI 명령 참조의 [put-bucket-lifecycle-configuration](#)을 참조하세요.



## s3control syntax

```
aws s3control put-bucket-lifecycle-configuration --bucket example-snow-bucket \\  
--lifecycle-configuration file://lifecycle-example.json \\  
--endpoint-url https://s3ctrlapi-endpoint-ip --profile your-profile
```

이 명령에 대한 자세한 내용은 AWS CLI 명령 참조의 [put-bucket-lifecycle-configuration](#)을 참조하세요.

## Snowball Edge의 Amazon S3 compatible storage on Snowball Edge 버킷에 객체 복사

다음 예시에서는 사용할 수 있는 쓰기 권한이 있는 Snowball Edge 버킷의 Amazon S3 호환 스토리지에 *sample-object.xml*이라는 파일을 업로드합니다 AWS CLI. 이 명령을 사용하려면 각 사용자 입력 자리 표시자를 사용자의 정보로 대체합니다.

```
aws s3api put-object --bucket sample-bucket --key sample-object.xml --body sample-object.xml --endpoint-url s3api-endpoint-ip --profile your-profile
```

다음 Amazon S3 compatible storage on Snowball Edge 예제에서는 Java용 SDK를 사용하여 객체를 동일한 버킷의 새 객체로 복사합니다. 이 명령을 사용하려면 각 사용자 입력 자리 표시자를 사용자의 정보로 대체합니다.

```
import com.amazonaws.AmazonServiceException;  
import com.amazonaws.SdkClientException;  
import com.amazonaws.services.s3.AmazonS3;  
import com.amazonaws.services.s3.AmazonS3ClientBuilder;  
import com.amazonaws.services.s3.model.CopyObjectRequest;  
add : import java.io.IOException;  
  
public class CopyObject {  
    public static void main(String[] args) {  
        String bucketName = "*** Bucket name ***";  
        String sourceKey = "*** Source object key ***";  
        String destinationKey = "*** Destination object key ***";  
  
        try {  
            // This code expects that you have AWS credentials set up per:
```

```

// https://docs.aws.amazon.com/sdk-for-java/v1/developer-guide/setup-
credentials.html
AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
    .enableUseArnRegion()
    .build();

// Copy the object into a new object in the same bucket.
CopyObjectRequest copyObjectRequest = new CopyObjectRequest(sourceKey,
destinationKey);
s3Client.copyObject(copyObjectRequest);
CopyObjectRequest copyObjectRequest = CopyObjectRequest.builder()
    .sourceKey(sourceKey)
    .destinationKey(destKey)
    .build();
} catch (AmazonServiceException e) {
    // The call was transmitted successfully, but Amazon S3 couldn't process
    // it, so it returned an error response.
    e.printStackTrace();
} catch (SdkClientException e) {
    // Amazon S3 couldn't be contacted for a response, or the client
    // couldn't parse the response from Amazon S3.
    e.printStackTrace();
}
}
}

```

## Snowball Edge의 Snowball Edge에서 Amazon S3 호환 스토리지의 버킷에 객체 나열

다음 예제에서는 `aws s3api`를 사용하여 Amazon S3 compatible storage on Snowball Edge 버킷의 객체를 나열합니다. AWS CLI SDK 명령은 `s3-snow:ListObjectsV2`입니다. 이 명령을 사용하려면 각각의 사용자 입력 자리 표시자를 사용자의 정보로 바꿉니다.

```
aws s3api list-objects-v2 --bucket sample-bucket --endpoint-url s3api-endpoint-ip --
profile your-profile
```

이 명령에 대한 자세한 내용은 [AWS CLI Command Reference](#)의 `list-objects-v2` 섹션을 참조하세요.

다음 Amazon S3 compatible storage on Snowball Edge 예제에서는 Java용 SDK를 사용하여 버킷의 객체를 나열합니다. 이 명령을 사용하려면 각 사용자 입력 자리 표시자를 사용자의 정보로 대체합니다.

이 예시에서는 ListObjects API 작업의 최신 버전인 [ListObjectsV2](#)를 사용합니다. 애플리케이션 개발 시 이 개정된 API 작업을 사용하는 것이 좋습니다. 이전 버전과의 호환성을 위해 Amazon S3는 이 API 작업의 이전 버전을 계속 지원합니다.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.ListObjectsV2Request;
import com.amazonaws.services.s3.model.ListObjectsV2Result;
import com.amazonaws.services.s3.model.S3ObjectSummary;

public class ListObjectsV2 {

    public static void main(String[] args) {
        String bucketName = "*** Bucket name ***";

        try {
            // This code expects that you have AWS credentials set up per:
            // https://docs.aws.amazon.com/sdk-for-java/v1/developer-guide/setup-
credentials.html
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                .enableUseArnRegion()
                .build();

            System.out.println("Listing objects");

            // maxKeys is set to 2 to demonstrate the use of
            // ListObjectsV2Result.getNextContinuationToken()
            ListObjectsV2Request req = new
ListObjectsV2Request().withBucketName(bucketName).withMaxKeys(2);
            ListObjectsV2Result result;

            do {
                result = s3Client.listObjectsV2(req);

                for (S3ObjectSummary objectSummary : result.getObjectSummaries()) {
                    System.out.printf(" - %s (size: %d)\n", objectSummary.getKey(),
objectSummary.getSize());
                }
                // If there are more than maxKeys keys in the bucket, get a
continuation token
```

```

        // and list the next objects.
        String token = result.getNextContinuationToken();
        System.out.println("Next Continuation Token: " + token);
        req.setContinuationToken(token);
    } while (result.isTruncated());
} catch (AmazonServiceException e) {
    // The call was transmitted successfully, but Amazon S3 couldn't process
    // it, so it returned an error response.
    e.printStackTrace();
} catch (SdkClientException e) {
    // Amazon S3 couldn't be contacted for a response, or the client
    // couldn't parse the response from Amazon S3.
    e.printStackTrace();
}
}
}
}

```

## Snowball Edge의 Snowball Edge에 있는 Amazon S3 호환 스토리지의 버킷에서 객체 가져오기

다음 예제에서는 `s3api`를 사용하여 Snowball Edge 버킷의 Amazon S3 호환 스토리지에서 `sample-object.xml`이라는 객체를 가져옵니다. AWS CLI. SDK 명령은 `s3-snow:GetObject`입니다. 이 명령을 사용하려면 각각의 사용자 입력 자리 표시자를 사용자의 정보로 바꿉니다.

```
aws s3api get-object --bucket sample-bucket --key sample-object.xml --endpoint-url s3api-endpoint-ip --profile your-profile
```

이 명령에 대한 자세한 내용은 [AWS CLI Command Reference](#)의 `get-object` 섹션을 참조하세요.

다음 Amazon S3 compatible storage on Snowball Edge 예제에서는 Java용 SDK를 사용하여 객체를 가져옵니다. 이 명령을 사용하려면 각 사용자 입력 자리 표시자를 사용자의 정보로 대체합니다. 자세한 내용은 [Amazon Simple Storage Service API Reference](#)의 `GetObject` 섹션을 참조하세요.

```

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.GetObjectRequest;
import com.amazonaws.services.s3.model.ResponseHeaderOverrides;

```

```
import com.amazonaws.services.s3.model.S3Object;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;

public class GetObject {
    public static void main(String[] args) throws IOException {
        String bucketName = "**** Bucket name ****";
        String key = "**** Object key ****";

        S3Object fullObject = null, objectPortion = null, headerOverrideObject = null;
        try {
            // This code expects that you have AWS credentials set up per:
            // https://docs.aws.amazon.com/sdk-for-java/v1/developer-guide/setup-
credentials.html
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                .enableUseArnRegion()
                .build();
            GetObjectRequest getObjectRequest = GetObjectRequest.builder()
                .bucket(bucketName)
                .key(key)
                .build();

s3Client.getObject(getObjectRequest);
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        } finally {
            // To ensure that the network connection doesn't remain open, close any
open input streams.
            if (fullObject != null) {
                fullObject.close();
            }
            if (objectPortion != null) {
                objectPortion.close();
            }
            if (headerOverrideObject != null) {
```

```

        headerOverrideObject.close();
    }
}

private static void displayTextInputStream(InputStream input) throws IOException {
    // Read the text input stream one line at a time and display each line.
    BufferedReader reader = new BufferedReader(new InputStreamReader(input));
    String line = null;
    while ((line = reader.readLine()) != null) {
        System.out.println(line);
    }
    System.out.println();
}
}
}

```

## Snowball Edge의 Amazon S3 호환 스토리지에 있는 버킷의 객체 삭제

Amazon S3 compatible storage on Snowball Edge 버킷에서 하나 이상의 객체를 삭제할 수 있습니다. 다음 예시에서는 AWS CLI를 사용하여 *sample-object.xml* 이름의 객체를 삭제합니다. 이 명령을 사용하려면 각각의 사용자 입력 자리 표시자를 사용자의 정보로 바꿉니다.

```
aws s3api delete-object --bucket sample-bucket --key key --endpoint-url s3api-endpoint-ip --profile your-profile
```

이 명령에 대한 자세한 내용은 AWS CLI Command Reference의 [delete-object](#) 섹션을 참조하세요.

다음 Amazon S3 compatible storage on Snowball Edge 예제에서는 SDK for Java를 사용하여 버킷의 객체를 삭제합니다. 이 예시를 사용하려면 삭제할 객체의 키 이름을 지정합니다. 자세한 내용은 Amazon Simple Storage Service API Reference에서 [DeleteObject](#) 섹션을 참조하세요.

```

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.DeleteObjectRequest;

public class DeleteObject {

```

```
public static void main(String[] args) {
    String bucketName = "**** Bucket name ****";
    String keyName = "**** key name ****";

    try {
        // This code expects that you have AWS credentials set up per:
        // https://docs.aws.amazon.com/sdk-for-java/v1/developer-guide/setup-
credentials.html
        AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
            .enableUseArnRegion()
            .build();

        DeleteObjectRequest deleteObjectRequest = DeleteObjectRequest.builder()
            .bucket(bucketName)
            .key(keyName)
            .build());
        s3Client.deleteObject(deleteObjectRequest);
    } catch (AmazonServiceException e) {
        // The call was transmitted successfully, but Amazon S3 couldn't process
        // it, so it returned an error response.
        e.printStackTrace();
    } catch (SdkClientException e) {
        // Amazon S3 couldn't be contacted for a response, or the client
        // couldn't parse the response from Amazon S3.
        e.printStackTrace();
    }
}
```

## Snowball Edge의 Amazon S3 호환 스토리지에 지원되는 REST API 작업

다음 목록은의 Amazon S3 관련 작업에 대한 링크를 포함하여 Snowball Edge의 Amazon S3 호환 스토리지에서 지원하는 API 작업을 보여줍니다 AWS 리전.

s3api 엔드포인트에 지원되는 버킷 API 작업:

- [CreateBucket](#)
- [DeleteBucket](#)
- [DeleteBucketLifecycle](#)

- [GetBucketLifecycleConfiguration](#)
- [ListBuckets](#)
- [PutBucketLifecycleConfiguration](#)

s3control 엔드포인트에 지원되는 버킷 API 작업:

- [CreateBucket](#)
- [DeleteBucket](#)
- [DeleteBucketLifecycle](#)
- [GetBucket](#)
- [GetBucketLifecycleConfiguration](#)
- [ListBuckets](#)
- [PutBucketLifecycleConfiguration](#)

지원되는 객체 API 작업:

- [AbortMultipartUpload](#)
- [CompleteMultipartUpload](#)
- [CopyObject](#)
- [CreateMultipartUpload](#)
- [DeleteObject](#)
- [DeleteObjects](#)
- [DeleteObjectTagging](#)
- [GetObject](#)
- [GetObjectTagging](#)
- [HeadBucket](#)
- [HeadObject](#)
- [ListMultipartUploads](#)
- [ListObjects](#)
- [ListObjectsV2](#)
- [ListParts](#)
- [PutObject](#)



- [PutObjectTagging](#)
- [UploadPart](#)
- [UploadPartCopy](#)

## Snow 디바이스 클러스터와 함께 Snowball Edge에서 Amazon S3 호환 스토리지 사용

클러스터는 로컬 스토리지 및 컴퓨팅 용도를 위해 단일 논리 단위로 사용되는 3개 이상의 Snowball Edge 디바이스 집합에 해당합니다. 클러스터는 로컬 스토리지 및 컴퓨팅을 위해 독립형 Snowball Edge 디바이스에 두 가지 주요 장점을 제공합니다.

- 향상된 내구성 - Snowball Edge 디바이스의 클러스터에 저장된 S3 데이터는 단일 디바이스에 대해 내구성이 향상됩니다. 또한 클러스터의 데이터는 클러스터에서 하드웨어가 중단되는 경우에도 안전하고 실행 가능한 상태를 유지합니다. 클러스터는 데이터가 위험에 처하기 전에 3~4개 디바이스의 클러스터에서 디바이스 하나, 5~16개 디바이스의 클러스터에서 최대 두 개의 디바이스가 손실되어도 견딜 수 있습니다. 비정상 노드를 교체하여 클러스터에 저장된 데이터의 내구성과 안전성을 유지할 수 있습니다.
- 스토리지 증가 - Snowball Edge 스토리지 최적화 디바이스를 사용하면 최대 2.6PB의 사용 가능한 S3-compatible 스토리지 용량을 갖춘 단일 16 노드 클러스터를 생성할 수 있습니다. Snowball Edge 컴퓨팅 최적화 디바이스를 사용하면 최대 501TB의 사용 가능한 S3-compatible 스토리지 용량으로 구성된 단일 16 노드 클러스터를 생성할 수 있습니다.

Snowball Edge 디바이스의 클러스터는 리더리스 노드로 구성됩니다. 노드는 전체 클러스터에 대해 데이터 쓰기 및 읽기가 가능하며, 모든 노드는 클러스터의 백그라운드 관리를 수행할 수 있습니다.

Snowball Edge 클러스터 사용을 계획할 때에는 다음 고려 사항에 유의하세요.

- 클러스터에 발생할 수 있는 성능 및 안정성 문제를 줄이려면 클러스터의 모든 디바이스에 대해 중복 전원 장치를 제공하는 것이 좋습니다.
- 독립형 로컬 스토리지 및 컴퓨팅 작업과 마찬가지로 별도의 가져오기 작업의 일부로 추가 디바이스를 주문하지 않고는 클러스터에 저장된 데이터를 Amazon S3로 가져올 수 없습니다. 추가 디바이스를 가져오기 작업으로 주문하는 경우 클러스터의 데이터를 가져오기 작업 디바이스로 전송할 수 있습니다.
- Amazon S3에서 클러스터로 데이터를 가져오려면 Amazon S3 API를 사용하여 클러스터에 Amazon S3 버킷을 생성하여 S3에서 객체를 저장하고 검색합니다. 또한 AWS DataSync 를 사용하여

Snowball Edge 디바이스의 Snowball Edge에서 AWS 스토리지 서비스와 Amazon S3 호환 스토리지 간에 객체를 전송할 수 있습니다. 자세한 내용은 [Snowball Edge의 S3 호환 스토리지를 사용한 전송 구성](#)을 참조하세요.

- 작업을 생성하여 AWS Snow 패밀리 관리 콘솔 AWS CLI, 또는 AWS SDKs. 자세한 내용은 [Snowball Edge 시작하기](#) 단원을 참조하십시오.
- 클러스터의 각 디바이스에는 노드 ID가 있습니다. 노드 ID는 독립 실행형 디바이스의 작업 ID와 같이 클러스터의 각 디바이스에 대한 고유 식별자입니다. 노드 IDs AWS Snow 패밀리 관리 콘솔, , AWS CLI AWS SDKs 및 Snowball Edge 클라이언트에서 가져올 수 있습니다. Snowball Edge 클라이언트 명령 describe-device 및 describe-cluster는 디바이스 또는 클러스터에 대한 다른 정보와 함께 노드 ID를 반환합니다.
- 클러스터의 수명은 클러스터가 프로비저닝될 때 클러스터 디바이스에 부여된 보안 인증서에 따라 정해집니다. 기본적으로 Snowball Edge 디바이스는 최대 360일간 사용한 후 반환해야 합니다. 그 기간이 종료되면 디바이스는 읽기/쓰기 요청에 응답하지 않습니다. 하나 이상의 디바이스를 360일 이상 보관해야 하는 경우에 문의하세요 AWS Support.
- 가 클러스터의 일부인 반환된 디바이스를 AWS 수신하면 디바이스를 완전히 삭제합니다. 이 삭제는 NIST(National Institute of Standards and Technology) 800-88 표준에 따른 것입니다.

Amazon S3 compatible storage on Snowball Edge 클러스터 내결함성 및 스토리지 용량

클러스터 크기	내결함성	Snowball Edge Compute Optimized의 스토리지 용량(AMD EPYC Gen2 및 NVMe GPU 포함 Compute Optimized)(TB)	Snowball Edge 스토리지 최적화 210TB 디바이스의 스토리지 용량 (TB)
3	최대 1개 노드 손실	38	438
4	최대 1개 노드 손실	57	657
5	최대 2개 노드 손실	57	657
6	최대 2개 노드 손실	76	904
7	최대 2개 노드 손실	95	1096
8	최대 2개 노드 손실	114	1315

클러스터 크기	내결함성	Snowball Edge Compute Optimized의 스토리지 용량(AMD EPYC Gen2 및 NVMe GPU 포함 Compute Optimized)(TB)	Snowball Edge 스토리지 최적화 210TB 디바이스의 스토리지 용량 (TB)
9	최대 2개 노드 손실	133	1534
10	최대 2개 노드 손실	152	1754
11	최대 2개 노드 손실	165	1970
12	최대 2개 노드 손실	171	1973
13	최대 2개 노드 손실	190	2192
14	최대 2개 노드 손실	209	2411
15	최대 2개 노드 손실	225	2625
16	최대 2개 노드 손실	228	2631

클러스터를 잠금 해제하면 해당 클러스터에서 데이터를 저장하고 액세스할 준비가 된 것입니다. Amazon S3 호환 엔드포인트를 사용하여 클러스터에서 데이터를 읽고 클러스터에 데이터를 쓸 수 있습니다.

클러스터에 데이터를 읽거나 쓰려면 사용할 수 없는 노드 허용 수보다 적은 읽기/쓰기 쿼럼이 디바이스 클러스터에 있어야 합니다.

## Snowball Edge 클러스터 쿼럼

쿼럼은 클러스터에서 읽기/쓰기 쿼럼을 유지하기 위해 서로 통신해야 하는 Snowball Edge 디바이스의 최소 수를 나타냅니다.

클러스터의 모든 디바이스가 정상인 경우에는 클러스터에 대한 읽기/쓰기 쿼럼이 존재합니다. 해당 디바이스 중 하나 이상이 오프라인이 되는 경우 클러스터의 운영 용량이 줄어든 것입니다. 하지만 그래도 클러스터에 읽기 및 쓰기가 가능합니다. 클러스터가 하나 또는 두 개 이상 이외의 모든 노드를 작동하

면 클러스터에는 계속 읽기/쓰기 쿼럼이 있습니다. 클러스터의 운영 용량이 영향을 받기 전에 오프라인 상태가 될 수 있는 노드 수는 [this table](#)에서 확인할 수 있습니다.

클러스터에서 디바이스 수가 [this table](#)에 표시된 수보다 많이 손실되면 쿼럼이 손실될 수 있습니다. 쿼럼이 손실되면 클러스터가 오프라인이 되고 클러스터의 데이터를 사용할 수 없습니다. 이 문제를 해결할 수도 있지만 해결하지 못하는 경우 이벤트의 심각도에 따라 데이터가 영구적으로 손실될 수 있습니다. 임시 외부 전원 이벤트인 경우 Snowball Edge 디바이스에 전원을 다시 공급하고 클러스터의 모든 노드를 잠금 해제하면 데이터를 다시 사용할 수 있습니다.

### Important

정상 노드의 최소 쿼럼이 존재하지 않는 경우 AWS Support에 문의하세요.

describe-cluster 명령을 사용하여 각 노드의 잠금 상태 및 네트워크 연결 가능성을 볼 수 있습니다. 클러스터 스토리지를 사용할 때 클러스터 디바이스가 정상이고 연결되어 있는지 확인할 관리 책임은 사용자에게 있습니다. 자세한 내용은 [디바이스 상태 가져오기](#)를 참조하십시오.

하나 이상의 노드가 비정상이라고 판단되면 클러스터의 노드를 교체하여 할당량과 데이터의 상태 및 안정성을 유지할 수 있습니다. 자세한 내용은 [클러스터에서 노드 교체](#) 단원을 참조하십시오.

## 사용할 수 없는 클러스터 노드 다시 연결

노드 또는 클러스터 내 디바이스는 어떤 문제(전원 또는 네트워크 끊김)로 인해 노드의 데이터가 손상되지 않고 일시적으로 사용할 수 없게 될 수 있습니다. 이러한 일이 발생하는 경우 클러스터의 상태에 영향을 줍니다. 노드의 네트워크 연결성과 잠금 상태는 snowballEdge describe-cluster 명령을 사용하여 Snowball Edge 클라이언트에서 보고됩니다.

클러스터를 모든 노드의 전면, 후면 및 상단에 접근할 수 있도록 물리적으로 배치하는 것이 좋습니다. 그렇게 하면 후면의 전원 및 네트워크 케이블, 상단의 배송 라벨에 접근하여 노드 ID를 확인하고 디바이스의 전면에 있는 LCD 화면에 접근하여 IP 주소 및 기타 관리 정보를 확인할 수 있습니다.

노드를 사용할 수 없음을 감지하면 사용 불가능의 원인이 된 시나리오에 따라 다음 절차 중 하나를 시도해 보는 것이 좋습니다.

사용할 수 없는 노드를 다시 연결하려면

1. 노트가 켜져 있는지 확인합니다.
2. 노드가 나머지 클러스터가 연결된 것과 동일한 내부 네트워크에 연결되어 있는지 확인합니다.

3. 노드 전원을 켜야 하는 경우 완료될 때까지 최대 20분 정도 기다리세요.
4. `snowballEdge unlock-cluster` 명령이나 `snowballEdge associate-device` 명령을 실행합니다. 예시는 [Unlocking Snowball Edge devices](#) 섹션을 참조하세요.

네트워크가 끊겼지만 전원이 끊기지 않은 사용할 수 없는 노드 다시 연결

1. 노드가 나머지 클러스터가 연결된 것과 동일한 내부 네트워크에 연결되어 있는지 확인합니다.
2. `snowballEdge describe-device` 명령을 실행하여 사용할 수 없는 노드가 클러스터에 다시 추가되도록 합니다. 예시는 [Getting Device Status](#) 섹션을 참조하세요.

이전 절차를 수행한 후에는 노드가 정상적으로 작동해야 합니다. 읽기/쓰기 쿼럼도 있어야 합니다. 그렇지 않은 경우 노드 중 하나 이상에 심각한 문제가 있어 클러스터에서 제거해야 할 수 있습니다.

## 클러스터에서 노드 교체

노드를 교체하려면 먼저 대체품을 주문해야 합니다. 콘솔, AWS CLI 또는 SDK 중 하나에서 대체 노드를 주문할 수 있습니다. SDKs 콘솔에서 대체 노드를 주문하는 경우 취소 또는 완료되지 않은 작업에 대한 대체품을 주문할 수 있습니다. 그런 다음 클러스터에서 비정상 노드를 연결 해제하고, 대체 노드를 네트워크에 연결하고, 대체 노드를 포함한 클러스터를 잠금 해제하고, 대체 노드를 클러스터와 연결하고, Amazon S3 compatible storage on Snowball Edge 서비스를 다시 시작합니다.

콘솔에서 대체 노드를 주문하려면

1. [AWS Snow 패밀리 관리 콘솔](#)에 로그인합니다.
2. 작업 대시보드에서 생성한 클러스터에 속한 노드에 대한 작업을 확인하고 선택합니다.
3. [Actions]에 대해 [Replace node]를 선택합니다.

이렇게 하면 설정이 클러스터가 원래 생성된 방법과 동일한 작업 생성 마법사의 마지막 단계가 열립니다.

4. 작업 생성을 선택합니다.

현재 대체 Snowball Edge가 배송 중입니다. 다음 절차에 따라 클러스터에서 비정상 노드를 제거합니다.

클러스터에서 노드를 제거하려면

1. 제거할 노드의 전원을 끕니다. 자세한 내용은 [Snowball Edge 전원 끄기](#)를 참조하십시오.

2. `describe-cluster` 명령을 사용하여 비정상 노드에 연결할 수 없도록 합니다. 이는 `NetworkReachability` 객체의 `State` 이름에 대한 `UNREACHABLE` 값으로 표시됩니다.

```
snowballEdge describe-cluster --manifest-file path/to/manifest/file.bin --unlock-code unlock-code --endpoint https://ip-address-of-device-in-cluster
```

### Example `describe-cluster` 출력

```
{
  "ClusterId": "CID12345678-1234-1234-1234-123456789012",
  "Devices": [
    {
      "DeviceId": "JID12345678-1234-1234-1234-123456789012",
      "UnlockStatus": {
        "State": "UNLOCKED"
      },
      "ActiveNetworkInterface": {
        "IpAddress": "10.0.0.0"
      },
      "ClusterAssociation": {
        "ClusterId": "CID12345678-1234-1234-1234-123456789012",
        "State": "ASSOCIATED"
      },
      "NetworkReachability": {
        "State": "REACHABLE"
      },
      "Tags": []
    },
    {
      "DeviceId": "JID12345678-1234-1234-1234-123456789013",
      "UnlockStatus": {
        "State": "UNLOCKED"
      },
      "ActiveNetworkInterface": {
        "IpAddress": "10.0.0.1"
      },
      "ClusterAssociation": {
        "ClusterId": "CID12345678-1234-1234-1234-123456789012",
        "State": "ASSOCIATED"
      }
    }
  ]
}
```

```

    },
    "NetworkReachability": {
        "State": "REACHABLE"
    },
    "Tags": []
},
{
    "DeviceId": "JID12345678-1234-1234-1234-123456789014",
    "ClusterAssociation": {
        "ClusterId": "CID12345678-1234-1234-1234-123456789012",
        "State": "ASSOCIATED"
    },
    "NetworkReachability": {
        "State": "UNREACHABLE"
    }
}
]
}

```

3. `describe-service` 명령을 사용하여 `s3-snow` 서비스의 상태가 `DEGRADED`인지 확인합니다.

```

snowballEdge describe-service --service-id s3-snow --device-ip-addresses snow-device-1-address snow-device-2-address --manifest-file path/to/manifest/file.bin --unlock-code unlock-code --endpoint https://snow-device-ip-address

```

#### Example/ describe-service 명령의 출력

```

{
  "ServiceId": "s3-snow",
  "Autostart": true,
  "Status": {
    "State": "DEGRADED"
  },
  "ServiceCapacities": [
    {
      "Name": "S3 Storage",
      "Unit": "Byte",
      "Used": 38768180432,
      "Available": 82961231819568
    }
  ]
}

```

```

    }
  ],
  "Endpoints": [
    {
      "Protocol": "https",
      "Port": 443,
      "Host": "10.0.0.10",
      "CertificateAssociation": {
        "CertificateArn": "arn:aws:snowball-
device:::certificate/7Rg2lP9tQaHnW4sC6xUzF1vGyD3jB5kN8MwEiYpT"
      },
      "Description" : "s3-snow bucket API endpoint (for s3control SDK)",
      "DeviceId": "JID-beta-207012320001-24-02-05-17-17-26",
      "Status": {
        "State": "ACTIVE"
      }
    },
    {
      "Protocol": "https",
      "Port": 443,
      "Host": "10.0.0.11",
      "CertificateAssociation": {
        "CertificateArn": "arn:aws:snowball-
device:::certificate/7Rg2lP9tQaHnW4sC6xUzF1vGyD3jB5kN8MwEiYpT"
      },
      "Description": "Description" : "s3-snow object & bucket API endpoint
(for s3api SDK)",
      "DeviceId": "JID-beta-207012320001-24-02-05-17-17-26",
      "Status": {
        "State": "ACTIVE"
      }
    },
    {
      "Protocol": "https",
      "Port": 443,
      "Host": "10.0.0.12",
      "CertificateAssociation": {
        "CertificateArn": "arn:aws:snowball-
device:::certificate/7Rg2lP9tQaHnW4sC6xUzF1vGyD3jB5kN8MwEiYpT"
      },
      "Description": "Description" : "s3-snow bucket API endpoint (for
s3control SDK)",
      "DeviceId": "JID-beta-207012240003-24-02-05-17-17-27",
      "Status": {

```



```

        "State": "ACTIVE"
      }
    },
    {
      "Protocol": "https",
      "Port": 443,
      "Host": "10.0.0.13",
      "CertificateAssociation": {
        "CertificateArn": "arn:aws:snowball-
device::certificate/7Rg2lP9tQaHnW4sC6xUzF1vGyD3jB5kN8MwEiYpT"
      },
      "Description": "Description" : "s3-snow object & bucket API endpoint
(for s3api SDK)",
      "DeviceId": "JID-beta-207012320001-24-02-05-17-17-27",
      "Status": {
        "State": "ACTIVE"
      }
    }
  ]
}

```

4. `disassociate-device` 명령을 사용하여 클러스터에서 비정상 노드의 연결을 해제하고 제거합니다.

```

snowballEdge disassociate-device --device-id device-id --manifest-file path/to/manifest/file.bin --unlock-code unlock-code --endpoint https://ip-address-of-unhealthy-device

```

#### Example `disassociate-device` 명령의 출력

Disassociating your Snowball Edge device from the cluster. Your Snowball Edge device will be disassociated from the cluster when it is in the "DISASSOCIATED" state. You can use the `describe-cluster` command to determine the state of your cluster.

5. `describe-cluster` 명령을 다시 사용하여 비정상 노드가 클러스터에서 연결 해제되었는지 확인합니다.

```
snowballEdge describe-cluster --manifest-file path/to/manifest/file.bin --unlock-code unlock-code --endpoint https:ip-address-of-healthy-device
```

Example/ 노드가 연결 해제되었음을 보여주는 describe-cluster 명령

```
{
  "ClusterId": "CID12345678-1234-1234-1234-123456789012",
  "Devices": [
    {
      "DeviceId": "JID12345678-1234-1234-1234-123456789012",
      "UnlockStatus": {
        "State": "UNLOCKED"
      },
      "ActiveNetworkInterface": {
        "IpAddress": "10.0.0.0"
      },
      "ClusterAssociation": {
        "ClusterId": "CID12345678-1234-1234-1234-123456789012",
        "State": "ASSOCIATED"
      },
      "NetworkReachability": {
        "State": "REACHABLE"
      },
      "Tags": []
    },
    {
      "DeviceId": "JID12345678-1234-1234-1234-123456789013",
      "UnlockStatus": {
        "State": "UNLOCKED"
      },
      "ActiveNetworkInterface": {
        "IpAddress": "10.0.0.1"
      },
      "ClusterAssociation": {
        "ClusterId": "CID12345678-1234-1234-1234-123456789012",
        "State": "ASSOCIATED"
      },
      "NetworkReachability": {
        "State": "REACHABLE"
      }
    }
  ]
}
```

```

    },
    "Tags": []
  },
  {
    "DeviceId": "JID12345678-1234-1234-1234-123456789014",
    "ClusterAssociation": {
      "ClusterId": "CID12345678-1234-1234-1234-123456789012",
      "State": "DISASSOCIATED"
    }
  }
]
}

```

- 전원을 끄고 비정상 디바이스에 반환합니다 AWS. 자세한 내용은 [Snowball Edge 전원 끄기 및 Snowball Edge 디바이스 반환](#)을 참조하세요.

교체 디바이스가 도착하면 다음 절차에 따라 클러스터에 추가합니다.

교체 디바이스를 추가하려면

- 클러스터의 교체 디바이스를 모든 디바이스의 전면, 후면 및 상단에 액세스할 수 있도록 배치합니다.
- 노드의 전원을 켜고 노드가 나머지 클러스터와 동일한 내부 네트워크에 연결되어 있는지 확인합니다. 자세한 내용은 [로컬 네트워크에 연결](#)을 참조하세요.
- `unlock-cluster` 명령을 사용하고 새 노드의 IP 주소를 포함합니다.

```

snowballEdge unlock-cluster --manifest-file path/to/manifest/file.bin --unlock-code unlock-code --endpoint https://ip-address-of-cluster-device --device-ip-addresses node-1-ip-address node-2-ip-address new-node-ip-address

```

새 노드의 상태는 다음 단계에서 클러스터와 연결할 때까지 DEGRADED입니다.

- `associate-device` 명령을 사용하여 교체 노드를 클러스터와 연결합니다.

```

snowballEdge associate-device --device-ip-address new-node-ip-address

```

## Example/ associate-device 명령 출력

Associating your Snowball Edge device with the cluster. Your Snowball Edge device will be associated with the cluster when it is in the ASSOCIATED state. You can use the describe-device command to determine the state of your devices.

5. describe-cluster 명령을 사용하여 새 노드가 클러스터와 연결되어 있는지 확인합니다.

```
snowballEdge describe-cluster --manifest-file path/to/manifest/file.bin --unlock-code unlock-code --endpoint https://node-ip-address
```

## Example/ describe-cluster 명령 출력

```
{
  "ClusterId": "CID12345678-1234-1234-1234-123456789012",
  "Devices": [
    {
      "DeviceId": "JID12345678-1234-1234-1234-123456789012",
      "UnlockStatus": {
        "State": "UNLOCKED"
      },
      "ActiveNetworkInterface": {
        "IpAddress": "10.0.0.0"
      },
      "ClusterAssociation": {
        "ClusterId": "CID12345678-1234-1234-1234-123456789012",
        "State": "ASSOCIATED"
      },
      "NetworkReachability": {
        "State": "REACHABLE"
      },
      "Tags": []
    },
    {
      "DeviceId": "JID-CID12345678-1234-1234-1234-123456789013",
      "UnlockStatus": {
```

```

        "State": "UNLOCKED"
    },
    "ActiveNetworkInterface": {
        "IpAddress": "10.0.0.1"
    },
    "ClusterAssociation": {
        "ClusterId": "CID12345678-1234-1234-1234-123456789012",
        "State": "ASSOCIATED"
    },
    "NetworkReachability": {
        "State": "REACHABLE"
    },
    "Tags": []
},
{
    "DeviceId": "JID-CID12345678-1234-1234-1234-123456789015",
    "UnlockStatus": {
        "State": "UNLOCKED"
    },
    "ActiveNetworkInterface": {
        "IpAddress": "10.0.0.2"
    },
    "ClusterAssociation": {
        "ClusterId": "CID12345678-1234-1234-1234-123456789012",
        "State": "ASSOCIATED"
    },
    "NetworkReachability": {
        "State": "REACHABLE"
    },
    "Tags": []
}
]
}

```

6. 새 노드에서 두 개의 가상 네트워크 인터페이스(VNI)를 생성합니다. 자세한 내용은 [Snowball Edge 서비스에서 Amazon S3 호환 스토리지 시작](#) 섹션을 참조하세요.
7. `stop-service` 명령을 사용하여 `s3-snow` 서비스를 중지합니다.

```

snowballEdge stop-service --service-id s3-snow --device-ip-addresses cluster-
device-1-ip-address cluster-device-2-ip-address cluster-device-3-ip-address --

```

```
manifest-file path/to/manifest/file.bin --unlock-code unlock-code --endpoint
https://snow-device-ip-address
```

### Example/ stop-service 명령 출력

Stopping the AWS service on your Snowball Edge. You can determine the status of the AWS service using the describe-service command.

8. start-service 명령을 사용하여 클러스터에 새 노드를 추가한 후 s3-snow 서비스를 시작합니다.

```
snowballEdge start-service --service-id s3-snow --device-ip-addresses cluster-
device-1-ip-address cluster-device-2-ip-address cluster-device-3-ip-address --
virtual-network-interface-arns "device-1-vni-ip-address-a" "device-1-vni-ip-
address-b" "device-2-vni-ip-address-a" "device-2-vni-ip-address-b" "device-3-vni-
ip-address-a" "device-3-vni-ip-address-b" --manifest-file path/to/manifest/file.bin
--unlock-code unlock-code --endpoint https://snow-device-ip-address
```

### Example/ start-service 명령 출력

Starting the AWS service on your Snowball Edge. You can determine the status of the AWS service using the describe-service command.

9. describe-service 명령을 사용하여 s3-snow 서비스가 시작되었는지 확인합니다.

```
snowballEdge describe-service --service-id s3-snow --device-ip-addresses snow-
device-1-address snow-device-2-address snow-device-3-address --manifest-file path/
to/manifest/file.bin --unlock-code unlock-code --endpoint https://snow-device-ip-
address
```

## Example/ descibe-service 명령 출력

```

{
  "ServiceId": "s3-snow",
  "Autostart": true,
  "Status": {
    "State": "ACTIVE"
  },
  "ServiceCapacities": [{
    "Name": "S3 Storage",
    "Unit": "Byte",
    "Used": 38768180432,
    "Available": 82961231819568
  }],
  "Endpoints": [{
    "Protocol": "https",
    "Port": 443,
    "Host": "10.0.0.10",
    "CertificateAssociation": {
      "CertificateArn": "arn:aws:snowball-
device:::certificate/7Rg2lP9tQaHnW4sC6xUzF1vGyD3jB5kN8MwEiYpT"
    },
    "Description": "s3-snow bucket API endpoint (for s3control SDK)",
    "DeviceId": "JID12345678-1234-1234-1234-123456789012",
    "Status": {
      "State": "ACTIVE"
    }
  }, {
    "Protocol": "https",
    "Port": 443,
    "Host": "10.0.0.11",
    "CertificateAssociation": {
      "CertificateArn": "arn:aws:snowball-
device:::certificate/7Rg2lP9tQaHnW4sC6xUzF1vGyD3jB5kN8MwEiYpT"
    },
    "Description": "s3-snow object & bucket API endpoint (for s3api SDK)",
    "DeviceId": "JID12345678-1234-1234-1234-123456789013",
    "Status": {
      "State": "ACTIVE"
    }
  }, {
    "Protocol": "https",

```

```
    "Port": 443,
    "Host": "10.0.0.12",
    "CertificateAssociation": {
      "CertificateArn": "arn:aws:snowball-
device:::certificate/7Rg2lP9tQaHnW4sC6xUzF1vGyD3jB5kN8MwEiYpT"
    },
    "Description": "s3-snow bucket API endpoint (for s3control SDK)",
    "DeviceId": "JID12345678-1234-1234-1234-123456789015",
    "Status": {
      "State": "ACTIVE"
    }
  }, {
    "Protocol": "https",
    "Port": 443,
    "Host": "10.0.0.13",
    "CertificateAssociation": {
      "CertificateArn": "arn:aws:snowball-
device:::certificate/7Rg2lP9tQaHnW4sC6xUzF1vGyD3jB5kN8MwEiYpT"
    },
    "Description": "s3-snow object & bucket API endpoint (for s3api SDK)",
    "DeviceId": "JID-beta-207012320001-24-02-05-17-17-27",
    "Status": {
      "State": "ACTIVE"
    }
  }, {
    "Protocol": "https",
    "Port": 443,
    "Host": "10.0.0.14",
    "CertificateAssociation": {
      "CertificateArn": "arn:aws:snowball-
device:::certificate/7Rg2lP9tQaHnW4sC6xUzF1vGyD3jB5kN8MwEiYpT"
    },
    "Description": "s3-snow bucket API endpoint (for s3control SDK)",
    "DeviceId": "JID-beta-207012240003-24-02-05-17-17-28",
    "Status": {
      "State": "ACTIVE"
    }
  }, {
    "Protocol": "https",
    "Port": 443,
    "Host": "10.0.0.15",
    "CertificateAssociation": {
      "CertificateArn": "arn:aws:snowball-
device:::certificate/7Rg2lP9tQaHnW4sC6xUzF1vGyD3jB5kN8MwEiYpT"
```



```

    },
    "Description": "s3-snow object & bucket API endpoint (for s3api SDK),
    "DeviceId": "JID-beta-207012320001-24-02-05-17-17-28",
    "Status": {
        "State": "ACTIVE"
    }
}
}]
}

```

## Snowball Edge에서 Amazon S3 호환 스토리지 이벤트 알림 구성

Snowball Edge의 Amazon S3 호환 스토리지는 MQTT(Message Queuing Telemetry Transport) 프로토콜을 기반으로 객체 API 호출에 대한 Amazon S3 이벤트 알림을 지원합니다.

Snowball Edge의 Amazon S3 호환 스토리지를 사용하여 S3 버킷에서 특정 이벤트가 발생할 때 알림을 받을 수 있습니다. 알림을 사용 설정하려면 게시하려는 이벤트를 식별하는 알림 구성을 추가합니다.

Snowball Edge의 Amazon S3 호환 스토리지는 다음 알림 유형을 지원합니다.

- 새 객체 생성 이벤트
- 객체 제거 이벤트
- 객체 태깅 이벤트

### Amazon S3 이벤트 알림 구성

1. 시작하기 전에 네트워크에 MQTT 인프라가 있어야 합니다.
2. Snowball Edge 클라이언트에서 `snowballEdge configure` 명령을 실행하여 Snowball Edge 디바이스를 설정합니다.

요청 메시지가 나타나면 다음 정보를 입력합니다.

- 매니페스트 파일 경로.
  - 디바이스의 잠금 해제 코드.
  - 디바이스의 엔드포인트(예: <https://10.0.0.1>).
3. 다음 `put-notification-configuration` 명령을 실행하여 외부 브로커에 알림을 전송합니다.

```
snowballEdge put-notification-configuration --broker-endpoint ssl://mqtt-broker-
ip-address:8883 --enabled true --service-id s3-snow --ca-certificate file:path-to-
mqtt-broker-ca-cert
```

4. 다음 `get-notification-configuration` 명령을 실행하여 모든 항목이 올바르게 설정되었는지 확인합니다.

```
snowballEdge get-notification-configuration --service-id s3-snow
```

그러면 브로커 엔드포인트와 활성화 필드가 반환됩니다.

네트워크의 MQTT 브로커에 알림을 보내도록 전체 클러스터를 구성한 후에는 모든 객체 API 직접 호출 시 이벤트 알림이 발생합니다.

#### Note

`s3SnowEvents/Device ID`(또는 클러스터인 경우 `Cluster ID`)/`bucketName` 항목을 구독해야 합니다. 와일드카드를 사용할 수도 있습니다. 예를 들어 주제 이름은 `#` 또는 `s3SnowEvents/#`일 수 있습니다.

다음은 Snowball Edge의 Amazon S3 호환 스토리지 이벤트 로그의 예입니다.

```
{
  "eventDetails": {
    "additionalEventData": {
      "AuthenticationMethod": "AuthHeader",
      "CipherSuite": "ECDHE-RSA-AES128-GCM-SHA256",
      "SignatureVersion": "SigV4",
      "bytesTransferredIn": 1205,
      "bytesTransferredOut": 0,
      "x-amz-id-2": "uLdTfvdGTK1X6TBgCZtDd9Beef8wzUurA+Wpht7rKtfdaNsnxeLILg=="
    },
    "eventName": "PutObject",
    "eventTime": "2023-01-30T14:13:24.772Z",
    "requestAuthLatencyMillis": 40,
    "requestBandwidthKBs": 35,
    "requestID": "140CD93455CB62B4",
  }
}
```

```

    "requestLatencyMillis": 77,
    "requestLockLatencyNanos": 1169953,
    "requestParameters": {
      "Content-Length": "1205",
      "Content-MD5": "GZdTU0hYHvHgQgmaw2g14w==",
      "Host": "10.0.2.251",
      "bucketName": "bucket",
      "key": "file-key"
    },
    "requestTTFBLatencyMillis": 77,
    "responseElements": {
      "ETag": "\"19975350e8581ef1e042099ac36825e3\"",
      "Server": "AmazonS3",
      "x-amz-id-2": "uLdTfvdGTK1X6TBgCZtDd9Beef8wzUurA+Wpht7rKtfdaNsnxeLILg==",
      "x-amz-request-id": "140CD93455CB62B4"
    },
    "responseStatusCode": 200,
    "sourceIPAddress": "172.31.37.21",
    "userAgent": "aws-cli/1.27.23 Python/3.7.16 Linux/4.14.301-224.520.amzn2.x86_64
    boto3/1.29.23",
    "userIdentity": {
      "identityType": "IAMUser",
      "principalId": "531520547609",
      "arn": "arn:aws:iam::531520547609:root",
      "userName": "root"
    }
  }
}

```

Amazon S3 이벤트 알림에 대한 자세한 내용은 [Amazon S3 이벤트 알림](#) 섹션을 참조하세요.

## Snowball Edge에서 로컬 SMTP 알림 구성

SMTP(Simple Mail Transfer Protocol)로 Snowball Edge 디바이스에 대한 로컬 알림을 설정할 수 있습니다. 로컬 알림은 서비스 상태(활성, 성능 저하, 비활성)가 변경되거나 용량 사용을 임계값이 80%, 90% 또는 100%를 초과할 경우 구성된 서버로 이메일을 보냅니다.

시작하기 전에 다음을 확인합니다.

- 최신 Snowball Edge 클라이언트에 액세스할 수 있습니다.
- 디바이스 잠금이 해제되어 사용할 준비가 되었습니다.

- 디바이스를 인터넷(Amazon Simple Email Service 또는 외부 SMTP 서버를 사용하는 경우) 또는 로컬 SMTP 서버에 연결할 수 있습니다.

## 로컬 알림을 위한 Snowball Edge 구성

이메일 알림을 보내도록 Snowball Edge를 설정합니다.

SMTP 알림을 사용하도록 디바이스 구성

1. 다음 명령을 실행하여 SMTP 구성을 디바이스에 추가합니다.

```
# If you don't specify a port, port 587 is the default.
SMTP_ENDPOINT=your-local-smtp-server-endpoint:port

# For multiple email recipients, separate with commas
RECIPIENTS_LIST=your-email-address

snowballEdge put-notification-configuration \
  --service-id local-monitoring \
  --enabled true \
  --type smtp \
  --broker-endpoint "$SMTP_ENDPOINT" \
  --sender example-sender@domain.com \
  --recipients "$RECIPIENTS_LIST"
```

성공하면 `example-sender@domain.com`으로부터 테스트 이메일을 받게 됩니다.

2. 다음 `get-notification-configuration` 명령을 실행하여 업데이트된 구성을 테스트합니다.

```
snowballEdge get-notification-configuration \
  --service-id local-monitoring
```

암호나 인증서를 제공하더라도 이들은 응답에 포함되지 않습니다.

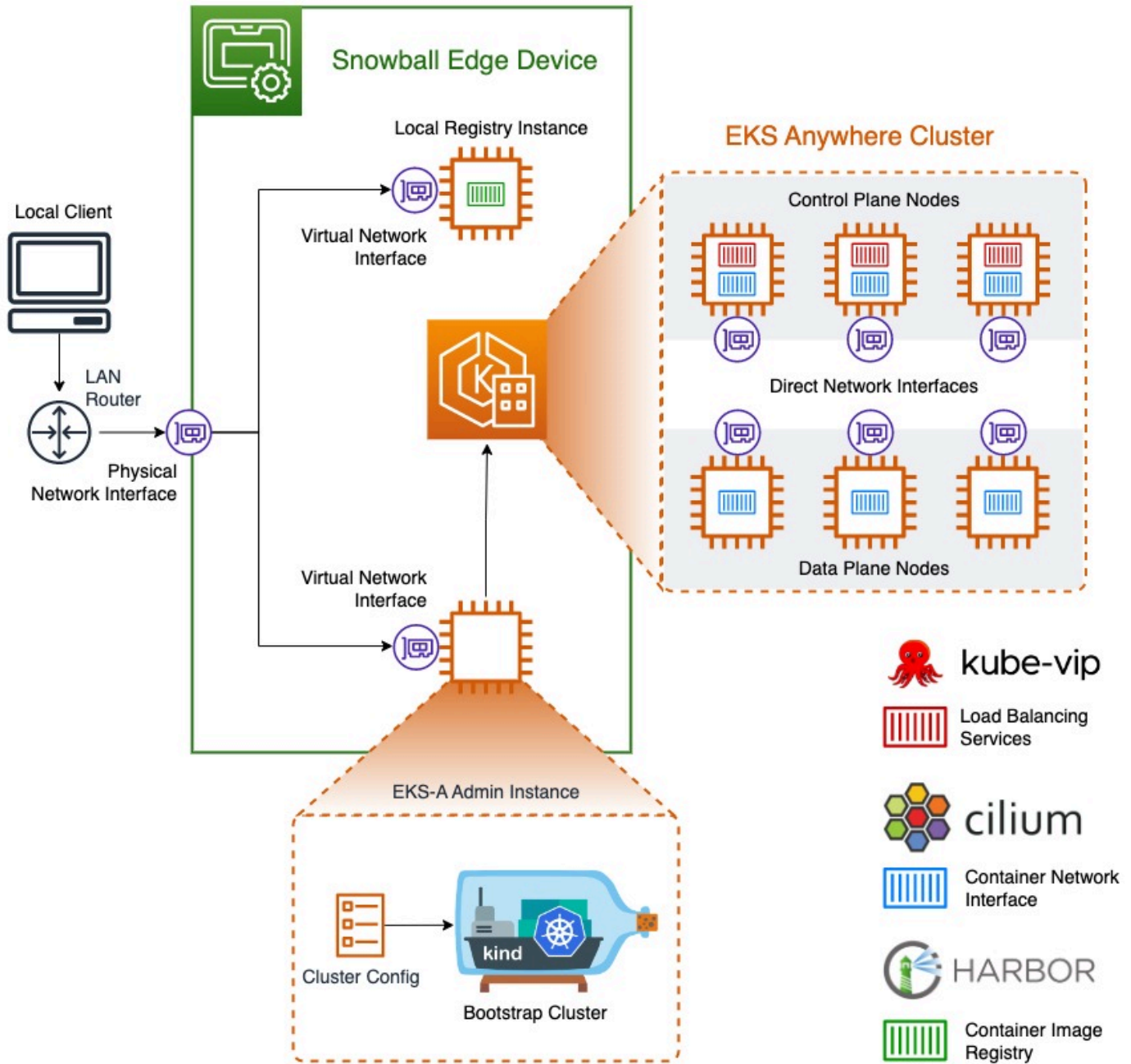
## AWS Snow에서 Amazon EKS Anywhere 사용

Amazon EKS Anywhere on AWS Snow는 Snowball Edge에서 Kubernetes 클러스터를 생성하고 운영하는 데 도움이 됩니다. Kubernetes는 컨테이너화된 애플리케이션의 배포, 조정 및 관리의 자동화를 위한 오픈 소스 소프트웨어입니다. 외부 네트워크 연결 여부에 관계없이 Snowball Edge 디바이스에서 Amazon EKS Anywhere를 사용할 수 있습니다. 외부 네트워크에 연결되지 않은 디바이스에서 Amazon EKS Anywhere를 사용하려면 Snowball Edge 디바이스에서 실행할 컨테이너 레지스트리를 제공하세요. Amazon EKS Anywhere에 대한 일반 정보는 [Amazon EKS Anywhere 설명서](#)를 참조하세요.

Amazon EKS Anywhere on AWS Snow를 사용하면 다음과 같은 기능을 사용할 수 있습니다.

- Snowball Edge 컴퓨팅 최적화 디바이스에서 Amazon EKS Anywhere CLI(eksctl anywhere)를 사용하여 Kubernetes(K8s) 클러스터를 프로비저닝하세요. Amazon EKS Anywhere는 단일 Snowball Edge 디바이스 또는 세 대 이상의 디바이스에서 프로비저닝하여고가용성을 확보할 수 있습니다.
- Cilium 컨테이너 네트워크 인터페이스(CNI)를 지원합니다.
- Ubuntu 20.04를 노드 운영 체제로 지원합니다.

이 다이어그램은 Snowball Edge 디바이스에 배포된 Amazon EKS Anywhere 클러스터를 보여줍니다.



Amazon EKS Anywhere에서 지원하는 사용 가능한 최신 Kubernetes 버전으로 Kubernetes 클러스터를 생성하는 것이 좋습니다. 자세한 내용은 [Amazon ECS Anywhere 버전 관리](#)를 참조하세요. 애플리케이션에 특정 버전의 Kubernetes가 필요한 경우 Amazon EKS에서 표준 또는 확장 지원으로 제공하는 모든 버전의 Kubernetes를 사용합니다. 배포 수명 주기를 계획할 때 Kubernetes 버전의 릴리스 및 지원 날짜를 고려합니다. 이렇게 하면 사용하려는 Kubernetes 버전에 대한 지원이 잠재적으로 손실되는 것을 방지할 수 있습니다. 자세한 내용은 [Amazon EKS Kubernetes 출시 일정](#)을 참조하세요.

Amazon EKS Anywhere on AWS Snow에 대한 자세한 내용은 [Amazon EKS Anywhere 설명서를 참조](#) 하세요.

## 주제

- [Amazon EKS Anywhere on Snow용 AWS Snowball Edge 디바이스 주문 전에 완료해야 할 작업](#)
- [Amazon EKS Anywhere on Snow와 함께 사용할 AWS Snowball Edge 디바이스 주문](#)
- [Snowball Edge 디바이스에서 Amazon EKS Anywhere를 구성하고 실행](#)
- [연결 해제된 작업을 위해 AWS Snow에서 Amazon EKS Anywhere 구성](#)
- [Snowball Edge 디바이스에서 클러스터 생성 및 유지 관리](#)

## Amazon EKS Anywhere on Snow용 AWS Snowball Edge 디바이스 주문 전에 완료해야 할 작업

현재 Amazon EKS Anywhere는 Snowball Edge 컴퓨팅 최적화 디바이스와 호환됩니다. Snowball Edge 디바이스를 주문하기 전에 몇 가지 준비해야 할 사항이 있습니다.

- 디바이스에서 가상 시스템을 생성하는 데 사용할 운영 체제 이미지를 빌드하고 제공하세요.
- 네트워크에는 K8s 컨트롤 플레인 엔드포인트에 사용할 수 있는 고정 IP 주소가 있어야 하며 주소 확인 프로토콜(ARP)을 허용해야 합니다.
- Snowball Edge 디바이스에는 특정 포트가 열려 있어야 합니다. 포트에 대한 자세한 내용은 Amazon EKS Anywhere 설명서의 [포트 및 프로토콜](#)을 참조하세요.

## 주제

- [Snowball Edge용 Ubuntu EKS Distro AMI 생성](#)
- [Snowball Edge용 Harbor AMI 빌드](#)

## Snowball Edge용 Ubuntu EKS Distro AMI 생성

Ubuntu EKS Distro AMI를 생성하려면 [Snow 노드 이미지 빌드](#)를 참조하세요.

생성된 AMI의 이름은 capa-ami-ubuntu-20.04-version-timestamp 패턴을 따릅니다. 예를 들어 capa-ami-ubuntu-20.04-v1.24-1672424524입니다.

## Snowball Edge용 Harbor AMI 빌드

Harbor 프라이빗 레지스트리 AMI를 Snowball Edge 디바이스에 포함하도록 설정하면 외부 네트워크 연결 없이 디바이스에서 Amazon EKS Anywhere를 사용할 수 있습니다. Snowball Edge 디바이스가 외부 네트워크에 연결되어 있지 않은 동안에는 Amazon EKS Anywhere를 사용하지 않을 예정이거나 AMI에 디바이스에서 사용할 프라이빗 Kubernetes 레지스트리가 있는 경우에는 이 섹션을 건너뛰어도 됩니다.

Harbor 로컬 레지스트리 AMI를 생성하려면 [Harbor AMI 구축](#)을 참조하세요.

## Amazon EKS Anywhere on Snow와 함께 사용할 AWS Snowball Edge 디바이스 주문

Snowball Edge 컴퓨팅 최적화를 주문하려면이 가이드 [Snowball Edge 디바이스 주문 작업 생성](#)의 섹션을 참조하고 주문 프로세스 중에 다음 항목을 염두에 두세요.

- 1단계에서 로컬 컴퓨팅 및 스토리지 전용 작업 유형을 선택합니다.
- 2단계에서 Snowball Edge 컴퓨팅 최적화 디바이스 유형을 선택합니다.
- 3단계에서 Amazon EKS Anywhere on AWS Snow를 선택한 다음 필요한 Kubernetes 버전을 선택합니다.

### Note

최신 소프트웨어를 제공하기 위해 현재 사용 가능한 버전보다 최신 버전의 ESK Anywhere로 디바이스를 구성할 수 있습니다. 자세한 내용은 Amazon EKS 사용 설명서의 [버전 관리](#)를 참조하세요.

Amazon EKS Anywhere에서 지원하는 사용 가능한 최신 Kubernetes 버전으로 Kubernetes 클러스터를 생성하는 것이 좋습니다. 자세한 내용은 [Amazon ECS Anywhere 버전 관리](#)를 참조하세요. 애플리케이션에 특정 버전의 Kubernetes가 필요한 경우 Amazon EKS에서 표준 또는 확장 지원으로 제공하는 모든 버전의 Kubernetes를 사용합니다. 배포 수명 주기를 계획할 때 Kubernetes 버전의 릴리스 및 지원 날짜를 고려합니다. 이렇게 하면 사용하려는 Kubernetes 버전에 대한 지원이 잠재적으로 손실되는 것을 방지할 수 있습니다. 자세한 내용은 [Amazon EKS Kubernetes 출시 일정](#)을 참조하세요.

- EKS Distro AMI([Snowball Edge용 Ubuntu EKS Distro AMI 생성](#) 참조) 및 선택적으로 구축한 Harbor AMI([Snowball Edge용 Harbor AMI 빌드](#) 참조)를 포함하여 디바이스에 포함할 AMI를 선택합니다.



- 고가용성을 위해 여러 Snowball Edge 디바이스가 필요한 경우 고가용성에서 필요한 디바이스 수를 선택합니다.

Snowball Edge 디바이스를 받은 후, [Snowball Edge 디바이스에서 Amazon EKS Anywhere를 구성하고 실행](#)에 따라 Amazon EKS Anywhere를 구성하세요.

## Snowball Edge 디바이스에서 Amazon EKS Anywhere를 구성하고 실행

다음 절차에 따라 Snowball Edge 디바이스에서 Amazon EKS Anywhere를 구성하고 시작하세요. 그런 다음 연결이 끊긴 디바이스에서도 작동하도록 Amazon EKS Anywhere를 구성하려면 외부 네트워크에서 디바이스 연결을 끊기 전에 추가 절차를 완료하세요. 자세한 내용은 [연결 해제된 작업을 위해 AWS Snow에서 Amazon EKS Anywhere 구성](#) 단원을 참조하십시오.

### 주제

- [Snowball Edge의 Amazon EKS Anywhere에 대한 초기 설정](#)
- [Snowball Edge 디바이스에서 Amazon EKS Anywhere를 자동으로 구성 및 실행](#)
- [Snowball Edge 디바이스에서 Amazon EKS Anywhere를 수동으로 구성 및 실행](#)

## Snowball Edge의 Amazon EKS Anywhere에 대한 초기 설정

각 Snowball Edge 디바이스를 로컬 네트워크에 연결하고, Snowball Edge 클라이언트를 다운로드하고, 자격 증명을 받고, 디바이스를 잠금 해제하여 초기 설정을 수행합니다.

초기 설정을 수행합니다.

1. Snowball Edge 클라이언트를 다운로드 및 설치합니다. 자세한 내용은 [Snowball Edge 클라이언트 다운로드 및 설치](#) 섹션을 참조하세요.
2. 디바이스를 로컬 네트워크에 연결 자세한 내용은 [로컬 네트워크에 Snowball Edge 연결](#) 섹션을 참조하세요.
3. 자격 증명을 받아 디바이스 잠금을 해제하세요. 자세한 내용은 [Snowball Edge에 액세스하기 위한 자격 증명 가져오기](#) 섹션을 참조하세요.
4. 디바이스의 잠금을 해제합니다. 자세한 내용은 [Snowball Edge 잠금 해제](#) 섹션을 참조하세요. 또한 디바이스를 수동으로 잠금 해제하는 대신 스크립트 도구를 사용할 수도 있습니다. [디바이스 잠금 해제](#)를 참조하세요.

## Snowball Edge 디바이스에서 Amazon EKS Anywhere를 자동으로 구성 및 실행

샘플 스크립트 도구를 사용하여 환경을 설정하고 Amazon EKS Anywhere 관리자 인스턴스를 실행하거나 수동으로 실행할 수 있습니다. 스크립트 도구를 사용하려면 [Amazon EKS Anywhere의 디바이스 잠금 해제 및 환경 설정](#)을 참조하세요. 환경을 설정하고 Amazon EKS Anywhere 관리자 인스턴스를 실행한 후, 네트워크 연결이 끊긴 상태에서도 Snowball Edge 디바이스에서 작동하도록 Amazon EKS Anywhere를 구성해야 하는 경우 [연결 해제된 작업을 위해 AWS Snow에서 Amazon EKS Anywhere 구성](#)을 참조하세요. 그렇지 않으면 [Snowball Edge 디바이스에서 클러스터 생성 및 유지 관리](#) 섹션을 참조하세요.

환경을 수동으로 설정하고 Amazon EKS Anywhere 관리자 인스턴스를 실행하려면 [Snowball Edge 디바이스에서 Amazon EKS Anywhere를 수동으로 구성 및 실행](#)을 참조하세요.

## Snowball Edge 디바이스에서 Amazon EKS Anywhere를 수동으로 구성 및 실행

Snowball Edge 디바이스에서 Amazon EKS Anywhere를 구성하기 전에 Snowball Edge 클라이언트에 대한 프로필을 설정합니다. 자세한 내용은 [Snowball Edge 클라이언트 구성 및 사용](#) 단원을 참조하십시오.

### 주제

- [Amazon EKS Anywhere IAM 로컬 사용자 생성](#)
- [\(선택 사항\) Snowball Edge에서 Secure Shell 키 생성 및 가져오기](#)
- [Snowball Edge에서 Amazon EKS Anywhere 관리자 인스턴스를 실행하고 자격 증명 및 인증서 파일을 해당 인스턴스로 전송](#)

### Amazon EKS Anywhere IAM 로컬 사용자 생성

최상의 보안 사례를 위해 Snowball Edge 디바이스에서 Amazon EKS Anywhere용 로컬 IAM 사용자를 생성하세요. 이를 위해 다음 절차를 사용할 수 있습니다.

#### Note

사용하는 각 Snowball Edge 디바이스에 대해 이 작업을 수행하세요.

## Snowball Edge에서 로컬 사용자 생성

`create-user` 명령을 사용하여 Amazon EKS Anywhere IAM 사용자를 생성합니다.

```
aws iam create-user --user-name user-name --endpoint http://snowball-ip:6078 --
profile profile-name
{
  "User": {
    "Path": "/",
    "UserName": "eks-a-user",
    "UserId": "AIDACKCEVSQ6C2EXAMPLE",
    "Arn": "arn:aws:iam::123456789012:user/eks-a-user",
    "CreateDate": "2022-04-06T00:13:35.665000+00:00"
  }
}
```

## Snowball Edge에서 로컬 사용자에 대한 정책 생성

정책 문서를 생성하고, 이를 사용하여 IAM 정책을 생성하고, 해당 정책을 Amazon EKS Anywhere 로컬 사용자에게 연결합니다.

정책 문서를 생성하여 Amazon EKS Anywhere 로컬 사용자에게 첨부하려면

1. 정책 문서를 생성하여 컴퓨터에 저장합니다. 아래 정책을 문서에 복사하세요.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "snowball-device:DescribeDevice",
        "snowball-device:CreateDirectNetworkInterface",
        "snowball-device>DeleteDirectNetworkInterface",
        "snowball-device:DescribeDirectNetworkInterfaces",
        "snowball-device:DescribeDeviceSoftware"
      ]
    }
  ],
}
```

```

    "Resource": ["*"]
  },
  {
    "Effect": "Allow",
    "Action": [
      "ec2:RunInstances",
      "ec2:DescribeInstances",
      "ec2:TerminateInstances",
      "ec2:ImportKeyPair",
      "ec2:DescribeKeyPairs",
      "ec2:DescribeInstanceTypes",
      "ec2:DescribeImages",
      "ec2>DeleteTags"
    ],
    "Resource": ["*"]
  }
]
}

```

2. `create-policy` 명령을 사용하여 정책 문서를 기반으로 IAM 정책을 생성합니다. `--policy-document` 파라미터 값은 정책 파일의 절대 경로를 사용해야 합니다. 예: `file:///home/user/policy-name.json`

```

aws iam create-policy --policy-name policy-name --policy-document file:///home/user/policy-name.json --endpoint http://snowball-ip:6078 --profile profile-name
{
  "Policy": {
    "PolicyName": "policy-name",
    "PolicyId":
"ANPACEMGEZDGNBVG3TQ0JQGEZAAAABP76TE5MKAAAABCC0TR2IJ43NBTJRZBU",
    "Arn": "arn:aws:iam::123456789012:policy/policy-name",
    "Path": "/",
    "DefaultVersionId": "v1",
    "AttachmentCount": 0,
    "IsAttachable": true,
    "CreateDate": "2022-04-06T04:46:56.907000+00:00",
    "UpdateDate": "2022-04-06T04:46:56.907000+00:00"
  }
}

```

3. `attach-user-policy` 명령을 사용하여 Amazon EKS Anywhere 로컬 사용자에게 IAM 정책을 연결합니다.

```
aws iam attach-user-policy --policy-arn policy-arn --user-name user-name --endpoint
http://snowball-ip:6078 --profile profile-name
```

## Snowball Edge에서 액세스 키 및 자격 증명 파일 생성

Amazon EKS Anywhere IAM 로컬 사용자를 위한 액세스 키를 생성합니다. 그런 다음 자격 증명 파일을 생성하고 로컬 사용자에게 대해 생성된 `AccessKeyId` 및 `SecretAccessKey` 값을 이 파일에 포함하세요. 자격 증명 파일은 나중에 Amazon EKS Anywhere 관리자 인스턴스에서 사용됩니다.

1. `create-access-key` 명령을 사용하여 Amazon EKS Anywhere 로컬 사용자를 위한 액세스 키를 생성합니다.

```
aws iam create-access-key --user-name user-name --endpoint http://snowball-ip:6078
--profile profile-name
{
  "AccessKey": {
    "UserName": "eks-a-user",
    "AccessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "Status": "Active",
    "SecretAccessKey": "RTT/wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY",
    "CreateDate": "2022-04-06T04:23:46.139000+00:00"
  }
}
```

2. 자격 증명 파일을 생성합니다. 여기에 `AccessKeyId` 및 `SecretAccessKey` 값을 다음 형식으로 저장합니다.

```
[snowball-ip]
aws_access_key_id = ABCDEFGHIJKLMNOPQR2T
aws_secret_access_key = AfSD7sYz/TBZtzkReB16PuuISzJ2WtNkeePw+nNzJ
region = snow
```

**Note**

여러 Snowball Edge 디바이스를 사용하는 경우 파일의 자격 증명 순서는 중요하지 않지만 모든 디바이스의 자격 증명이 하나의 파일에 있어야 합니다.

## Snowball Edge에서 관리자 인스턴스에 대한 인증서 파일 생성

Amazon EKS Anywhere 관리자 인스턴스를 실행하려면 Snowball Edge 디바이스의 인증서가 있어야 합니다. 인증서가 들어 있는 인증서 파일을 생성하여 나중에 Amazon EKS Anywhere 관리자 인스턴스에서 사용할 수 있도록 Snowball Edge 디바이스에 액세스할 수 있습니다.

## 인증서 파일을 생성하는 방법

1. `list-certificates` 명령을 사용하여 사용하려는 각 Snowball Edge 디바이스의 인증서를 가져올 수 있습니다.

```
PATH_TO_Snowball_Edge_CLIENT/bin/snowballEdge list-certificates --endpoint
https://snowball-ip --manifest-file path-to-manifest-file --unlock-code unlock-
code
{
  "Certificates" : [ {
    "CertificateArn" : "arn:aws:snowball-device:::certificate/xxx",
    "SubjectAlternativeNames" : [ "ID:JID-xxx" ]
  } ]
}
```

2. `CertificateArn`의 값을 `get-certificate` 명령의 `--certificate-arn` 파라미터 값으로 사용합니다.

```
PATH_TO_Snowball_Edge_CLIENT/bin/snowballEdge get-certificate --certificate-arn ARN
--endpoint https://snowball-ip --manifest-file path-to-manifest-file --unlock-
code unlock-code
```

3. 디바이스 인증서 파일을 생성하는 방법 `get-certificate`의 출력을 인증서 파일에 넣습니다. 다음은 출력을 저장하는 방법의 예시입니다.

**Note**

여러 Snowball Edge 디바이스를 사용하는 경우 파일의 자격 증명 순서는 중요하지 않지만 모든 디바이스의 자격 증명이 하나의 파일에 있어야 합니다.

```
-----BEGIN CERTIFICATE-----
ZWtzYSBzbm93IHRlc3QgY2VydG1maWNhdGUgZWtzYSBzbm93IHRlc3QgY2VydG1m
aWNhdGV1a3NhIHhNub3cgdGVzdCBjZXJ0aWZpY2F0ZWVrc2Egc25vdyB0ZXN0IGN1
cnRpZm1jYXRlZWtzYSBzbm93IHRlc3QgY2VydG1maWNhdGV1a3NhIHhNub3cgdGVz
dCBjZXJ0aWZpY2F0ZQMIIDXCcAkSgAwIBAgIJAISM0nTVmbj+MA0GCSqGSIb3DQ
...
-----END CERTIFICATE-----
```

4. 모든 Snowball Edge 디바이스에서 Amazon EKS Anywhere용 IAM 로컬 사용자를 생성하려면 [Amazon EKS Anywhere IAM 로컬 사용자 생성](#) 단계를 반복합니다.

## (선택 사항) Snowball Edge에서 Secure Shell 키 생성 및 가져오기

이 선택적 절차를 사용하여 모든 Amazon EKS Anywhere 노드 인스턴스에 액세스할 수 있는 보안 셸 (SSH) 키를 생성하고 퍼블릭 키를 모든 Snowball Edge 디바이스로 가져올 수 있습니다. 이 키 파일을 잘 보관하고 보호하세요.

이 절차를 건너뛰면 Amazon EKS Anywhere는 필요할 때 SSH 키를 자동으로 생성하고 가져옵니다. 이 키는 `${PWD}/${CLUSTER_NAME}/eks-a-id_rsa`의 관리자 인스턴스에 저장됩니다.

SSH 키를 생성하여 Amazon EKS Anywhere 인스턴스로 가져옵니다.

1. `ssh-keygen` 명령을 사용하여 SSH 키를 생성합니다.

```
ssh-keygen -t rsa -C "key-name" -f path-to-key-file
```

2. `import-key-pair` 명령을 사용하여 컴퓨터에서 Snowball Edge 디바이스로 키를 가져옵니다.

**Note**

키를 모든 디바이스로 가져올 때는 `key-name` 파라미터 값이 같아야 합니다.

```
aws ec2 import-key-pair --key-name key-name --public-key-material fileb:///path/to/key-file --endpoint http://snowball-ip:8008 --profile profile-name
{
  "KeyFingerprint": "5b:0c:fd:e1:a0:69:05:4c:aa:43:f3:3b:3e:04:7f:51",
  "KeyName": "default",
  "KeyPairId": "s.key-85edb5d820c92a6f8"
}
```

## Snowball Edge에서 Amazon EKS Anywhere 관리자 인스턴스를 실행하고 자격 증명 및 인증서 파일을 해당 인스턴스로 전송

### Snowball Edge에서 Amazon EKS Anywhere 관리자 인스턴스 실행

이 절차에 따라 Amazon EKS Anywhere 관리자 인스턴스를 수동으로 실행하고, 관리자 인스턴스의 가상 네트워크 인터페이스(VNI)를 구성하고, 인스턴스 상태를 확인하고, SSH 키를 생성하고, 이 키를 사용하여 관리자 인스턴스에 연결합니다. 샘플 스크립트 도구를 사용하여 Amazon EKS Anywhere 관리자 인스턴스를 생성하고 이 인스턴스로 자격 증명 및 인증서 파일을 전송하는 작업을 자동화할 수 있습니다. [Amazon EKS Anywhere 관리자 인스턴스 생성](#)을 참조하세요. 스크립트 도구가 완료되면 [Snowball Edge 디바이스에서 클러스터 생성 및 유지 관리](#)를 참조하여 인스턴스에 ssh로 연결하고 클러스터를 생성할 수 있습니다. Amazon EKS Anywhere 인스턴스를 수동으로 설정하려면 다음 단계를 사용하세요.

**Note**

클러스터를 프로비저닝하는 데 두 개 이상의 Snowball Edge 디바이스를 사용하는 경우, 모든 Snowball Edge 디바이스에서 Amazon EKS Anywhere 관리자 인스턴스를 시작할 수 있습니다.



## Amazon EKS Anywhere 관리자 인스턴스를 실행하려면

1. `create-key-pair` 명령을 사용하여 Amazon EKS Anywhere 관리자 인스턴스용 SSH 키를 생성합니다. 이 명령은 키를 `$PWD/key-file-name`에 저장합니다.

```
aws ec2 create-key-pair --key-name key-name --query 'KeyMaterial' --output text --  
endpoint http://snowball ip:8008 > key-file-name --profile profile-name
```

2. `describe-images` 명령을 사용하여 출력에서 `eks-anywhere-admin`로 시작하는 이미지 이름을 찾을 수 있습니다.

```
aws ec2 describe-images --endpoint http://snowball-ip:8008 --profile profile-name
```

3. `run-instance` 명령을 사용하여 Amazon EKS Anywhere 관리 이미지로 `eks-a` 관리자 인스턴스를 시작합니다.

```
aws ec2 run-instances --image-id eks-a-admin-image-id --key-name key-name --  
instance-type sbe-c.xlarge --endpoint http://snowball-ip:8008 --profile profile-  
name
```

4. `describe-instances` 명령을 사용하여 Amazon EKS Anywhere 인스턴스의 상태를 확인합니다. 명령에서 인스턴스 상태가 `running`으로 표시될 때까지 기다린 다음 계속하세요.

```
aws ec2 describe-instances --instance-id instance-id --endpoint http://snowball-  
ip:8008 --profile profile-name
```

5. `describe-device` 명령 출력에서 네트워크에 연결된 물리적 네트워크 인터페이스의 `PhysicalNetworkInterfaceId` 값을 기록해 둡니다. 이를 사용하여 VNI를 생성합니다.

```
PATH_TO_Snowball_Edge_CLI/bin/snowballEdge describe-device --endpoint  
https://snowball-ip --manifest-file path-to-manifest-file --unlock-code unlock-  
code
```

6. Amazon EKS Anywhere 관리자 인스턴스용 VNI를 생성합니다. `PhysicalNetworkInterfaceId`의 값을 `physical-network-interface-id` 파라미터의 값으로 사용합니다.

```
PATH_TO_Snowball_Edge_CLIENT/bin/snowballEdge create-virtual-network-interface
--ip-address-assignment dhcp --physical-network-interface-id PNI --endpoint
https://snowball-ip --manifest-file path-to-manifest-file --unlock-code unlock-
code
```

7. `IpAddress`의 값을 `associate-address` 명령의 `public-ip` 파라미터 값으로 사용하여 퍼블릭 주소를 Amazon EKS Anywhere 관리자 인스턴스에 연결합니다.

```
aws ec2 associate-address --instance-id instance-id --public-ip VNI-IP --endpoint
http://snowball-ip:8008 --profile profile-name
```

8. SSH를 통해 Amazon EKS Anywhere 관리자 인스턴스에 연결합니다.

```
ssh -i path-to-key ec2-user@VNI-IP
```

인증서 및 자격 증명 파일을 Snowball Edge의 관리자 인스턴스로 전송

Amazon EKS Anywhere 관리자 인스턴스가 실행된 후 Snowball Edge 디바이스의 자격 증명과 인증서를 관리자 인스턴스로 전송합니다. 자격 증명과 인증서 파일을 [Snowball Edge에서 액세스 키 및 자격 증명 파일 생성](#) 및 [Snowball Edge에서 관리자 인스턴스에 대한 인증서 파일 생성](#)에 저장한 디렉터리에서 다음 명령을 실행합니다.

```
scp -i path-to-key path-to-credentials-file path-to-certificates-file ec2-user@eks-
admin-instance-ip:~
```

Amazon EKS Anywhere 관리자 인스턴스에서 파일의 내용을 확인합니다. 다음은 자격 증명 및 인증서 파일의 예입니다.

```
[192.168.1.1]
```

```
aws_access_key_id = EMGEZDGNBVGy3TQ0JQGEZB5ULEAAIWHWUJDXEXAMPLE
aws_secret_access_key = AUHpqj00GZQHEyXDbN0neLN1fR0gEXAMPLE
region = snow

[192.168.1.2]
aws_access_key_id = EMGEZDGNBVGy3TQ0JQGEZG507F3FJUcMYRMI4KPIEXAMPLE
aws_secret_access_key = kY4C18+RJAwq/bu28Y8fUJepwqhDEXAMPLE
region = snow
```

```
-----BEGIN CERTIFICATE-----
ZWtzYSBzbm93IHRlc3QgY2VydGlmawNhdGUgZWtzYSBzbm93IHRlc3QgY2VydGlm
aWNhdGVla3NhIHhNub3cgcGVzdCBjZXJ0aWZpY2F0ZWVrc2Egc25vdyB0ZXN0IGN1
cnRpZmljYXRlZWtzYSBzbm93IHRlc3QgY2VydGlmawNhdGVla3NhIHhNub3cgcGVz
dCBjZXJ0aWZpY2F0ZQMIIDXCcAkSgAwIBAgIJJAISM0nTVmbj+MA0GCSqGSIb3DQ
...
-----END CERTIFICATE-----

-----BEGIN CERTIFICATE-----
KJ0FP12PAYPEjxr81/PoCXfZeARBzN9WLUH5yz1ta+sYUJouzHzWuLJYA1xqcCPY
mhV1kRsN4hVd1BNRnCCpRF766yjdJeibKVzXQxoXoZBjr0kuGwqRy3d3ndjK77h4
OR5Fv9mjGf7CjcaSjk/4iwmZvRSaQacb0YG5GvEb4mfUAuVtuFoMeYfnAgMBAAGj
azBpMAwGA1UdEwQFMAMBAf8wHQYDVR00BBYEFL/bRcnBRuSM5+FcYFa8HfIBomdF
...
-----END CERTIFICATE-----
```

## 연결 해제된 작업을 위해 AWS Snow에서 Amazon EKS Anywhere 구성

네트워크에 연결된 상태에서 Snowball Edge 디바이스에서 Amazon EKS Anywhere의 추가 구성을 완료하면 외부 네트워크 연결이 없는 환경에서 Amazon EKS Anywhere를 실행할 준비를 할 수 있습니다.

로컬 프라이빗 Kubernetes 레지스트리와 연결이 끊긴 상태에서 사용하도록 Amazon EKS Anywhere를 구성하려면 EKS Anywhere 설명서의 [레지스트리 미리 구성](#)을 참조하세요.

Harbor 프라이빗 레지스트리 AMI를 생성한 경우 이 섹션의 절차를 따르세요.

### 주제

- [Snowball Edge 디바이스에서 Harbor 레지스트리를 구성합니다.](#)

- [Snowball Edge의 Amazon EKS Anywhere 관리자 인스턴스에서 Harbor 레지스트리 사용](#)

Snowball Edge 디바이스에서 Harbor 레지스트리를 구성합니다.

[Snowball Edge 디바이스에서 Harbor 구성](#)을 참조하세요.

## Snowball Edge의 Amazon EKS Anywhere 관리자 인스턴스에서 Harbor 레지스트리 사용

[Snowball Edge 디바이스의 로컬 Harbor 레지스트리로 Amazon EKS Anywhere 컨테이너 이미지 가져오기](#)를 참조하세요.

## Snowball Edge 디바이스에서 클러스터 생성 및 유지 관리

### Snowball Edge에서 클러스터를 생성하는 모범 사례

Amazon EKS Anywhere 클러스터를 생성하려면 [Snow 클러스터 생성](#)을 참조하세요.

Snowball Edge 디바이스에서 Amazon EKS Anywhere 클러스터를 생성할 때는 다음 모범 사례를 염두에 두세요.

- 정적 IP 주소 범위에서 클러스터를 생성하기 전에 Snowball Edge 디바이스에 동일한 IP 주소 범위를 사용하는 다른 클러스터가 없는지 확인합니다.
- Snowball Edge 디바이스에서 DHCP 주소 지정을 사용하여 클러스터를 생성하려면 클러스터에 사용하는 모든 고정 IP 주소 범위가 DHCP 풀 서브넷에 있지 않은지 확인합니다.
- 둘 이상의 클러스터를 생성할 때는 클러스터가 성공적으로 프로비저닝되고 실행될 때까지 기다린 후 다른 클러스터를 생성합니다.

### Snowball Edge에서 클러스터 업그레이드

Amazon EKS Anywhere 관리자 AMI 또는 EKS Distro AMI를 업그레이드하려면 contact AWS Support. 지원 는 업그레이드된 AMI가 포함된 Snowball Edge 업데이트를 제공합니다. 그런 다음 Snowball Edge 업데이트를 다운로드하여 설치합니다. [Snowball Edge 디바이스에 업데이트 다운로드 및 Snowball Edge 디바이스에 업데이트 설치](#)를 참조하세요.

Amazon EKS Anywhere AMI를 업그레이드한 후에는 새 Amazon EKS Anywhere 관리자 인스턴스를 시작해야 합니다. [Snowball Edge에서 Amazon EKS Anywhere 관리자 인스턴스 실행](#) 섹션을 참조하세요.

요. 그런 다음 이전 관리자 인스턴스의 키 파일, 클러스터 폴더, 자격 증명 및 인증서를 업그레이드된 인스턴스로 복사합니다. 이러한 항목은 클러스터의 이름이 지정된 폴더에 있습니다.

## Snowball Edge에서 클러스터 리소스 정리

Snowball Edge 디바이스에서 여러 클러스터를 생성한 후 올바르게 삭제하지 않거나 클러스터에 문제가 발생하여 클러스터가 재개 후 대체 노드를 생성하는 경우 리소스 누락이 발생합니다. 샘플 스크립트 도구를 수정하거나 사용하여 Amazon EKS Anywhere 관리자 인스턴스와 Snowball Edge 디바이스를 정리할 수 있습니다. [Amazon EKS Anywhere on AWS Snow 정리 도구를](#) 참조하세요.

# Snowball Edge에서 로컬로 IAM 사용

AWS Identity and Access Management (IAM)를 사용하면 AWS Snowball Edge 디바이스에서 실행되는 AWS 리소스에 대한 액세스를 안전하게 제어할 수 있습니다. IAM을 사용하여 리소스를 사용하도록 인증(로그인) 및 권한 부여(권한 있음)된 대상을 제어합니다.

IAM은 디바이스에서 로컬로 지원됩니다. 로컬 IAM 서비스를 사용하여 새 사용자를 생성하고 사용자에게 IAM 정책을 연결할 수 있습니다. 이러한 정책을 사용하여 할당된 작업을 수행하는 데 필요한 액세스를 허용할 수 있습니다. 예를 들어 사용자에게 데이터를 전송하는 권한을 부여하지만 새 Amazon EC2 호환 인스턴스를 생성하는 권한은 제한할 수 있습니다.

또한 디바이스에서 AWS Security Token Service (AWS STS)를 사용하여 로컬 세션 기반 자격 증명을 생성할 수 있습니다. IAM 서비스에 대한 자세한 내용은 IAM 사용 설명서의 [시작하기](#) 섹션을 참조하세요.

디바이스의 루트 자격 증명은 비활성화할 수 없으며 계정 내의 정책을 사용하여 AWS 계정 루트 사용자에 대한 액세스를 명시적으로 거부할 수 없습니다. 루트 사용자 액세스 키를 보호하고 디바이스와의 일상적인 상호 작용을 위한 사용자 보안 인증 정보를 생성하는 것이 좋습니다.

## Important

이 섹션의 설명서는 AWS Snowball Edge 디바이스에서 로컬로 IAM을 사용하는 데 적용됩니다. 에서 IAM을 사용하는 방법에 대한 자세한 내용은 단원을 [AWS 클라우드참조하십시오의 자격 증명 및 액세스 관리 AWS Snowball Edge](#).

AWS 서비스가 Snowball Edge에서 제대로 작동하려면 서비스의 포트를 허용해야 합니다. 자세한 내용은 [Snowball Edge의 AWS 서비스에 대한 포트 요구 사항](#)을 참조하세요.

## 주제

- [Snowball Edge에서 AWS CLI 및 API 작업 사용](#)
- [Snowball Edge에서 지원되는 IAM AWS CLI 명령 목록](#)
- [Snowball Edge의 IAM 정책 예제](#)
- [Snowball Edge의 TrustPolicy 예제](#)

## Snowball Edge에서 AWS CLI 및 API 작업 사용

AWS CLI 또는 API 작업을 사용하여 Snowball Edge에서 IAM, AWS STS, Amazon S3 및 Amazon EC2 명령을 실행할 때는 `""region`로 지정해야 합니다. 다음 예제와 같이 명령 자체 내에서 `aws configure` 또는 `aws configure`를 사용하여 작업을 수행할 수 있습니다.

```
aws configure --profile abc
AWS Access Key ID [None]: AKIAIOSFODNN7EXAMPLE
AWS Secret Access Key [None]: 1234567
Default region name [None]: snow
Default output format [None]: json
```

Or

```
aws iam list-users --endpoint http://192.0.2.0:6078 --region snow --profile
snowballEdge
```

### Note

에서 AWS Snowball Edge 로컬로 사용되는 액세스 키 ID 및 액세스 보안 키는의 키와 교환할 수 없습니다 AWS 클라우드.

## Snowball Edge에서 지원되는 IAM AWS CLI 명령 목록

다음은 Snowball Edge 디바이스에서 지원되는 IAM에 대한 AWS CLI 명령 및 옵션의 하위 집합에 대한 설명입니다. 아래에 나열되지 않은 명령이나 옵션은 지원되지 않습니다. 명령에 지원되지 않는 파라미터는 별도로 명시되어 있습니다.

- [attach-role-policy](#) - 지정된 관리형 정책을 지정된 IAM 역할에 연결합니다.
- [attach-user-policy](#) - 지정된 관리형 정책을 지정된 사용자에게 연결합니다.
- [create-access-key](#) - 지정된 사용자에게 대한 새 로컬 IAM 보안 액세스 키와 해당 AWS 액세스 키 ID를 생성합니다.
- [create-policy](#) - 디바이스에 대한 새 IAM 관리형 정책을 생성합니다.

- [create-role](#) - 디바이스에 대한 새 로컬 IAM 역할을 생성합니다. 다음 파라미터는 지원되지 않습니다.
  - Tags
  - PermissionsBoundary
- [create-user](#) - 디바이스에 대한 새 로컬 IAM 사용자를 생성합니다. 다음 파라미터는 지원되지 않습니다.
  - Tags
  - PermissionsBoundary
- [delete-access-key](#) - 지정된 사용자의 새 로컬 IAM 보안 액세스 키와 해당 AWS 액세스 키 ID를 삭제합니다.
- [delete-policy](#) - 지정된 관리형 정책을 삭제합니다.
- [delete-role](#) - 지정된 역할을 삭제합니다.
- [delete-user](#) - 지정된 사용자를 삭제합니다.
- [detach-role-policy](#) - 지정된 역할에서 지정된 관리형 정책을 제거합니다.
- [detach-user-policy](#) - 지정된 사용자에서 지정된 관리형 정책을 제거합니다.
- [get-policy](#) - 정책의 기본 버전과 정책이 연결된 로컬 IAM 사용자, 그룹 및 역할의 총 수를 포함하여 지정된 관리형 정책에 대한 정보를 검색합니다.
- [get-policy-version](#) - 정책 문서를 포함하여 지정된 관리형 정책의 지정된 버전에 대한 정보를 검색합니다.
- [get-role](#) - 역할의 경로, GUID, ARN 및 역할을 수임할 수 있는 권한을 부여하는 역할의 신뢰 정책을 포함하여 지정된 역할에 대한 정보를 검색합니다.
- [get-user](#) - 사용자의 생성 날짜, 경로, 고유 ID, ARN을 포함하여 지정된 IAM 사용자에 대한 정보를 검색합니다.
- [list-access-keys](#) - 지정된 IAM 사용자와 연결된 액세스 키 ID에 대한 정보를 반환합니다.
- [list-attached-role-policies](#) - 지정된 IAM 역할에 연결된 모든 관리형 정책을 나열합니다.
- [list-attached-user-policies](#) - 지정된 IAM 사용자에게 연결된 모든 관리형 정책을 나열합니다.
- [list-entities-for-policy](#) - 지정된 관리형 정책이 연결된 모든 로컬 IAM 사용자, 그룹 및 역할을 나열합니다.
  - --EntityFilter: user 및 role 값만 지원됩니다.
- [list-policies](#) - 로컬 AWS 계정에서 사용 가능한 모든 관리형 정책을 나열합니다. 다음 파라미터는 지원되지 않습니다.
  - --PolicyUsageFilter
- [list-roles](#) - 지정된 경로 접두사가 있는 로컬 IAM 역할을 나열합니다.



- [list-users](#) - 지정된 경로 접두사가 있는 IAM 사용자를 나열합니다.
- [update-access-key](#) - 지정된 액세스 키의 상태를 활성에서 비활성으로 또는 이와 반대로 변경합니다.
- [update-assume-role-policy](#) - IAM 개체에 역할을 수임할 권한을 부여하는 정책을 업데이트합니다.
- [update-role](#) - 역할의 설명 또는 최대 세션 기간 설정을 업데이트합니다.
- [update-user](#) - 지정된 IAM 사용자의 이름 및/또는 경로를 업데이트합니다.

## Snowball Edge에서 지원되는 IAM API 작업

다음은 IAM API 참조의 설명 링크와 함께 Snowball Edge에서 사용할 수 있는 IAM API 작업입니다.

- [AttachRolePolicy](#) - 지정된 관리형 정책을 지정된 IAM 역할에 연결합니다.
- [AttachUserPolicy](#) - 지정된 관리형 정책을 지정된 사용자에게 연결합니다.
- [CreateAccessKey](#) - 지정된 사용자에게 대한 새 로컬 IAM 보안 액세스 키와 해당 AWS 액세스 키 ID를 생성합니다.
- [CreatePolicy](#) - 디바이스에 대한 새 IAM 관리형 정책을 생성합니다.
- [CreateRole](#) - 디바이스에 대한 새 로컬 IAM 역할을 생성합니다.
- [CreateUser](#) - 디바이스에 대한 새 로컬 IAM 사용자를 생성합니다.

다음 파라미터는 지원되지 않습니다.


- Tags
- PermissionsBoundary
- [DeleteAccessKey](#) - 지정된 액세스 키를 삭제합니다.
- [DeletePolicy](#) - 지정된 관리형 정책을 삭제합니다.
- [DeleteRole](#) - 지정된 역할을 삭제합니다.
- [DeleteUser](#) - 지정된 사용자를 삭제합니다.
- [DetachRolePolicy](#) - 지정된 역할에서 지정된 관리형 정책을 제거합니다.
- [DetachUserPolicy](#) - 지정된 사용자에서 지정된 관리형 정책을 제거합니다.
- [GetPolicy](#) - 정책의 기본 버전과 정책이 연결된 로컬 IAM 사용자, 그룹 및 역할의 총 수를 포함하여 지정된 관리형 정책에 대한 정보를 검색합니다.
- [GetPolicyVersion](#) - 정책 문서를 포함하여 지정된 관리형 정책의 지정된 버전에 대한 정보를 검색합니다.

- [GetRole](#) - 역할의 경로, GUID, ARN 및 역할을 수임할 수 있는 권한을 부여하는 역할의 신뢰 정책을 포함하여 지정된 역할에 대한 정보를 검색합니다.
- [GetUser](#) - 사용자의 생성 날짜, 경로, 고유 ID, ARN을 포함하여 지정된 IAM 사용자에 대한 정보를 검색합니다.
- [ListAccessKeys](#) - 지정된 IAM 사용자와 연결된 액세스 키 ID에 대한 정보를 반환합니다.
- [ListAttachedRolePolicies](#) - 지정된 IAM 역할에 연결된 모든 관리형 정책을 나열합니다.
- [ListAttachedUserPolicies](#) - 지정된 IAM 사용자에게 연결된 모든 관리형 정책을 나열합니다.
- [ListEntitiesForPolicy](#) - 사용자의 생성 날짜, 경로, 고유 ID, ARN을 포함하여 지정된 IAM 사용자에 대한 정보를 검색합니다.
  - --EntityFilter: user 및 role 값만 지원됩니다.
- [ListPolicies](#) - 로컬 AWS 계정에서 사용 가능한 모든 관리형 정책을 나열합니다. 다음 파라미터는 지원되지 않습니다.
  - --PolicyUsageFilter
- [ListRoles](#) - 지정된 경로 접두사가 있는 로컬 IAM 역할을 나열합니다.
- [ListUsers](#) - 지정된 경로 접두사가 있는 IAM 사용자를 나열합니다.
- [UpdateAccessKey](#) - 지정된 액세스 키의 상태를 활성에서 비활성으로 또는 이와 반대로 변경합니다.
- [UpdateAssumeRolePolicy](#) - IAM 개체에 역할을 수임할 권한을 부여하는 정책을 업데이트합니다.
- [UpdateRole](#) - 역할의 설명 또는 최대 세션 기간 설정을 업데이트합니다.
- [UpdateUser](#) - 지정된 IAM 사용자의 이름 및/또는 경로를 업데이트합니다.

## Snowball Edge에서 지원되는 IAM 정책 버전 및 문법

다음은 로컬 IAM 지원 버전 2012-10-17의 IAM 정책과 일부 정책 문법입니다.

정책 유형	지원되는 문법
자격 증명 기반 정책(사용자/역할 정책)	"Effect", "Action" 및 "Resource"

 **Note**  
 로컬 IAM 정책은 "Condition ", "NotAction ", "NotResource " 및 "Principal "을 지원합니다.

정책 유형	지원되는 문법
리소스 기반 정책(역할 신뢰 정책)	"Effect", "Action" 및 "Principal "

**Note**  
보안 주체의 경우 AWS 계정 ID 또는 보안 주체 ID만 허용됩니다.

## Snowball Edge의 IAM 정책 예제

### Note

AWS Identity and Access Management (IAM) 사용자는 [AWS OpsHub for Snow Family 애플리케이션](#)을 사용하여 Snowball Edge를 관리할 수 있는 "snowballdevice:\*" 권한이 필요합니다.

다음은 Snowball Edge 디바이스에 권한을 부여하는 정책의 예시입니다.

## IAM API를 통해 Snowball Edge에서 샘플 사용자에게 GetUser 호출 허용

다음 정책을 사용하여 IAM API를 통한 샘플 사용자에게 대한 GetUser 직접 호출을 허용합니다.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": "iam:GetUser",
      "Resource": "arn:aws:iam::user/example-user"
    }
  ]
}
```

## Snowball Edge에서 Amazon S3 API에 대한 전체 액세스 허용

다음 정책을 사용하여 Amazon S3 API에 대한 모든 액세스를 허용합니다.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": "s3:*",
      "Resource": "*"
    }
  ]
}
```

## Snowball Edge의 Amazon S3 버킷에 대한 읽기 및 쓰기 액세스 허용

다음 정책은 특정 버킷에 대한 읽기 및 쓰기 액세스를 허용합니다.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ListObjectsInBucket",
      "Effect": "Allow",
      "Action": "s3:ListBucket",
      "Resource": "arn:aws:s3:::bucket-name"
    },
    {
      "Sid": "AllObjectActions",
      "Effect": "Allow",
      "Action": "s3:*Object",
      "Resource": "arn:aws:s3:::bucket-name/*"
    }
  ]
}
```

```
]
}
```

## Snowball Edge의 Amazon S3 버킷에 대한 목록, 가져오기 및 액세스 허용

다음 정책을 사용하여 특정 S3 버킷에 대한 나열, 가져오기 및 배치 액세스를 허용합니다.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:PutObject",
        "s3:GetObject",
        "s3:List*"
      ],
      "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*"
    }
  ]
}
```

## Snowball Edge에서 Amazon EC2 API에 대한 전체 액세스 허용

다음 정책을 사용하여 Amazon EC2에 대한 모든 액세스를 허용합니다.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:*",
      "Resource": "*"
    }
  ]
}
```

```
]
}
```

## Snowball Edge에서 Amazon EC2-compatible 인스턴스를 시작하고 중지할 수 있는 액세스 허용

다음 정책을 사용하여 Amazon EC2 인스턴스를 시작 및 중지하는 액세스를 허용합니다.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:StartInstances",
        "ec2:StopInstances"
      ],
      "Resource": "*"
    }
  ]
}
```

## DescribeLaunchTemplates에 대한 호출을 거부하지만 Snowball Edge에서 DescribeImages에 대한 모든 호출 허용

다음 정책은 DescribeLaunchTemplates에 대한 호출을 거부하고 DescribeImages에 대한 모든 호출은 허용합니다.

JSON

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

```

        "Action": [
            "ec2:DescribeLaunchTemplates"
        ],
        "Resource": "*"
    },
    {
        "Effect": "Allow",
        "Action": [
            "ec2:DescribeImages"
        ],
        "Resource": "*"
    }
]
}

```

## Snowball Edge의 API 호출 정책

고객이 정의한 관리형 정책을 포함하여 Snow 디바이스에서 사용할 수 있는 모든 관리형 정책을 나열합니다. 자세한 내용은 [list-policies](#) 섹션을 참조하세요.

```

aws iam list-policies --endpoint http://ip-address:6078 --region snow --profile
snowballEdge
{
  "Policies": [
    {
      "PolicyName": "Administrator",
      "Description": "Root user admin policy for Account 123456789012",
      "CreateDate": "2020-03-04T17:44:59.412Z",
      "AttachmentCount": 1,
      "IsAttachable": true,
      "PolicyId": "policy-id",
      "DefaultVersionId": "v1",
      "Path": "/",
      "Arn": "arn:aws:iam::123456789012:policy/Administrator",
      "UpdateDate": "2020-03-04T19:10:45.620Z"
    }
  ]
}

```

## Snowball Edge의 TrustPolicy 예제

신뢰 정책은 일반적으로 액세스할 수 없는 AWS 리소스에 액세스하는 데 사용할 수 있는 임시 보안 자격 증명 세트를 반환합니다. 이러한 임시 보안 인증은 액세스 키 ID, 보안 액세스 키 및 보안 토큰으로 구성됩니다. 일반적으로 계정에서 교차 계정 액세스를 위해 AssumeRole을 사용합니다.

다음은 신뢰 정책의 예입니다. 신뢰 정책에 대한 자세한 내용은 AWS Security Token Service API Reference의 [AssumeRole](#) 섹션을 참조하세요.

### JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::AccountId:root" //You can use the Principal ID
instead of the account ID.
        ]
      },
      "Action": [
        "sts:AssumeRole"
      ]
    }
  ]
}
```



# Snowball Edge AWS Security Token Service 에서 사용

AWS Security Token Service (AWS STS)를 사용하면 IAM 사용자에게 대해 제한된 권한의 임시 자격 증명을 요청할 수 있습니다.

## ⚠ Important

AWS 서비스가 Snowball Edge에서 제대로 작동하려면 서비스에 대한 포트를 허용해야 합니다. 자세한 내용은 [Snowball Edge의 AWS 서비스에 대한 포트 요구 사항](#)을 참조하세요.

## 주제

- [Snowball Edge에서 AWS CLI 및 API 작업 사용](#)
- [Snowball Edge에서 지원되는 AWS STSAWS CLI 명령](#)
- [Snowball Edge에서 지원되는 AWS STS API 작업](#)

## Snowball Edge에서 AWS CLI 및 API 작업 사용

AWS CLI 또는 API 작업을 사용하여 Snowball Edge 디바이스에서 IAM AWS STS,, Amazon S3 및 Amazon EC2 명령을 실행할 때는 ""region로 지정해야 합니다snow. 다음 예제와 같이 명령 자체 내에서 AWS configure 또는를 사용하여이 작업을 수행할 수 있습니다.

```
aws configure --profile snowballEdge
AWS Access Key ID [None]: defgh
AWS Secret Access Key [None]: 1234567
Default region name [None]: snow
Default output format [None]: json
```

Or

```
aws iam list-users --endpoint http://192.0.2.0:6078 --region snow --profile
snowballEdge
```

**Note**

에서 로컬로 사용하는 액세스 키 ID 및 액세스 보안 키는의 키와 교환할 AWS Snowball Edge 수 없습니다 AWS 클라우드.

## Snowball Edge에서 지원되는 AWS STS AWS CLI 명령

[assume-role](#) 명령만 로컬에서 지원됩니다.

assume-role에는 다음 파라미터가 지원됩니다.

- role-arn
- role-session-name
- duration-seconds

## Snowball Edge에서 역할을 수임하는 명령 예제

역할을 수임하려면 다음 명령을 사용합니다.

```
aws sts assume-role --role-arn "arn:aws:iam::123456789012:role/example-role" --role-session-name AWSCLI-Session --endpoint http://snow-device-IP-address:7078
```

assume-role 명령 사용에 대한 자세한 내용은 [AWS CLI를 사용하여 IAM 역할을 위임하려면 어떻게 해야 합니까?](#)를 참조하세요.

사용에 대한 자세한 내용은 IAM 사용 설명서의 임시 보안 자격 증명 사용을 AWS STS참조하세요. <https://docs.aws.amazon.com/STS/latest/UsingSTS/>

## Snowball Edge에서 지원되는 AWS STS API 작업

[AssumeRole](#) API만 로컬에서 지원됩니다.

AssumeRole에는 다음 파라미터가 지원됩니다.

- RoleArn

- RoleSessionName
- DurationSeconds

### Example역할 수입

```
https://sts.amazonaws.com/  
?Version=2011-06-15  
&Action=AssumeRole  
&RoleSessionName=session-example  
&RoleArn=arn:aws:iam::123456789012:role/demo  
&DurationSeconds=3600
```

## Snowball Edge에서 퍼블릭 키 인증서 관리

퍼블릭 키 인증서를 제공하여 HTTPS 프로토콜을 통해 Snowball Edge 디바이스 또는 Snowball Edge 디바이스 클러스터에서 실행되는 AWS 서비스와 안전하게 상호 작용할 수 있습니다. HTTPS 프로토콜을 사용하여 IAM, Amazon EC2, S3 어댑터, Snowball Edge의 Amazon S3 호환 스토리지, Amazon EC2 Systems Manager 및 Snowball Edge 디바이스와 같은 AWS 서비스와 상호 작용할 수 있습니다. AWS STS도 있습니다. 디바이스 클러스터의 경우 단일 인증서가 필요하며 클러스터의 모든 디바이스에서 생성할 수 있습니다. Snowball Edge 디바이스에서 인증서를 생성하고 디바이스를 잠금 해제하면 Snowball Edge 클라이언트 명령을 사용하여 인증서를 나열하고, 가져오고, 삭제할 수 있습니다.

Snowball Edge 디바이스는 다음 이벤트가 발생할 때 인증서를 생성합니다.

- Snowball Edge 디바이스 또는 클러스터가 처음으로 잠금 해제됩니다.
- 인증서를 삭제한 후 Snowball Edge 디바이스 또는 클러스터가 잠금 해제됩니다(의 `delete-certificate` 명령 또는 인증서 갱신 사용 AWS OpsHub).
- 인증서가 만료되면 Snowball Edge 디바이스 또는 클러스터가 재부팅되고 잠금이 해제됩니다.

새 인증서가 생성될 때마다 이전 인증서는 더 이상 유효하지 않습니다. 인증서는 생성된 날로부터 1년 동안 유효합니다.

AWS OpsHub 를 사용하여 퍼블릭 키 인증서를 관리할 수도 있습니다. 자세한 내용은 이 설명서의 [Managing public key certificates using OpsHub](#) 섹션을 참조하세요.

### 주제

- [Snowball Edge에 인증서 나열](#)
- [Snowball Edge에서 인증서 가져오기](#)
- [Snowball Edge에서 인증서 삭제](#)

## Snowball Edge에 인증서 나열

`list-certificates` 명령을 사용하여 현재 인증서의 Amazon 리소스 이름(ARN)을 볼 수 있습니다.

```
snowballEdge list-certificates
```

## Example `list-certificates` 출력

```
{
  "Certificates" : [ {
    "CertificateArn" : "arn:aws:snowball-
device:::certificate/78EXAMPLE516EXAMPLEf538EXAMPLEa7",
    "SubjectAlternativeNames" : [ "192.0.2.0" ]
  } ]
}
```

## Snowball Edge에서 인증서 가져오기

`get-certificate` 명령을 사용하여 제공된 ARN을 기반으로 인증서의 내용을 확인합니다. `list-certificates` 명령을 사용하여 `certificate-arn` 파라미터로 사용할 인증서의 ARN을 가져옵니다.

```
snowballEdge get-certificate --certificate-arn arn:aws:snowball-
device:::certificate/78EXAMPLE516EXAMPLEf538EXAMPLEa7
```

## Example `get-certificate` 출력

```
-----BEGIN CERTIFICATE-----
Certificate
-----END CERTIFICATE-----
```

인증서 구성에 대한 자세한 내용은 [Snowball Edge에서 S3 어댑터를 엔드포인트로 AWS CLI 사용하도록 구성](#) 섹션을 참조하세요.

## Snowball Edge에서 인증서 삭제

`delete-certificate` 명령을 실행하여 현재 인증서를 삭제합니다. `list-certificates` 명령을 사용하여 `certificate-arn` 파라미터로 사용할 인증서의 ARN을 가져옵니다. 새 인증서를 생성하려면 Snowball Edge 또는 클러스터의 각 Snowball Edge를 재부팅합니다. [Snowball Edge 디바이스 재부팅](#)을 참조하거나 `snowballEdge reboot-device` 명령을 사용합니다.

```
snowballEdge delete-certificate --certificate-arn arn:aws:snowball-  
device::certificate/78EXAMPLE516EXAMPLEf538EXAMPLEa7
```

### Example **delete-certificate** 출력

The certificate has been deleted from your Snow device. Please reboot your Snowball Edge or Snowball Edge cluster to generate a new certificate.

## Snowball Edge의 AWS 서비스에 대한 포트 요구 사항

AWS Snowball Edge 디바이스에서 AWS 서비스가 제대로 작동하려면 서비스의 네트워크 포트를 허용해야 합니다.

다음은 각 AWS 서비스에 필요한 네트워크 포트 목록입니다.

Port	프로토콜	설명
22 (HTTP)	TCP	디바이스 상태 확인 및 EC2 SSH용 디바이스
443(HTTPS)	TCP	S3 API 및 S3 제어 API HTTPS 엔드포인트
2049 (HTTP)	TCP	NFS 엔드포인트
6078(HTTP)	TCP	IAM HTTP 엔드포인트
6089(HTTPS)	TCP	IAM HTTPS 엔드포인트
7078(HTTP)	TCP	STS HTTP 엔드포인트
7089(HTTPS)	TCP	STS HTTPS 엔드포인트
8080(HTTP)	TCP	S3 어댑터 HTTP 엔드포인트
8008(HTTP)	TCP	EC2 HTTP 엔드포인트
8243(HTTPS)	TCP	EC2 HTTPS 엔드포인트
8443(HTTPS)	TCP	S3 어댑터 HTTPS 엔드포인트
9091 (HTTP)	TCP	디바이스 관리용 엔드포인트
9092	TCP	EKS Anywhere 및 CAPAS 디바이스 컨트롤러 인바운드
8242	TCP	EKS Anywhere용 EC2 HTTPS 엔드포인트 인바운드

Port	프로토콜	설명
6443	TCP	EKS Anywhere Kubernetes API 엔드포인트 인바운드
2379	TCP	EKS Anywhere Etcd API 엔드포인트 인바운드
2380	TCP	EKS Anywhere Etcd API 엔드포인트 인바운드



# AWS Snowball Edge Device Management 를 사용하여 Snowball Edge 관리

AWS Snowball Edge Device Management 를 사용하면 Snowball Edge 및 로컬 AWS 서비스를 원격으로 관리할 수 있습니다. 모든 Snowball Edge는 Snowball Edge 디바이스 관리를 지원하며 Snowball Edge를 사용할 수 있는 대부분의 AWS 리전 에서 새 디바이스에 설치됩니다.

Snowball Edge 디바이스 관리를 사용하면 다음 작업을 수행할 수 있습니다.

- 작업 생성
- 태스크 상태 확인
- 태스크 메타데이터 확인
- 태스크 취소
- 디바이스 정보 확인
- Amazon EC2 호환 인스턴스 상태 확인
- 명령 및 구문 나열
- 원격 관리 가능 디바이스 나열
- 디바이스 전반의 태스크 상태 나열
- 사용 가능한 리소스 나열
- 상태별로 태스크 나열
- 디바이스 또는 태스크 태그 나열
- 태그 적용
- 태그 제거

## 주제

- [Snowball Edge 주문 시 Snowball Edge 디바이스 관리 상태 선택](#)
- [Snowball Edge에서 Snowball Edge 디바이스 관리 활성화](#)
- [Snowball Edge의 IAM 역할에 Snowball Edge 디바이스 관리에 대한 권한 추가](#)
- [Snowball Edge Device Management CLI 명령](#)

## Snowball Edge 주문 시 Snowball Edge 디바이스 관리 상태 선택

Snow 디바이스를 주문하는 작업을 생성할 때 디바이스를 수신할 때 Snowball Edge 디바이스 관리의 상태를 선택할 수 있습니다. 설치되었지만 활성화되지 않았거나 설치 및 활성화되지 않았습니다. 설치되었지만 활성화되지 않은 경우 사용하기 전에 AWS OpsHub 또는 Snowball Edge 클라이언트를 사용하여 활성화해야 합니다. 설치 및 활성화된 경우 디바이스를 수신하고 로컬 네트워크에 연결한 후 Snowball Edge Device Management를 사용할 수 있습니다. 작업을 생성할 때 Snowball Edge 디바이스 관리 상태를 선택하여 AWS Snow 패밀리 관리 콘솔, Snowball Edge 클라이언트 AWS CLI, 또는 Snow 작업 관리 API를 통해 디바이스를 주문할 수 있습니다.

에서 Snowball Edge 디바이스 관리 상태를 선택하려면 AWS Snow 패밀리 관리 콘솔

1. Snowball Edge Device Management를 설치하고 활성화하도록 선택하려면 AWS OpsHub 또는 Snowball 클라이언트를 사용하여 원격으로 Snow 디바이스 관리를 선택합니다.
2. Snowball Edge Device Management를 설치하되 활성화하지 않도록 선택하려면 AWS OpsHub 또는 Snowball 클라이언트를 사용하여 원격으로 Snow 디바이스 관리를 선택하지 마십시오.

자세한 내용은 이 안내서의 [3단계: 기능 및 옵션 선택](#) 을 참조하세요.

AWS CLI, Snowball Edge 클라이언트 또는 Snow 작업 관리 API에서 Snowball Edge 디바이스 관리 상태를 선택하려면

- `remote-management` 파라미터를 사용하여 Snowball Edge 디바이스 관리 상태를 지정합니다. 파라미터 `INSTALLED_ONLY` 값은 Snowball Edge Device Management가 설치되었지만 활성화되지 않았음을 의미합니다. 파라미터 `INSTALLED_AUTOSTART` 값은 Snowball Edge Device Management가 설치 및 활성화되었음을 의미합니다. 이 파라미터에 대한 값을 지정하지 않을 경우 기본값은 `INSTALLED_ONLY`입니다.

Example `create-job` 명령 `remote-management` 파라미터의 구문

```
aws snowball create-job \
  --job-type IMPORT \
  --remote-management INSTALLED_AUTOSTART
  --device-configuration '{"SnowconeDeviceConfiguration": {"WirelessConnection":
{"IsWifiEnabled": false} } }' \
  --resources '{"S3Resources":[{"BucketArn":"arn:aws:s3:::bucket-name"}]}' \
  --description "Description here" \
```

```

--address-id ADID00000000-0000-0000-0000-000000000000 \
--kms-key-arn arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab \
--role-arn arn:aws:iam::000000000000:role/SnowconeImportGamma \
--snowball-capacity-preference T8 \
--shipping-option NEXT_DAY \
--snowball-type SNC1_HDD \
--region us-west-2 \

```

자세한 내용은 [API 참조의 작업 관리](#) AWS Snowball Edge API 참조를 참조하세요.

## Snowball Edge에서 Snowball Edge 디바이스 관리 활성화

다음 절차에 따라 Snowball Edge 클라이언트를 사용하여 Snowball Edge 디바이스 관리를 활성화합니다.

이 절차를 사용하기 전에 다음을 수행하십시오.

- 최신 버전의 Snowball Edge 클라이언트 CLI를 다운로드 및 설치합니다. 자세한 내용은 [Snowball 클라이언트 다운로드 및 설치](#)를 참조하십시오.
- 매니페스트 파일을 다운로드하고 Snowball Edge 디바이스의 잠금 해제 코드를 가져옵니다. 자세한 내용은 [자격 증명 및 도구 가져오기](#)를 참조하세요.
- Snowball Edge 디바이스를 로컬 네트워크에 연결합니다. 자세한 내용은 [로컬 NetworkDevice 사양에 연결을 참조하세요](#).
- Snowball Edge 디바이스의 잠금을 해제합니다. 자세한 내용은 [Snowball Edge 잠금 해제](#)를 참조하세요.

```

snowballEdge set-features /
--remote-management-state INSTALLED_AUTOSTART /
--manifest-file JID1717d8cc-2dc9-4e68-aa46-63a3ad7927d2_manifest.bin /
--unlock-code 7c0e1-bab84-f7675-0a2b6-f8k33 /
--endpoint https://192.0.2.0:9091

```

Snowball Edge 클라이언트는 명령이 성공하면 다음을 반환합니다.

```
{
  "RemoteManagementState" : "INSTALLED_AUTOSTART"
}
```

## Snowball Edge의 IAM 역할에 Snowball Edge 디바이스 관리에 대한 권한 추가

디바이스가 주문된 AWS 계정에서 (IAM) 역할을 생성하고 AWS Identity and Access Management 역할에 다음 정책을 추가합니다. 그런 다음 Snowball Edge Device Management로 디바이스를 원격으로 관리하기 위해 로그인할 IAM 사용자에게 역할을 할당합니다. 자세한 내용은 [IAM 역할 생성](#) 및 [AWS 계정에서 IAM 사용자 생성](#)을 참조하세요.

### 정책

#### JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "snow-device-management:ListDevices",
        "snow-device-management:DescribeDevice",
        "snow-device-management:DescribeDeviceEc2Instances",
        "snow-device-management:ListDeviceResources",
        "snow-device-management:CreateTask",
        "snow-device-management:ListTasks",
        "snow-device-management:DescribeTask",
        "snow-device-management:CancelTask",
        "snow-device-management:DescribeExecution",
        "snow-device-management:ListExecutions",
        "snow-device-management:ListTagsForResource",
        "snow-device-management:TagResource",
        "snow-device-management:UntagResource"
      ],
      "Resource": "*"
    }
  ]
}
```

```
    ]
}
```

## Snowball Edge Device Management CLI 명령

이 섹션에서는 Snowball Edge 디바이스 관리를 사용하여 Snowball Edge를 원격으로 관리하는 데 사용할 수 있는 AWS CLI 명령을 설명합니다. 를 사용하여 일부 원격 관리 작업을 수행할 수도 있습니다 AWS OpsHub. 자세한 내용은 [디바이스에서 AWS 서비스 관리를 참조하세요](#).

### Note

디바이스를 관리하기 전에 디바이스가 켜져 있고 네트워크에 연결되어 있으며 프로비저닝된 AWS 리전 에 연결할 수 있는지 확인합니다.

### 주제

- [Snowball Edge 디바이스 관리를 사용하여 Snowball Edge를 관리하는 작업 생성](#)
- [Snowball Edge를 관리하기 위한 작업 상태 확인](#)
- [Snowball Edge 디바이스 관리를 사용하여 Snowball Edge에 대한 정보 확인](#)
- [Snowball Edge 디바이스 관리를 사용하여 Snowball Edge에서 Amazon EC2-compatible 인스턴스 상태 확인](#)
- [Snowball Edge 디바이스 관리를 사용하여 Snowball Edge에서 태스크 메타데이터 보기](#)
- [Snowball Edge 디바이스 관리를 사용하여 Snowball Edge에서 작업 취소](#)
- [Snowball Edge 디바이스 관리 명령 및 구문 나열](#)
- [원격 관리에 사용할 수 있는 Snowball Edge 나열](#)
- [Snowball Edge에서 Snowball Edge 디바이스 관리 작업 상태 나열](#)
- [Snowball Edge 디바이스 관리를 사용하여 Snowball Edge에서 사용 가능한 리소스 나열](#)
- [Snowball Edge 또는 Snowball Edge 디바이스 관리 태그 나열](#)
- [상태별 Snowball Edge 디바이스 관리 작업 나열](#)
- [Snowball Edge 디바이스 관리 작업 또는 Snowball Edge에 태그 적용](#)
- [작업 또는 Snowball Edge에서 Snowball Edge 디바이스 관리 태그 제거](#)

## Snowball Edge 디바이스 관리를 사용하여 Snowball Edge를 관리하는 작업 생성

하나 이상의 대상 디바이스에 잠금 해제 또는 재부팅과 같은 작업을 수행하도록 지시하려면 `create-task`를 사용합니다. `--targets` 파라미터와 함께 관리 대상 디바이스 ID 목록을 제공하여 대상 디바이스를 지정하고, `--command` 파라미터로 수행할 작업을 지정합니다. 디바이스에서 한 번에 하나의 명령을 실행할 수 있습니다.

지원되는 명령:

- `unlock`(인수 없음)
- `reboot`(인수 없음)

대상 디바이스에서 실행할 작업을 생성하려면 다음 명령을 사용합니다. 각 *user input placeholder*를 사용자의 정보로 바꿉니다.

명령

```
aws snow-device-management create-task
--targets smd-fictbgr3rbcjeqa5
--command reboot={}
```

예외

```
ValidationException
ResourceNotFoundException
InternalServerError
ThrottlingException
AccessDeniedException
ServiceQuotaExceededException
```

출력

```
{
```

```

    "taskId": "st-ficthmqoc2pht111",
    "taskArn": "arn:aws:snow-device-management:us-west-2:000000000000:task/st-
    cjkwhmqoc2pht111"
  }

```

## Snowball Edge를 관리하기 위한 작업 상태 확인

하나 이상의 대상 디바이스에서 실행 중인 원격 태스크의 상태를 확인하려면 `describe-execution` 명령을 사용합니다.

태스크는 다음 중 하나의 상태를 가질 수 있습니다.

- QUEUED
- IN\_PROGRESS
- CANCELED
- FAILED
- COMPLETED
- REJECTED
- TIMED\_OUT

태스크 상태를 확인하려면 다음 명령을 사용합니다. 각 *user input placeholder*를 사용자의 정보로 바꿉니다.

### 명령

```

aws snow-device-management describe-execution \
  --taskId st-ficthmqoc2pht1ef \
  --managed-device-id smd-fictqic6gcldf111

```

### 출력

```

{
  "executionId": "1",
  "lastUpdatedAt": "2021-07-22T15:29:44.110000+00:00",

```

```

    "managedDeviceId": "smd-fictqic6gcldf111",
    "startedAt": "2021-07-22T15:28:53.947000+00:00",
    "state": "SUCCEEDED",
    "taskId": "st-ficthmqoc2pht111"
  }

```

## Snowball Edge 디바이스 관리를 사용하여 Snowball Edge에 대한 정보 확인

디바이스 유형, 소프트웨어 버전, IP 주소, 잠금 상태 등 디바이스별 정보를 확인하려면 `describe-device` 명령을 사용합니다. 출력에는 다음이 포함됩니다.

- `lastReachedOutAt` - 디바이스가 AWS 클라우드에 마지막으로 연락한 시점입니다. 디바이스가 운 라인 상태임을 나타냅니다.
- `lastUpdatedAt` - 디바이스에서 데이터가 마지막으로 업데이트된 시점. 디바이스 캐시가 새로 고쳐진 시간을 나타냅니다.

다음 명령을 사용하여 디바이스를 확인합니다. 각 *user input placeholder*를 사용자의 정보로 바꿉니다.

### 명령

```

aws snow-device-management describe-device \
  --managed-device-id smd-fictqic6gcldf111

```

### 예외

```

ValidationException
ResourceNotFoundException
InternalServerError
ThrottlingException
AccessDeniedException

```



## Snowball Edge 디바이스 관리를 사용하여 Snowball Edge에서 Amazon EC2-compatible 인스턴스 상태 확인

Amazon EC2 인스턴스의 현재 상태를 확인하려면 `describe-ec2-instances` 명령을 사용합니다. 출력은 `describe-device` 명령의 출력과 비슷하지만 결과는 디바이스 캐시에서 가져오 AWS 클라우드 고 사용 가능한 필드의 하위 집합을 포함합니다.

Amazon EC2 호환 인스턴스의 현재 상태를 확인하려면 다음 명령을 사용합니다. 각 *user input placeholder*를 사용자의 정보로 바꿉니다.

### 명령

```
aws snow-device-management describe-device-ec2-instances \  
--managed-device-id smd-fictbgr3rbcje111 \  
--instance-ids s.i-84fa8a27d3e15e111
```

### 예외

```
ValidationException  
ResourceNotFoundException  
InternalServerError  
ThrottlingException  
AccessDeniedException
```

### 출력

```
{  
  "instances": [  
    {  
      "instance": {  
        "amiLaunchIndex": 0,  
        "blockDeviceMappings": [  
          {  
            "deviceName": "/dev/sda",  
            "ebs": {  
              "attachTime": "2021-07-23T15:25:38.719000-07:00",
```

```

        "deleteOnTermination": true,
        "status": "ATTACHED",
        "volumeId": "s.vol-84fa8a27d3e15e111"
    }
}
],
"cpuOptions": {
    "coreCount": 1,
    "threadsPerCore": 1
},
"createdAt": "2021-07-23T15:23:22.858000-07:00",
"imageId": "s.ami-03f976c3cadaa6111",
"instanceId": "s.i-84fa8a27d3e15e111",
"state": {
    "name": "RUNNING"
},
"instanceType": "snc1.micro",
"privateIpAddress": "34.223.14.193",
"publicIpAddress": "10.111.60.160",
"rootDeviceName": "/dev/sda",
"securityGroups": [
    {
        "groupId": "s.sg-890b6b4008bdb3111",
        "groupName": "default"
    }
],
"updatedAt": "2021-07-23T15:29:42.163000-07:00"
},
"lastUpdatedAt": "2021-07-23T15:29:58.
071000-07:00"
}
]
}

```

## Snowball Edge 디바이스 관리를 사용하여 Snowball Edge에서 태스크 메타 데이터 보기

디바이스에서 특정 태스크의 메타데이터를 확인하려면 `describe-task` 명령을 사용합니다. 태스크의 메타데이터에는 다음 항목이 포함됩니다.

- 대상 디바이스

- 태스크 상태
- 태스크 생성 시점
- 디바이스에서 데이터가 마지막으로 업데이트된 시점
- 태스크 완료 시점
- 태스크가 생성될 때 제공된 설명(있는 경우)

태스크 메타데이터를 확인하려면 다음 명령을 사용합니다. 각 *user input placeholder*를 사용자의 정보로 바꿉니다.

## 명령

```
aws snow-device-management describe-task \
--task-id st-ficthmqoc2pht111
```

## 예외

```
ValidationException
ResourceNotFoundException
InternalServerError
ThrottlingException
AccessDeniedException
```

## 출력

```
{
  "completedAt": "2021-07-22T15:29:46.758000+00:00",
  "createdAt": "2021-07-22T15:28:42.613000+00:00",
  "lastUpdatedAt": "2021-07-22T15:29:46.758000+00:00",
  "state": "COMPLETED",
  "tags": {},
  "targets": [
    "smd-fictbgr3rbcje111"
  ],
  "taskId": "st-ficthmqoc2pht111",
```

```
"taskArn": "arn:aws:snow-device-management:us-west-2:000000000000:task/st-
ficthmqoc2pht111"
}
```

## Snowball Edge 디바이스 관리를 사용하여 Snowball Edge에서 작업 취소

특정 태스크의 취소 요청을 보내려면 `cancel-task` 명령을 사용합니다. 아직 실행되지 않은 QUEUED 상태의 태스크만 취소할 수 있습니다. 이미 실행 중인 태스크는 취소할 수 없습니다.

### Note

취소하려는 태스크가 `cancel-task` 명령으로 태스크 상태를 변경하기 전에 대기열에서 처리 되면 계속 실행될 수 있습니다.

태스크를 취소하려면 다음 명령을 사용합니다. 각 *user input placeholder*를 사용자의 정보로 바꿉니다.

### 명령

```
aws snow-device-management cancel-task \
--task-id st-ficthmqoc2pht111
```

### 예외

```
ValidationException
ResourceNotFoundException
InternalServerError
ThrottlingException
AccessDeniedException
```

### 출력

```
{
  "taskId": "st-ficthmqoc2pht111"
```

}

## Snowball Edge 디바이스 관리 명령 및 구문 나열

Snowball Edge Device Management API에 대해 지원되는 모든 명령 목록을 반환하려면 `help` 명령을 사용합니다. 또한 `help` 명령을 사용하여 해당 명령에 대한 세부 정보와 구문을 반환할 수 있습니다.

지원되는 명령을 모두 나열하려면 다음 명령을 사용합니다.

### 명령

```
aws snow-device-management help
```

명령의 구체적인 정보 및 구문을 반환하려면 다음 명령을 사용합니다. *command*을 관심 있는 명령의 이름으로 바꿉니다.

### 명령

```
aws snow-device-management command help
```

## 원격 관리에 사용할 수 있는 Snowball Edge 나열

AWS 리전 명령이 실행되는에서 Snowball Edge Device Management가 활성화된 계정의 모든 디바이스 목록을 반환하려면 `list-devices` 명령을 사용합니다. `--max-results` 및는 선택 사항 `--next-token`입니다. 자세한 내용은 "AWS 명령줄 인터페이스 사용 설명서"의 [AWS CLI 페이지 매김 옵션 사용](#)을 참조하세요.

원격 관리 가능 디바이스의 목록을 보려면 다음 명령을 사용합니다. 각 *user input placeholder*를 사용자의 정보로 바꿉니다.

### 명령

```
aws snow-device-management list-devices \  
--max-results 10
```

### 예외

```

ValidationException
InternalServerError
ThrottlingException
AccessDeniedException

```

## 출력

```

{
  "devices": [
    {
      "associatedWithJob": "ID2bf11d5a-ea1e-414a-b5b1-3bf7e6a6e111",
      "managedDeviceId": "smd-fictbgr3rbcjeqa5",
      "managedDeviceArn": "arn:aws:snow-device-management:us-
west-2:000000000000:managed-device/smd-fictbgr3rbcje111"
      "tags": {}
    }
  ]
}

```

## Snowball Edge에서 Snowball Edge 디바이스 관리 작업 상태 나열

하나 이상의 대상 디바이스에 대한 태스크 상태를 반환하려면 `list-executions` 명령을 사용합니다. 반환 목록을 필터링하여 현재 하나의 특정 상태에 있는 작업을 표시하려면 `--state` 파라미터를 사용합니다. `--max-results` 및 `--next-token`은 선택 사항입니다. 자세한 내용은 "AWS 명령줄 인터페이스 사용 설명서"의 [AWS CLI 페이지 매김 옵션 사용](#)을 참조하세요.

태스크는 다음 중 하나의 상태를 가질 수 있습니다.

- QUEUED
- IN\_PROGRESS
- CANCELED
- FAILED
- COMPLETED
- REJECTED
- TIMED\_OUT

여러 디바이스의 태스크 상태를 나열하려면 다음 명령을 사용합니다. 각 *user input placeholder*를 사용자의 정보로 바꿉니다.

## 명령

```
aws snow-device-management list-executions \
--taskId st-ficthmqoc2phtlef \
--state SUCCEEDED \
--max-results 10
```

## 예외

```
ValidationException
InternalServerError
ThrottlingException
AccessDeniedException
```

## 출력

```
{
  "executions": [
    {
      "executionId": "1",
      "managedDeviceId": "smd-fictbgr3rbcje111",
      "state": "SUCCEEDED",
      "taskId": "st-ficthmqoc2pht111"
    }
  ]
}
```

## Snowball Edge 디바이스 관리를 사용하여 Snowball Edge에서 사용 가능한 리소스 나열

디바이스에 사용할 수 있는 AWS 리소스 목록을 반환하려면 `list-device-resources` 명령을 사용합니다. 특정 유형의 리소스별로 목록을 필터링하려면 `--type` 파라미터를 사용합니다. 현재 지원되는 리소스 유형은 Amazon EC2 호환 인스턴스뿐입니다. `--max-results` 및 `--next-token`은 선택 사

항입니다. 자세한 내용은 "AWS 명령줄 인터페이스 사용 설명서"의 [AWS CLI 페이지 매김 옵션 사용](#)을 참조하세요.

디바이스의 사용 가능한 리소스를 나열하려면 다음 명령을 사용합니다. 각 *user input placeholder*를 사용자의 정보로 바꿉니다.

## 명령

```
aws snow-device-management list-device-resources \
--managed-device-id smd-fictbgr3rbcje111 \
--type AWS::EC2::Instance
--next-
token YAQGPwAT9L3wVKaGYjt4yS34MiQLWvzcShe9oIeDJr05AT4rXSprqcqQhhBEYRfcerAp0YYbJmRT=
--max-results 10
```

## 예외

```
ValidationException
InternalServerError
ThrottlingException
AccessDeniedException
```

## 출력

```
{
  "resources": [
    {
      "id": "s.i-84fa8a27d3e15e111",
      "resourceType": "AWS::EC2::Instance"
    }
  ]
}
```

## Snowball Edge 또는 Snowball Edge 디바이스 관리 태그 나열

관리 대상 디바이스 또는 태스크의 태그 목록을 반환하려면 `list-tags-for-resource` 명령을 사용합니다.



디바이스 태그를 나열하려면 다음 명령을 사용합니다. Amazon 리소스 이름(ARN)을 디바이스의 ARN으로 바꿉니다.

## 명령

```
aws snow-device-management list-tags-for-resource
--resource-arn arn:aws:snow-device-management:us-west-2:123456789012:managed-device/smd-fictbgr3rbcjeqa5
```

## 예외

```
AccessDeniedException
InternalServerError
ResourceNotFoundException
ThrottlingException
```

## 출력

```
{
  "tags": {
    "Project": "PrototypeA"
  }
}
```

## 상태별 Snowball Edge 디바이스 관리 작업 나열

`list-tasks` 명령을 사용하여 명령이 실행되는 AWS 리전의 디바이스에서 작업 목록을 반환합니다. `IN_PROGRESS`, `COMPLETED` 또는 `CANCELED` 상태별로 결과를 필터링하려면 `--state` 파라미터를 사용합니다. `--max-results` 및 `--next-token`은 선택 사항입니다. 자세한 내용은 "AWS 명령줄 인터페이스 사용 설명서"의 [AWS CLI 페이지 매김 옵션 사용](#)을 참조하세요.

상태별로 태스크를 나열하려면 다음 명령을 사용합니다. 각 *user input placeholder*를 사용자의 정보로 바꿉니다.

## 명령

```
aws snow-device-management list-tasks \
--state IN_PROGRESS \
--next-token K8VAMqKiP2Cf4xGkmH8GMyZrg0F8FUb+d10KTP9+P4pUb+8PhW+6MiXh4= \
--max-results 10
```

## 예외

```
ValidationException
InternalServerError
ThrottlingException
AccessDeniedException
```

## 출력

```
{
  "tasks": [
    {
      "state": "IN_PROGRESS",
      "tags": {},
      "taskId": "st-ficthmqoc2phtlef",
      "taskArn": "arn:aws:snow-device-management:us-west-2:000000000000:task/st-ficthmqoc2phtlef"
    }
  ]
}
```

## Snowball Edge 디바이스 관리 작업 또는 Snowball Edge에 태그 적용

디바이스 또는 디바이스상의 태스크에 태그를 추가하거나 바꾸려면 `tag-resource` 명령을 사용합니다. `--tags` 파라미터는 쉼표로 구분된 Key=Value 페어 목록을 허용합니다.

디바이스 태그를 적용하려면 다음 명령을 사용합니다. 각 *user input placeholder*를 사용자의 정보로 바꿉니다.

## 명령

```
aws snow-device-management tag-resource \
```

```
--resource-arn arn:aws:snow-device-management:us-west-2:123456789012:managed-device/  
smd-fictbgr3rbcjeqa5 \  
--tags Project=PrototypeA
```

## 예외

```
AccessDeniedException  
InternalServerError  
ResourceNotFoundException  
ThrottlingException
```

## 작업 또는 Snowball Edge에서 Snowball Edge 디바이스 관리 태그 제거

디바이스 또는 디바이스상의 태스크에서 태그를 제거하려면 `untag-resources` 명령을 사용합니다.

디바이스에서 태그를 제거하려면 다음 명령을 사용합니다. 각 *user input placeholder*를 사용자의 정보로 바꿉니다.

## 명령

```
aws snow-device-management untag-resources \  
--resource-arn arn:aws:snow-device-management:us-west-2:123456789012:managed-device/  
smd-fictbgr3rbcjeqa5 \  
--tag-keys Project
```

## 예외

```
AccessDeniedException  
InternalServerError  
ResourceNotFoundException  
ThrottlingException
```

# Snowball Edge 디바이스의 소프트웨어 업데이트

AWS 는 Snowball Edge에 새 소프트웨어를 사용할 수 있게 되면 사용자에게 알립니다. 알림은 이메일을 통해 CloudWatch 이벤트 Health Dashboard로 제공됩니다. 이메일 알림은 Amazon Web Services, Inc.에서 Snowball Edge 디바이스 주문에 사용된 AWS 계정에 연결된 이메일 주소로 전송됩니다. 알림을 받으면 이 주제의 지침에 따라 가능한 한 빨리 업데이트를 다운로드하고 설치하여 디바이스 사용 중단을 방지합니다. 에 대한 자세한 내용은 [AWS Health 사용 설명서를](#) Health Dashboard참조하세요. CloudWatch Events에 대한 자세한 내용은 [Amazon CloudWatch Events 사용 설명서](#)를 참조하세요.

에서 소프트웨어 업데이트를 다운로드 AWS 하여 온프레미스 환경의 Snowball Edge 디바이스에 설치할 수 있습니다. 이러한 업데이트는 백그라운드에서 발생합니다. 최신 소프트웨어가에서 디바이스로 안전하게 다운로드되는 동안 디바이스를 정상적으로 계속 사용할 수 AWS 있습니다. 하지만 다운로드한 업데이트를 적용하려면 디바이스의 워크로드를 중지하고 다시 시작해야 합니다.

Snowball Edge/Snowball Edge 디바이스(어플라이언스)에 AWS 대해에서 제공하는 소프트웨어 업데이트는 서비스 약관의 섹션 9에 따른 어플라이언스 소프트웨어입니다.

소프트웨어 업데이트는 AWS를 대신하여 해당 어플라이언스에 소프트웨어 업데이트를 설치할 목적으로만 제공됩니다. 귀하는 (i) 해당 어플라이언스에 소프트웨어 업데이트를 설치하는 데 필요한 것 이외의 소프트웨어 업데이트 사본을 만들거나 (ii) 소프트웨어 업데이트에 적용된 암호화를 포함하되 이에 국한되지 않는 소프트웨어 업데이트의 기능 또는 조치를 우회 또는 비활성화하거나 이를 시도하지 않으며, 제3자가 이를 수행하거나 수행을 시도하는 것을 허용하거나 승인하지 않습니다. 해당 어플라이언스에 소프트웨어 업데이트를 설치한 후에는 어플라이언스에 소프트웨어 업데이트를 설치하는 데 사용된 모든 미디어에서 소프트웨어 업데이트를 삭제하는 데 동의합니다.

## Warning

업데이트를 설치하기 전에 디바이스의 모든 활동을 일시 중지하는 것이 좋습니다. 디바이스를 업데이트하고 다시 시작하면 인스턴스 실행이 중지되고 로컬 Amazon S3 버킷에 대한 모든 쓰기가 중단됩니다.

## 주제

- [Snowball Edge 디바이스의 소프트웨어 업데이트를 위한 사전 조건](#)
- [Snowball Edge 디바이스에 업데이트 다운로드](#)
- [Snowball Edge 디바이스에 업데이트 설치](#)
- [Snowball Edge 디바이스에서 SSL 인증서 업데이트](#)

- [Snowball Edge에서 Amazon Linux 2 AMIs 업데이트](#)

## Snowball Edge 디바이스의 소프트웨어 업데이트를 위한 사전 조건

디바이스를 업데이트하기 전에 다음 사전 조건이 충족되어야 합니다.

- 작업을 생성했고, 온프레미스에 디바이스가 있으며, 이를 잠금 해제했습니다. 자세한 내용은 [Snowball Edge 시작하기](#) 섹션을 참조하십시오.
- Snowball Edge 디바이스 업데이트는 Snowball Edge 클라이언트를 통해 이루어집니다. 업데이트할 디바이스에 네트워크로 연결되어 있는 로컬 환경의 컴퓨터에 최신 버전의 Snowball Edge 클라이언트를 다운로드하여 설치해야 합니다. 자세한 내용은 [Snowball Edge 클라이언트 사용](#)을 참조하십시오.
- (선택 사항) Snowball Edge 클라이언트에 대한 프로필을 구성하는 것이 좋습니다. 자세한 내용은 [Snowball Edge 클라이언트의 프로필 구성](#) 섹션을 참조하십시오.
- 클러스터링된 Snowball Edge 디바이스의 Snowball Edge에 있는 Amazon S3 호환 스토리지의 경우 S3-Snow 서비스를 중지하고 자동 시작을 비활성화합니다. [클 사용하여 자동으로 시작하도록 Snowball Edge에서 Amazon S3 호환 스토리지 구성 AWS OpsHub](#)을 참조하세요.

### Note

클러스터링된 디바이스의 경우 각 디바이스에 대해 모든 명령을 실행해야 합니다.

이러한 작업을 완료했다면 Snowball Edge 디바이스에 대한 업데이트를 다운로드 및 설치할 수 있습니다.

## Snowball Edge 디바이스에 업데이트 다운로드

Snowball Edge에 대한 업데이트를 다운로드할 수 있는 두 가지 방법이 있습니다.

- 특정 Snowball Edge 클라이언트 명령을 사용하여 언제든지 수동 업데이트를 트리거할 수 있습니다.
- 프로그래밍 방식으로 디바이스를 자동으로 업데이트할 시간을 결정할 수 있습니다.

다음 절차는 수동 업데이트 다운로드 프로세스를 간략하게 설명합니다. Snowball Edge 디바이스의 자동 업데이트에 대한 자세한 내용은 [Snowball Edge 업데이트](#)의 `configure-auto-update-strategy`를 참조하십시오.

**Note**

디바이스에서 인터넷에 액세스할 수 없는 경우 [GetSoftwareUpdates](#) API를 사용하여 업데이트 파일을 다운로드할 수 있습니다. 그리고 다음 예와 같이 `uri` 파라미터를 사용하여 `download-updates`를 직접적으로 호출할 때 로컬 파일 위치를 가리킵니다.

```
snowballEdge download-updates --uri file:///tmp/local-update
```

Windows 운영 체제의 경우 `uri` 파라미터 값의 형식을 다음과 같이 지정합니다.

```
snowballEdge download-updates --uri file:/C:/path/to/local-update
```

독립형 디바이스를 위한 Snowball Edge 소프트웨어 업데이트를 확인하고 다운로드하려면

1. 터미널 창을 열고 Snowball Edge 디바이스가 `describe-device` 명령으로 잠금 해제되었는지 확인합니다. 디바이스가 잠긴 경우 `unlock-device` 명령을 사용하여 잠금 해제합니다. 자세한 내용은 [Snowball Edge 잠금 해제](#)를 참조하세요.
2. 디바이스가 잠금 해제되면 `snowballEdge check-for-updates` 명령을 사용합니다. 이 명령은 Snowball Edge 소프트웨어의 최신 사용 가능한 버전을 반환하고, 디바이스의 현재 설치된 버전을 반환합니다.
3. 디바이스 소프트웨어가 최신이 아닌 경우 `snowballEdge download-updates` 명령을 실행합니다.

**Note**

디바이스가 인터넷에 연결되어 있지 않으면 먼저 [GetSoftwareUpdates](#) API를 사용하여 업데이트 파일을 다운로드합니다. 그리고 다음 예와 같이 다운로드한 파일에 대한 로컬 경로와 함께 `uri` 파라미터를 사용하여 `snowballEdge download-updates` 명령을 실행합니다.

```
snowballEdge download-updates --uri file:///tmp/local-update
```

Windows 운영 체제의 경우 `uri` 파라미터 값의 형식을 다음과 같이 지정합니다.

```
snowballEdge download-updates --uri file:/C:/path/to/local-update
```

4. `snowballEdge describe-device-software` 명령을 사용하여 다운로드 상태를 확인할 수 있습니다. 업데이트를 다운로드하는 동안 이 명령을 사용하면 상태가 표시됩니다.

#### Example `describe-device-software` 명령의 출력

```
Install State: Downloading
```

디바이스의 클러스터를 위한 Snowball Edge 소프트웨어 업데이트를 확인하고 다운로드하려면

1. 터미널 창을 열고 클러스터의 모든 Snowball Edge 디바이스가 `snowballEdge describe-device` 명령으로 잠금 해제되었는지 확인합니다. 디바이스가 잠긴 경우 `snowballEdge unlock-cluster` 명령을 사용하여 잠금 해제합니다. 자세한 내용은 [Snowball Edge 잠금 해제](#)를 참조하세요.
2. 클러스터의 모든 디바이스가 잠금 해제되면 클러스터의 각 디바이스에 대해 `check-for-updates` 명령을 실행합니다. 이 명령은 Snowball Edge 소프트웨어의 최신 사용 가능한 버전을 반환하고, 디바이스의 현재 설치된 버전을 반환합니다.

```
snowballEdge check-for-updates --unlock-code 29-character-unlock-code --manifest-file path/to/manifest/file.bin --endpoint https://ip-address-of-snow-device
```

#### Note

잠금 해제 코드와 매니페스트 파일은 클러스터의 모든 디바이스에서 동일합니다.

#### Example/ `check-for-updates` 명령

```
{
  "InstalledVersion" : "118",
  "LatestVersion" : "119"
}
```

LatestVersion 이름 값이 InstalledVersion 이름 값보다 큰 경우 업데이트를 사용할 수 있습니다.

- 클러스터의 각 디바이스에 대해 download-updates 명령을 사용하여 업데이트를 다운로드합니다.

```
snowballEdge download-updates --uri file:///tmp/local-update
```

#### Note

Windows 운영 체제의 경우 uri 파라미터 값의 형식을 다음과 같이 지정합니다.

```
snowballEdge download-updates --uri file:/C:/path/to/local-update
```

- 클러스터의 각 디바이스에 대한 이 다운로드 상태를 확인하려면 describe-device-software 명령을 사용합니다.

```
snowballEdge describe-device-software --unlock-code 29-character-unlock-code --manifest-file path/to/manifest/file.bin --endpoint https://ip-address-of-snow-device
```

#### Example/ describe-device-software 명령의 출력

```
{
  "InstalledVersion" : "118",
  "InstallingVersion" : "119",
  "InstallState" : "DOWNLOADED",
  "CertificateExpiry" : "Sat Mar 30 16:47:51 UTC 2024"
}
```

InstallState 이름 값이 DOWNLOADED인 경우 업데이트가 다운로드 완료되고 설치할 수 있습니다.



## Snowball Edge 디바이스에 업데이트 설치

업데이트를 다운로드한 후 이를 설치하고 디바이스를 재시작해야 업데이트가 적용됩니다. 다음 절차는 수동으로 업데이트를 설치하는 방법을 안내합니다.

Snowball Edge 디바이스 클러스터의 경우 클러스터의 각 디바이스에 대해 업데이트를 다운로드하고 설치해야 합니다.

### Note

소프트웨어 업데이트를 설치하기 전에 디바이스의 모든 활동을 일시 중단합니다. 업데이트를 설치하면 인스턴스 실행이 중지되고 디바이스의 Amazon S3 버킷에 대한 모든 쓰기가 중단됩니다. 이로 인해 데이터가 손실될 수 있습니다.

독립 실행형 Snowball Edge에 이미 다운로드한 소프트웨어 업데이트를 설치하려면

1. 터미널 창을 열고 Snowball Edge 디바이스가 `describe-device` 명령으로 잠금 해제되었는지 확인합니다. 디바이스가 잠긴 경우 `unlock-device` 명령을 사용하여 잠금 해제합니다. 자세한 내용은 [Snowball Edge 잠금 해제](#)를 참조하세요.
2. `list-services` 명령을 실행하여 디바이스에서 사용 가능한 서비스를 확인합니다. 명령은 디바이스에서 사용할 수 있는 각 서비스의 서비스 ID를 반환합니다.

```
snowballEdge list-services
```

Example/ `list-services` 명령의 출력

```
{
  "ServiceIds" : [ "greengrass", "fileinterface", "s3", "ec2", "s3-snow" ]
}
```

3. `list-services` 명령으로 식별된 각 서비스 ID에 대해 `describe-service` 명령을 실행하여 상태를 확인합니다. 이 정보를 사용하여 중지할 서비스를 식별합니다.

```
snowballEdge describe-service --service-id service-id
```

### Example/ describe-service 명령의 출력

```
{
  "ServiceId" : "s3",
  "Status" : {
    "State" : "ACTIVE"
  },
  "Storage" : {
    "TotalSpaceBytes" : 99608745492480,
    "FreeSpaceBytes" : 99608744468480
  },
  "Endpoints" : [ {
    "Protocol" : "http",
    "Port" : 8080,
    "Host" : "192.0.2.0"
  }, {
    "Protocol" : "https",
    "Port" : 8443,
    "Host" : "192.0.2.0",
    "CertificateAssociation" : {
      "CertificateArn" : "arn:aws:snowball-
device::certificate/6d955EXAMPLEdb71798146EXAMPLE3f0"
    }
  } ]
}
```

이 출력은 s3 서비스가 활성 상태이며 stop-service 명령을 사용하여 중지해야 함을 나타냅니다.

4. stop-service 명령을 사용하여 State 이름 값이 list-services 명령의 출력에 있는 ACTIVE인 각 서비스를 중지합니다. 둘 이상의 서비스가 실행 중인 경우 계속하기 전에 각 서비스를 중지합니다.

#### Note

Amazon S3 어댑터, Amazon EC2 AWS STS 및 IAM 서비스는 중지할 수 없습니다. Snowball Edge에서 Amazon S3 호환 스토리지가 실행 중인 경우 업데이트를 설치하기 전

에 중지합니다. Snowball Edge의 Amazon S3 호환 스토리지를 `s3-snow`로 사용합니다. `serviceId`.

```
snowballEdge stop-service --service-id service-id --device-ip-addresses snow-device-1-ip-address snow-device-2-ip-address snow-device-3-ip-address --manifest-file path/to/manifest/file.bin --unlock-code 29-character-unlock-code --endpoint https://snow-device-ip-address
```

#### Example/ stop-service 명령의 출력

Stopping the AWS service on your Snowball Edge. You can determine the status of the AWS service using the describe-service command.

5. `snowballEdge install-updates` 명령을 실행합니다.
6. `snowballEdge describe-device-software` 명령을 사용하여 설치 상태를 확인할 수 있습니다. 업데이트를 설치하는 동안 이 명령을 사용하면 상태가 표시됩니다.

#### 출력 예시

```
Install State: Installing //Possible values[NA, Installing, Requires Reboot]
```

Snowball Edge 디바이스에 대한 소프트웨어 업데이트가 설치되었습니다. 업데이트를 설치해도 디바이스에 대한 업데이트는 자동으로 적용되지 않습니다. 업데이트 설치를 완료하려면 디바이스를 재시작해야 합니다.

#### Warning

디바이스의 모든 활동을 정지하고 않고 Snowball Edge 디바이스를 재시작하면 데이터 손실이 발생할 수 있습니다.

7. 디바이스의 모든 서비스가 중지되면 디바이스를 재부팅하고 디바이스를 잠금 해제한 다음 다시 재부팅합니다. 이렇게 하면 다운로드한 소프트웨어 업데이트의 설치가 완료됩니다. 디바이스 재

부팅에 대한 자세한 내용은 [Snowball Edge 재부팅](#). 디바이스 잠금 해제에 대한 자세한 내용은 [Snowball Edge 잠금 해제](#).

8. 두 번째 재부팅 후 디바이스가 켜지면 디바이스의 잠금을 해제합니다.
9. `check-for-updates` 명령을 실행합니다. 이 명령은 Snowball Edge 소프트웨어의 최신 사용 가능한 버전을 반환하고, 디바이스의 현재 설치된 버전도 반환합니다.

이미 다운로드한 소프트웨어 업데이트를 Snowball Edge 디바이스 클러스터에 설치하려면

1. 클러스터의 각 디바이스에 대해 `describe-device` 명령을 실행하여 디바이스가 잠금 해제되었는지 확인합니다. 디바이스가 잠긴 경우 `unlock-cluster` 명령을 사용하여 잠금 해제합니다. 자세한 내용은 [Snowball Edge 잠금 해제](#)를 참조하세요.
2. 클러스터의 각 디바이스에 대해 `list-services` 명령을 실행하여 디바이스에서 사용 가능한 서비스를 확인합니다. 명령은 디바이스에서 사용할 수 있는 각 서비스의 서비스 ID를 반환합니다.

```
snowballEdge list-services
```

Example/ `list-services` 명령의 출력

```
{
  "ServiceIds" : [ "greengrass", "fileinterface", "s3", "ec2", "s3-snow" ]
}
```

3. `list-services` 명령으로 식별된 각 서비스 ID에 대해 `describe-service` 명령을 실행하여 상태를 확인합니다. 이 정보를 사용하여 중지할 서비스를 식별합니다.

```
snowballEdge describe-service --service-id service-id
```

Example/ `describe-service` 명령의 출력

```
{
  "ServiceId" : "s3",
```

```

"Status" : {
  "State" : "ACTIVE"
},
"Storage" : {
"TotalSpaceBytes" : 99608745492480,
"FreeSpaceBytes" : 99608744468480
},
"Endpoints" : [ {
"Protocol" : "http",
"Port" : 8080,
"Host" : "192.0.2.0"
}, {
"Protocol" : "https",
"Port" : 8443,
"Host" : "192.0.2.0",
"CertificateAssociation" : {
"CertificateArn" : "arn:aws:snowball-
device::certificate/6d955EXAMPLEdb71798146EXAMPLE3f0"
}
} ]
}

```

이 출력은 s3 서비스가 활성 상태이며 stop-service 명령을 사용하여 중지해야 함을 나타냅니다.

- 클러스터의 각 디바이스에 대해 stop-service 명령을 사용하여 list-services 명령의 출력에서 State 이름 값이 ACTIVE인 모든 서비스를 중지합니다. 둘 이상의 서비스가 실행 중인 경우 계속하기 전에 각 서비스를 중지합니다.

#### Note

Amazon S3 어댑터, Amazon EC2 AWS STS 및 IAM 서비스는 중지할 수 없습니다. Snowball Edge에서 Amazon S3 호환 스토리지가 실행 중인 경우 업데이트를 설치하기 전에 중지합니다. Snowball Edge의 Amazon S3 호환 스토리지는 `s3-snow`로 사용됩니다. `serviceId`.

```

snowballEdge stop-service --service-id service-id --device-ip-addresses snow-
device-1-ip-address snow-device-device-2-ip-address snow-device-3-ip-address --

```

```
manifest-file path/to/manifest/file.bin --unlock-code 29-character-unlock-code --
endpoint https://snow-device-ip-address
```

### Example/ stop-service 명령의 출력

```
Stopping the AWS service on your Snowball Edge. You can determine the status of the
AWS service using the describe-service command.
```

- 클러스터의 각 디바이스에 대해 `install-updates` 명령을 실행합니다.

```
snowballEdge install-updates
```

- `describe-device-software` 명령을 사용하여 설치 상태를 확인할 수 있습니다.

```
snowballEdge describe-device-software
```

### Example/ describe-device-service 명령의 출력

```
Install State: Installing //Possible values[NA, Installing, Requires Reboot]
```

Install State가 Requires Reboot인 경우 Snowball Edge 디바이스에 대한 소프트웨어 업데이트가 설치되었습니다. 업데이트를 설치해도 디바이스에 대한 업데이트는 자동으로 적용되지 않습니다. 업데이트 설치를 완료하려면 디바이스를 재시작해야 합니다.

#### Warning

디바이스의 모든 활동을 중지하지 않고 Snowball Edge 디바이스를 재시작하면 데이터 손실이 발생할 수 있습니다.

- 클러스터의 모든 디바이스를 재부팅하고, 클러스터의 잠금을 해제하고, 클러스터의 모든 디바이스를 다시 재부팅합니다. 이렇게 하면 다운로드한 소프트웨어 업데이트의 설치가 완료됩니다. 디

바이스 재부팅에 대한 자세한 내용은 [Snowball Edge 재부팅을 참조하세요](#). 디바이스 클러스터 잠금 해제에 대한 자세한 내용은 [Snowball Edge 잠금 해제](#)를 참조하세요.

- 클러스터의 각 디바이스를 두 번 재부팅한 후 클러스터를 잠금 해제한 다음 `check-for-updates` 명령을 사용하여 디바이스가 업데이트되었는지 확인합니다. 이 명령은 Snowball Edge 소프트웨어의 최신 사용 가능한 버전을 반환하고, 디바이스의 현재 설치된 버전도 반환합니다. 현재 버전과 사용 가능한 최신 버전이 동일한 경우 디바이스가 성공적으로 업데이트된 것입니다.

이제 Snowball Edge 또는 디바이스 클러스터를 성공적으로 업데이트하고 최신 Snowball Edge 소프트웨어로 업데이트했음을 확인했습니다.

## Snowball Edge 디바이스에서 SSL 인증서 업데이트

Snowball Edge를 360일 이상 유지하려는 경우 디바이스 사용이 중단되지 않도록 디바이스에서 Secure Sockets Layer(SSL) 인증서를 업데이트해야 합니다. 인증서가 만료되면 디바이스를 사용할 수 없으므로 디바이스를 AWS에 반환해야 합니다.

AWS 는 보유한 Snowball Edge에 대한 SSL 인증서가 만료되기 30일 전에 사용자에게 알립니다. 알림은 이메일을 통해 Health Dashboard제공되며 AWS CloudTrail 이벤트로 제공됩니다. 이메일 알림은 Amazon Web Services, Inc.에서 Snowball Edge 디바이스 주문에 사용된 AWS 계정에 연결된 이메일 주소로 전송됩니다. 알림을 받으면 이 주제의 지침에 따라 가능한 한 빨리 업데이트를 요청하여 디바이스 사용 중단을 방지합니다. 에 대한 자세한 내용은 [AWS Health 사용 설명서를](#) Health Dashboard참조하세요. CloudTrail 이벤트에 대한 자세한 내용은 [CloudTrail 이벤트 기록 작업을](#) 참조하세요.

SSL 인증서 업데이트는 Snowball Edge 클라이언트를 통해 수행됩니다. 업데이트할 디바이스에 네트워크로 연결되어 있는 로컬 환경의 컴퓨터에 최신 버전의 Snowball Edge 클라이언트를 다운로드하여 설치해야 합니다. 자세한 내용은 [Snowball Edge 클라이언트 사용](#).

이 항목에서는 인증서 만료 시기를 결정하는 방법과 디바이스를 업데이트하는 방법에 대해 설명합니다.

- `snowballEdge describe-device-software` 명령을 사용하여 인증서 만료 시기를 결정하십시오. 명령 출력의 값에는 인증서가 만료되는 날짜 및 시간이 `CertificateExpiry` 값에 포함됩니다.

Example **describe-device-software** 출력

```
Installed version: 101
```

```
Installing version: 102
Install State: Downloading
CertificateExpiry : Thur Jan 01 00:00:00 UTC 1970
```

2. 에 문의하여 SSL 인증서 업데이트를 지원 요청합니다.
3. 지원 는 업데이트 파일을 제공합니다. 업데이트 파일을 [다운로드](#)하고 [설치](#)합니다.
4. [Snowball Edge 잠금 해제](#)를 수행할 때 새로운 잠금 해제 코드 및 매니페스트 파일을 사용합니다.

## Snowball Edge에서 Amazon Linux 2 AMIs 업데이트

보안을 위한 모범 사례로 Snowball Edge에서 Amazon Linux 2 AMIs up-to-date 유지하세요. 의 [Amazon Linux 2 AMI\(HVM\), SSD 볼륨 유형\(64비트 x86\)](#) AWS Marketplace 에서 업데이트를 정기적으로 확인합니다. AMI 업데이트의 필요성이 확인되면 최신 Amazon Linux 2 이미지를 Snow 디바이스로 가져옵니다. [Amazon EC2 호환 AMI로 디바이스로 이미지 가져오기](#)를 참조하십시오.

AWS CLI의 `ssm get-parameters` 명령을 사용하여 최신 Amazon Linux 2 이미지 ID를 가져올 수도 있습니다.

```
aws ssm get-parameters --names /aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2 --query 'Parameters[0].[Value]' --region your-region
```

이 명령은 AMI의 최신 이미지 ID를 반환합니다. 예시:

```
ami-0ccb473bada910e74
```



## Snowball Edge 작업 이해

의 작업은 콘솔 또는 작업 관리 API에서 작업을 생성할 때 정의되는 개별 작업 단위 AWS Snowball Edge입니다. AWS Snowball Edge 디바이스를 사용하면 세 가지 작업 유형이 있으며, 모두 로컬 스토리지 및 컴퓨팅 기능을 사용할 수 있습니다. 이 기능은 NFS 인터페이스 또는 Amazon S3 인터페이스를 사용하여 데이터를 읽고 씁니다. AWS Snowball Edge 디바이스에서 로컬로 실행되는 Amazon S3 PUT 객체 API 작업을 기반으로 Lambda 함수를 트리거합니다.

- [Snowball Edge 디바이스를 사용하여 Amazon S3로 데이터를 가져오는 작업](#) - 로컬 데이터 210TB 이하를 단일 디바이스로 복사한 다음 Amazon S3로 이전합니다. 가져오기 작업의 경우 Snowball 디바이스와 작업 간에 일대일 관계로 수행됩니다. 각 작업에는 디바이스가 정확히 하나씩 연결되어 있습니다. 더 많은 데이터를 가져와야 하는 경우 새 가져오기 작업을 생성하거나 기존 가져오기 작업을 복제할 수 있습니다. 이 작업 유형의 디바이스를 반환하면 디바이스에 있는 해당 데이터가 Amazon S3로 가져오기됩니다.
- [Snowball Edge 디바이스를 사용하여 Amazon S3에서 데이터를 내보내는 작업](#) - 원하는 양의 데이터 전송(Amazon S3에 위치)을 원하는 수의 Snowball Edge 디바이스로 복사한 다음 한 번에 하나의 AWS Snowball Edge 디바이스를 온프레미스 데이터 대상으로 이동했습니다. 내보내기 작업을 생성할 때 작업은 작업 파트로 분할됩니다. 각 작업 파트의 크기는 210TB 이하이며, 각 작업 파트에는 정확히 하나의 AWS Snowball Edge 디바이스가 연결되어 있습니다. 이 작업 유형의 디바이스를 반환하면 해당 디바이스는 삭제됩니다.
- [Snowball Edge 디바이스를 사용하여 로컬 컴퓨팅 및 스토리지 기능을 제공하는 방법에 대한 정보](#) - 이러한 작업에는 하나의 AWS Snowball Edge 디바이스 또는 클러스터에 사용되는 여러 디바이스가 포함됩니다. 이 작업은 내보내기 작업처럼 버킷에 있는 데이터로 시작되지 않으며 가져오기 작업처럼 종료할 때 데이터를 Amazon S3로 가져올 수 없습니다. 이 작업 유형의 디바이스를 반환하면 해당 디바이스는 삭제됩니다. 이 작업 유형을 사용하는 경우 디바이스 클러스터를 생성할 수 있는 옵션을 사용할 수 있습니다. 클러스터는 로컬 스토리지 내구성을 향상시키며 로컬 스토리지 용량을 확장하거나 축소할 수 있습니다.

Lambda를 사용할 수 없는 리전에서는 이 작업 유형을 로컬 스토리지 전용이라고 합니다.

## Snowball Edge 작업에 대한 세부 정보

작업을 생성하기 전에 [사전 조건](#)을 충족하는지 확인하세요. 각 작업은 생성할 때 지정된 세부 정보에 의해 정의됩니다. 다음 테이블에서는 작업에 대한 모든 세부 정보를 설명합니다.

콘솔 식별자	API 식별자	세부 정보 설명
작업 이름	Description	작업에 대한 이름이며 영숫자, 공백, 유니코드 특수 문자를 포함합니다.
작업 유형	JobType	가져오기, 내보내기, 로컬 컴퓨팅 및 스토리지를 비롯한 작업의 유형입니다.
작업 ID	JobId	작업을 식별하는 고유한 39자 라벨입니다. 작업 ID는 전자 잉크 디스플레이와 작업 매니페스트 파일의 이름에 표시되는 배송 라벨의 아래쪽에 표시됩니다.
Address	AddressId	다비이스를 배송할 주소입니다. API의 경우 주소 데이터 유형에 대한 ID입니다.
생성된 날짜	CreationDate	이 작업을 생성한 날짜입니다.
배송 기간	ShippingOption	기간 옵션은 리전별로 다릅니다. 자세한 내용은 <a href="#">Snowball Edge의 배송 속도</a> 단원을 참조하십시오.
IAM 역할 ARN	RoleARN	이 Amazon 리소스 이름(ARN)은 Amazon S3 버킷에 대한 쓰기 권한을 사용하여 작업을 생성하는 동안 생성되는 AWS Identity and Access Management (IAM) 역할입니다. 생성 프로세스는 자동이며 수임 AWS Snowball Edge 하도록 허용하는 IAM 역할은 S3 버킷과 Snowball 간에 데

콘솔 식별자	API 식별자	세부 정보 설명
		이터를 복사하는 데에만 사용 됩니다. 자세한 내용은 <a href="#">AWS Snowball Edge 콘솔을 사용하는 데 필요한 권한</a> 단원을 참조 하십시오.
AWS KMS 키	KmsKeyARN	에서 AWS Snowball Edge AWS Key Management Service (AWS KMS)는 각 Snowball의 키를 암호화합니다. 작업을 생성할 때 소유한 AWS KMS 암호화 키에 대한 ARN도 선택하거나 생성합니다. 자세한 내용은 <a href="#">AWS Key Management Service in AWS Snowball Edge</a> 단원을 참조 하십시오.
Snowball 용량	SnowballCapacityPreference	이 작업에서 주문한 AWS Snowball Edge 디바이스의 스토리지 용량입니다. 사용 가능한 크기에는 따라 다릅니다 AWS 리전.
스토리지 서비스	N/A	이 작업과 연결된 AWS 스토리지 서비스,이 경우 Amazon S3.
리소스	Resources	작업과 연결된 AWS 스토리지 서비스 리소스입니다. 이 경우 데이터를 주고 받을 Amazon S3 버킷입니다.
작업 유형	JobType	가져오기, 내보내기, 로컬 컴퓨팅 및 스토리지를 비롯한 작업의 유형입니다.

콘솔 식별자	API 식별자	세부 정보 설명
Snowball 유형	SnowballType	이 작업에서 주문한 Snowball Edge 디바이스의 유형입니다.
클러스터 ID	ClusterId	클러스터를 식별하는 고유한 39자 라벨입니다.

## Snowball Edge 작업의 상태

각 AWS Snowball Edge 디바이스 작업의 상태는 작업의 현재 상태를 나타내도록 변경됩니다. 이 작업 상태 정보는 상태, 현재 처리 상태, 연결된 디바이스에 사용되는 스토리지 등을 반영하지 않습니다.

### 작업 상태 보기

1. [AWS Snow 패밀리 관리 콘솔](#)에 로그인합니다.
2. 작업 대시보드에서 작업을 선택합니다.
3. 콘솔에서 작업 이름을 클릭합니다.
4. 작업 상태 창은 상단 근처에 있으며 작업 상태를 반영합니다.

### AWS Snowball Edge 디바이스 작업 상태

콘솔 식별자	API 식별자	상태 설명
작업 생성됨	New	작업이 방금 생성되었습니다. 작업이 내보내기 작업인 경우 이상 태인 동안에만 작업 또는 작업 파트를 취소할 수 있습니다.
어플라이언스 준비 중	PreparingAppliance	AWS 가 작업을 위한 디바이스를 준비하고 있습니다.
내보내는 중	InProgress	AWS 가 Amazon S3 에서 디바이스로 데이

콘솔 식별자	API 식별자	상태 설명
		터를 내보내고 있습니다.
배송 준비 중	PreparingShipment	AWS 가 디바이스를 배송할 준비를 하고 있습니다. 해당 상태에 있는 고객에게는 예상 배송 추적 정보가 제공 됩니다.
배송 중	InTransitToCustomer	작업 생성 중에 제공한 주소로 디바이스가 배송되었습니다.
사용자에게 배달됨	WithCustomer	작업 생성 중에 제공한 주소로 디바이스가 도착했습니다.
로 전송 중 AWS	InTransitToAWS	디바이스를 다시 배송했습니다 AWS.
분류 시설에 있음	WithAWSSortingFacility	이 작업의 디바이스는 내부 분류 시설에 있습니다. Amazon S3로 가져오기 작업에 대한 모든 추가 처리는 일반적으로 향후 2일 내에 시작됩니다.
의 AWS	WithAWS	배송물이 AWS에 도착했습니다. 데이터를 가져올 경우 가져오기는 일반적으로 도착 날짜 내에 시작됩니다.

콘솔 식별자	API 식별자	상태 설명
가져오기	InProgress	AWS 가 Amazon Simple Storage Service(Amazon S3) 로 데이터를 가져오고 있습니다.
완료됨	Complete	작업 또는 작업 파트를 성공적으로 완료했습니다.
취소됨	Cancelled	작업을 취소했습니다.

## Snowball Edge 클러스터 작업의 상태

각 클러스터에는 상태가 있습니다. 이 상태는 클러스터의 현재 일반 진행률 상태를 나타내도록 변경됩니다. 클러스터의 각 개별 노드에는 자체 작업 상태가 있습니다.

이 클러스터 상태 정보는 상태, 현재 처리 상태, 클러스터 또는 노드에 사용되는 스토리지 등을 반영하지 않습니다.

콘솔 식별자	API 식별자	상태 설명
Quorum 대기중	AwaitingQuorum	클러스터 요청을 처리하는 데 충분한 노드가 없어서 클러스터가 아직 생성되지 않았습니다. 클러스터를 생성하려면 최소한 다섯 개의 노드가 있어야 합니다.
보류중	Pending	클러스터가 생성되었으며 노드를 배송 준비하고 있습니다. 노드의 작업 상태를 통해 각

콘솔 식별자	API 식별자	상태 설명
		노드의 상태를 추적할 수 있습니다.
사용자에게 배달됨	InUse	하나 이상의 클러스터 노드가 작업 생성 중에 제공한 주소에 있습니다.
완료됨	Complete	클러스터의 모든 노드가 로 반환되었습니다 AWS.
취소됨	Cancelled	클러스터 생성 요청이 취소되었습니다. 클러스터 요청은 [Pending] 상태로 전환되기 이전에만 취소할 수 있습니다.

## Snowball Edge 디바이스를 사용하여 Amazon S3로 데이터를 가져오는 작업

가져오기 작업을 사용하면 데이터가 내장 Amazon S3 어댑터 또는 NFS 탑재 지점이 있는 AWS Snowball Edge 디바이스로 복사됩니다. 가져오기 작업에 대한 데이터 소스는 온프레미스에 있어야 합니다. 다시 말해서, 전송할 데이터가 들어 있는 스토리지 디바이스가 작업을 생성할 때 제공된 주소에 실제로 있어야 합니다.

파일을 가져올 때 각 파일은 Amazon S3에서 객체가 되고 각 디렉터리는 접두사가 됩니다. 데이터를 기존 버킷으로 가져올 경우 새로 가져온 객체와 동일한 이름을 가진 기존 객체를 덮어씁니다. 가져오기 작업 유형은 로컬 스토리지 및 컴퓨팅 기능을 수행할 수 있습니다. 이 기능은 NFS 인터페이스 또는 Amazon S3 어댑터를 사용하여 데이터를 읽고 쓰며 AWS Snowball Edge 디바이스에서 로컬로 실행되는 Amazon S3 PUT 객체 API 작업을 기반으로 Lambda 함수를 트리거합니다.

모든 데이터들의 지정된 Amazon S3 버킷으로 가져오면 AWS 클라우드 AWS 가 디바이스를 완전히 삭제합니다. 이 삭제는 NIST 800-88 표준에 따른 것입니다.

가져오기가 완료된 이후에 작업 보고서를 다운로드할 수 있습니다. 이 보고서는 가져오기 프로세스를 실패한 객체를 알려 줍니다. 성공 및 실패 로그에서 추가 정보를 확인할 수 있습니다.

#### Important

작업 완료 보고서의 결과를 확인하고 가져오기 로그를 검토할 수 있을 때까지 전송된 데이터의 로컬 복사본을 삭제하면 안 됩니다.

## Snowball Edge 디바이스를 사용하여 Amazon S3에서 데이터를 내보내는 작업

#### Note

태그와 메타데이터는 현재 지원되지 않습니다. 즉, S3 버킷에서 객체를 내보낼 때 모든 태그와 메타데이터가 제거됩니다.

내보내기 작업에 대한 데이터 소스는 하나 이상의 Amazon S3 버킷입니다. 작업 부분의 데이터가 Amazon S3에서 AWS Snowball Edge 디바이스로 이동되면 작업 보고서를 다운로드할 수 있습니다. 이 보고서는 디바이스로 전송하는 데 실패한 객체가 있으면 이에 대해 알려줍니다. 작업의 성공 및 실패 로그에서 자세한 정보를 확인할 수 있습니다.

전송을 완료하는 데 필요한 디바이스 수만큼 사용하여 각 내보내기 작업에 대해 원하는 수의 객체를 내보낼 수 있습니다. 내보내기 작업의 작업 파트를 위한 각 AWS Snowball Edge 디바이스는 차례로 배송되며, 이전 작업 파트가 전송 중 AWS 상태로 전환된 후 후속 디바이스가 배송됩니다.

Amazon S3 어댑터 또는 NFS 탑재 지점을 사용하여 디바이스에서 온프레미스 데이터 대상 주소로 객체를 복사할 경우 해당 객체는 파일로 저장됩니다. 파일이 이미 있는 위치에 객체를 복사할 경우 동일한 이름을 가진 기존 파일을 덮어씁니다. 내보내기 작업 유형은 로컬 스토리지 및 컴퓨팅 기능을 수행할 수 있습니다. 이 기능은 NFS 인터페이스 또는 Amazon S3 어댑터를 사용하여 데이터를 읽고 쓰며 AWS Snowball Edge 디바이스에서 로컬로 실행되는 Amazon S3 PUT 객체 API 작업을 기반으로 Lambda 함수를 트리거합니다.

가 반환된 디바이스를 AWS 수신하면 NIST 800-88 표준에 따라 완전히 지워집니다.



**⚠ Important**

Snow 디바이스로 내보내려는 데이터는 Amazon S3에 있어야 합니다. Snow 디바이스로 내보내 Amazon Glacier 려는의 모든 데이터는 S3 스토리지 클래스로 해동하거나 이동해야 내보낼 수 있습니다. Snow 내보내기 작업을 생성하기 전에 이 작업을 수행합니다. 전체 작업에 대한 모든 콘텐츠가 온프레미스 데이터 대상에 복사되었는지 확인할 수 있을 때까지 내보낸 Amazon S3 객체를 변경, 업데이트 또는 삭제하면 안 됩니다.

내보내기 작업을 생성할 때 전체 Amazon S3 버킷 또는 객체 키의 특정 범위를 내보낼 수 있습니다.

## Snowball Edge 디바이스로 데이터를 내보낼 때 Amazon S3 객체 키 사용

[AWS Snow 패밀리 관리 콘솔](#) 또는 작업 관리 API를 사용하여 내보내기 작업을 생성할 경우 전체 Amazon S3 버킷 또는 객체 키의 특정 범위를 내보낼 수 있습니다. 객체 키 이름은 버킷에서 객체를 고유하게 식별합니다. 범위를 내보낼 경우 포함 범위 시작, 포함 범위 종료 또는 둘 모두를 제공하여 범위의 길이를 정의합니다.

범위는 UTF-8 이진으로 정렬됩니다. UTF-8 이진 데이터는 다음과 같은 방법으로 정렬됩니다.

- 숫자 0~9는 영어 대/소문자보다 앞에 옵니다.
- 대문자 영어는 모든 소문자 영어보다 앞에 옵니다.
- 소문자 영어는 대문자 영어 및 숫자를 기준으로 정렬할 때 맨 마지막에 옵니다.
- 특수 문자는 다른 문자 집합을 따라 정렬됩니다.

특정 UTF-8에 대한 자세한 내용은 [UTF-8 on Wikipedia](#) 섹션을 참조하세요.

## Snowball Edge 디바이스로 데이터를 내보낼 때 Amazon S3 객체 키를 사용하는 예제

UTF-8 이진 순서로 정렬된 다음 객체 및 접두사가 버킷에 포함되어 있다고 가정합니다.

- 01
- Aardvark
- Aardwolf
- Aasvogel/apple
- Aasvogel/arrow/object1

- Aasvogel/arrow/object2
- Aasvogel/banana
- Aasvogel/banker/object1
- Aasvogel/banker/object2
- Aasvogel/cherry
- Banana
- Car

지정된 범위 시작	지정된 범위 종료	내보낼 범위 내 객체
(none)	(none)	버킷의 모든 객체
(none)	Aasvogel	01 Aardvark Aardwolf Aasvogel/apple Aasvogel/arrow/object1 Aasvogel/arrow/object2 Aasvogel/banana Aasvogel/banker/object1 Aasvogel/banker/object2 Aasvogel/cherry

지정된 범위 시작	지정된 범위 종료	내보낼 범위 내 객체
(none)	Aasvogel/banana	01  Aardvark  Aardwolf  Aasvogel/apple  Aasvogel/arrow/object1  Aasvogel/arrow/object2  Aasvogel/banana

지정된 범위 시작	지정된 범위 종료	내보낼 범위 내 객체
Aasvogel	(none)	Aasvogel/apple Aasvogel/arrow/object1 Aasvogel/arrow/object2 Aasvogel/banana Aasvogel/banker/object1 Aasvogel/banker/object2 Aasvogel/cherry Banana Car

지정된 범위 시작	지정된 범위 종료	내보낼 범위 내 객체
Aardwolf	(none)	Aardwolf Aasvogel/apple Aasvogel/arrow/object1 Aasvogel/arrow/object2 Aasvogel/banana Aasvogel/banker/object1 Aasvogel/banker/object2 Aasvogel/cherry Banana Car

지정된 범위 시작	지정된 범위 종료	내보낼 범위 내 객체
Aar	(none)	Aardvark Aardwolf Aasvogel/apple Aasvogel/arrow/object1 Aasvogel/arrow/object2 Aasvogel/banana Aasvogel/banker/object1 Aasvogel/banker/object2 Aasvogel/cherry Banana Car
car	(none)	객체가 내보내지지 않고 작업을 생성하려고 하면 오류 메시지가 표시됩니다. car는 UTF-8 이진 값에 따라 Car 아래에 정렬됩니다.

지정된 범위 시작	지정된 범위 종료	내보낼 범위 내 객체
Aar	Aarr	Aardvark Aardwolf
Aasvogel/arrow	Aasvogel/arrox	Aasvogel/arrow/ object1  Aasvogel/arrow/ object2
Aasvogel/apple	Aasvogel/banana	Aasvogel/apple  Aasvogel/arrow/ object1  Aasvogel/arrow/ object2  Aasvogel/ banana

지정된 범위 시작	지정된 범위 종료	내보낼 범위 내 객체
Aasvogel/apple	Aasvogel/banker	Aasvogel/apple Aasvogel/arrow/object1 Aasvogel/arrow/object2 Aasvogel/banana Aasvogel/banker/object1 Aasvogel/banker/object2
Aasvogel/apple	Aasvogel/cherry	Aasvogel/apple Aasvogel/arrow/object1 Aasvogel/arrow/object2 Aasvogel/banana Aasvogel/banker/object1 Aasvogel/banker/object2 Aasvogel/cherry



세 개의 버킷이 있고 folder2에 있는 모든 객체를 복사하려고 한다고 가정해 보겠습니다.

- s3://bucket/folder1/
- s3://bucket/folder2/
- s3://bucket/folder3/

지정된 범위 시작	지정된 범위 종료	내보낼 범위 내 객체
folder2/	folder2/	버킷 folder2 내 모든 객체

## Amazon S3에서 Snowball Edge 디바이스로 데이터를 내보내는 작업에 대한 모범 사례

- 데이터가 Amazon S3에 있는지 확인하고, 작업을 주문하기 전에 소용량 파일을 일괄 처리합니다.
- 버킷에 수백만 개의 객체가 있는 경우에는 내보내기 작업 정의에 키 범위를 지정해야 합니다.
- 이름에 슬래시(/ 또는 \)가 있는 객체는 Snowball Edge로 전송되지 않으므로 객체 키를 업데이트하여 이름에서 슬래시를 제거해야 합니다.
- S3 버킷의 경우 객체 길이 제한은 255자입니다.
- 버전 관리가 활성화된 S3 버킷의 경우 현재 버전의 객체만 내보냅니다.
- 삭제 마커는 내보내지 않습니다.

## Snowball Edge 디바이스를 사용하여 로컬 컴퓨팅 및 스토리지 기능을 제공하는 방법에 대한 정보

로컬 컴퓨팅 및 스토리지 작업을 사용하면 인터넷 연결 없이 로컬에서 Snowball Edge의 Amazon S3 호환 스토리지를 사용할 수 있습니다. 디바이스를 반환하면 Amazon S3에서 디바이스로 데이터를 내보내거나 Amazon S3로 데이터를 가져올 수 없습니다.

### 주제

- [Snowball Edge 디바이스에 로컬로 데이터를 저장하는 작업에 대한 정보](#)
- [Snowball Edge 디바이스 클러스터에 로컬 스토리지를 제공하는 작업에 대한 정보](#)

## Snowball Edge 디바이스에 로컬로 데이터를 저장하는 작업에 대한 정보

Snowball Edge의 Amazon S3 호환 스토리지 또는 S3 어댑터를 사용하여 AWS Snowball Edge 디바이스에 객체를 읽고 쓸 수 있습니다. 디바이스를 주문할 때 S3 어댑터를 사용하도록 선택하는 경우 디바이스를 받을 때 디바이스에 포함할 Amazon S3 버킷도 선택합니다. Snowball Edge에서 Amazon S3 호환 스토리지를 사용하도록 선택하면 디바이스를 수신할 때 디바이스에 Amazon S3 버킷이 포함되지 않습니다.

Snowball Edge 디바이스에 Amazon S3 버킷을 생성하여, 로컬 데이터 액세스, 로컬 데이터 처리, 데이터 레지던시가 필요한 애플리케이션에서 온프레미스로 객체를 쉽게 저장하고 검색할 수 있습니다. Snowball Edge의 Amazon S3 호환 스토리지는 Amazon S3 APIs를 사용하는 새로운 스토리지 클래스 SNOW인를 제공하며 여러 Snowball Edge 디바이스에 데이터를 내구성 있고 중복적으로 저장하도록 설계되었습니다. 버킷 수명 주기 정책, 암호화, 태그 지정을 포함하여 Amazon S3에서와 같이 Snowball Edge 버킷에서 동일한 API 및 기능을 사용할 수 있습니다. 디바이스 또는 디바이스가 반환되면 Snowball Edge의 Amazon S3 호환 스토리지에서 생성되거나 저장된 AWS모든 데이터가 삭제됩니다. 자세한 내용은 [로컬 컴퓨팅 및 스토리지 전용 작업](#) 섹션을 참조하세요.

자세한 내용은 이 설명서의 [Amazon S3 compatible storage on Snowball Edge](#)를 참조하세요.

디바이스 사용을 마치면 로 되돌리면 AWS디바이스가 지워집니다. 이 삭제는 NIST(National Institute of Standards and Technology) 800-88 표준에 따른 것입니다.

## Snowball Edge 디바이스 클러스터에 로컬 스토리지를 제공하는 작업에 대한 정보

클러스터는 3~16개 디바이스로 그룹화된 Snowball Edge 디바이스의 논리적 그룹입니다. 클러스터는 단일 작업으로 생성되므로 다른 AWS Snowball Edge 작업 제품과 비교할 때 향상된 내구성과 스토리지 크기를 제공합니다. 클러스터 작업에 대한 자세한 내용은 이 설명서의 [Clustering overview](#) 섹션을 참조하세요.

# Snowball Edge 디바이스 사용 모범 사례

AWS Snowball Edge 디바이스의 이점과 만족도를 극대화하려면 다음 모범 사례를 따르는 것이 좋습니다.

## Snowball Edge에 대한 보안 권장 사항

다음은 AWS Snowball Edge 디바이스로 작업하는 동안 보안을 유지하기 위한 권장 사항 및 모범 사례입니다.

### 일반 보안

- AWS Snowball Edge 디바이스에 대해 의심스러운 점이 발견되면 내부 네트워크에 연결하지 마십시오. 그 대신에 [AWS Support](#)에 연락하면 새 AWS Snowball Edge 디바이스를 받을 수 있습니다.
- 잠금 해제 코드의 복사본을 해당 작업에 대한 매니페스트와 동일한 워크스테이션 내 위치에 저장하지 않는 것이 좋습니다. 이를 서로 다른 위치에 저장하면 권한이 없는 당사자가 AWS Snowball Edge 디바이스에 액세스하는 것을 방지할 수 있습니다. 예를 들어 매니페스트 복사본을 로컬 서버에 저장하고 디바이스를 잠금 해제하려는 사용자에게 코드를 이메일로 전송할 수 있습니다. 이 접근 방식은 서버에 저장된 파일과 사용자의 이메일 주소에 액세스할 수 있는 개인으로 AWS Snowball Edge 디바이스에 대한 액세스를 제한합니다.
- Snowball Edge 클라이언트 명령 목록 액세스 키와 get-secret-access key를 실행할 때 표시되는 보안 인증 정보는 디바이스에 액세스하는 데 사용되는 액세스 키 페어입니다.

이러한 키는 작업과 디바이스의 로컬 리소스에만 연결됩니다. 사용자 AWS 계정 또는 다른 데 매핑되지 않습니다 AWS 계정. 이러한 키를 사용하여의 서비스 및 리소스에 액세스하려고 하면 작업과 연결된 로컬 리소스에서만 작동하므로 AWS 클라우드 실패합니다.

- 보안 인증 정보가 분실되거나 도용되었다고 생각되면 디바이스의 SSL 인증서를 업데이트하는 절차에 따라 새 매니페스트 파일과 잠금 해제 코드를 요청합니다. [Snowball Edge 디바이스에서 SSL 인증서 업데이트](#)을(를) 참조하세요.

AWS Identity and Access Management (IAM) 정책을 사용하여 액세스를 제어하는 방법에 대한 자세한 내용은 [섹션을 참조하세요 AWS에 대한 관리형\(미리 정의된\) 정책 AWS Snowball Edge](#).

### 네트워크 보안

- AWS Snowball Edge 디바이스의 로컬 버킷에 데이터를 읽고 쓰는 데 한 번에 한 가지 방법만 사용하는 것이 좋습니다.

- 데이터 손상을 방지하려면 데이터를 전송하는 동안 AWS Snowball Edge 디바이스의 연결을 끊거나 네트워크 설정을 변경하지 마십시오.
- 파일은 디바이스에 기록되는 동안 정적 상태에 있어야 합니다. 파일이 쓰여지는 동안 수정된 파일은 읽기/쓰기 충돌을 일으킬 수 있습니다.
- AWS Snowball Edge 디바이스 성능 향상에 대한 자세한 내용은 섹션을 참조하세요 [Snowball Edge와 최상의 데이터 전송 성능을 위한 권장 사항](#).

## Snowball Edge의 리소스 관리 모범 사례

AWS Snowball Edge 디바이스에서 작업 및 리소스를 관리하기 위한 다음 모범 사례를 고려해 보세요.

- 온프레미스 데이터 전송을 수행하는 15일 무료 기간은 AWS Snowball Edge 디바이스가 데이터 센터에 도착한 다음 날 시작됩니다. 이는 Snowball Edge 디바이스 유형에만 적용됩니다.
- [Job created] 상태에서만 작업을 취소할 수 있습니다. 작업이 다른 상태로 변경되면 작업을 취소할 수 없습니다. 이는 클러스터에도 적용됩니다.
- 가져오기 작업의 경우 Amazon S3로 가져오기가 성공할 때까지 전송된 데이터의 로컬 복사본을 삭제하면 안 됩니다. 프로세스의 일환으로 데이터 전송 결과를 확인하세요.

## Snowball Edge와 최상의 데이터 전송 성능을 위한 권장 사항

### Note

사용자가 경험하는 데이터 전송 성능은 네트워크 환경, 운영 체제, 복사 방법, 프로토콜, 소스 데이터 읽기 성능, 데이터 세트 특성(예: 파일 크기)에 따라 달라집니다. 정확한 데이터 전송 속도와 데이터 전송 시간을 결정하려면 해당 환경에서 개념 증명 테스트를 통해 성능을 측정하는 것이 좋습니다.

아래에서 AWS Snowball Edge 디바이스 성능에 대한 권장 사항 및 정보를 확인할 수 있습니다. 온프레미스 환경은 네트워크 기술, 하드웨어, 운영 체제, 절차 등이 다르기 때문에 이 섹션에서는 성능을 일반적인 용어로 설명합니다.

다음 테이블은 네트워크의 전송 속도가 Snowball Edge 디바이스를 데이터로 채우는 데 걸리는 시간에 어떻게 영향을 주는지를 간략하게 보여줍니다. 더 작은 파일들을 전송하면 증가한 오버헤드로 인해 전송 속도가 줄어듭니다. 작은 파일이 많은 경우 파일을 Snowball Edge 디바이스로 전송하기 전에 더 큰 아카이브로 압축하는 것이 좋습니다.

속도(MB/s)	82TB 전송 시간
800	1.22일
450	2.11일
400	2.37일
300	3.16일
277	3.42일
200	4.75일
100	9.49일
60	15.53일
30	31.06일
10	85.42일

성능에 대한 의미 있는 지침을 제공하기 위해 다음 섹션에서는 AWS Snowball Edge 디바이스를 사용하는 시기를 결정하는 방법과 서비스를 최대한 활용하는 방법을 설명합니다.

다음과 같은 관행을 따르는 것이 좋습니다. 이러한 사항은 데이터 전송의 성능 향상에 가장 큰 영향을 미칩니다.

- 각 디렉터리 내의 파일이나 디렉터리를 500,000개 이하로 유지하는 것이 좋습니다.
- Snowball Edge 디바이스로 전송되는 모든 파일은 크기가 1MB 이상인 것이 좋습니다.
- 1MB 미만의 작은 파일이 많은 경우 파일을 Snowball Edge 디바이스로 전송하기 전에 더 큰 아카이브로 압축하는 것이 좋습니다.

## Snowball Edge와의 데이터 전송 속도 개선

AWS Snowball Edge 디바이스의 성능을 개선할 수 있는 가장 좋은 방법 중 하나는 디바이스와 주고받는 데이터의 전송 속도를 높이는 것입니다. 일반적으로 다음과 같은 방법으로 데이터 소스에서 디바이

스로 전송하는 속도를 향상할 수 있습니다. 다음 목록에는 성능에 미치는 긍정적인 영향이 가장 큰 것부터 작은 것까지 순서대로 나열되어 있습니다.

1. 한 번에 여러 쓰기 작업 수행 - 이 작업을 수행하려면 단일 AWS Snowball Edge 디바이스에 네트워크로 연결된 컴퓨터의 여러 터미널 창에서 각 명령을 실행합니다.
2. 소용량 파일 배치 전송 - 암호화 때문에 각 복사 작업에는 일정한 오버헤드가 존재합니다. 프로세스 속도를 향상하려면 단일 아카이브에서 파일을 함께 배치 처리하는 것이 좋습니다. 파일을 함께 배치 처리할 때 Amazon S3로 파일을 가져올 경우 자동으로 압축이 풀릴 수 있습니다. 자세한 내용은 [Snowball Edge로의 데이터 전송 성능을 개선하기 위해 작은 파일 일괄 처리](#) 단원을 참조하십시오.
3. 전송 중에는 파일에 대해 다른 작업을 수행하지 않음 - 전송 중 파일 이름 바꾸기, 메타데이터 변경 또는 복사 작업 중 파일에 데이터 쓰기를 수행하면 전송 성능에 부정적인 영향이 미칠 수 있습니다. 파일을 전송하는 동안 파일을 정적 상태로 유지하는 것이 좋습니다.
4. 로컬 네트워크 사용 감소 - AWS Snowball Edge 디바이스는 로컬 네트워크에서 통신합니다. 이로 인해 AWS Snowball Edge 디바이스와 해당 디바이스가 연결되는 스위치, 그리고 데이터 소스를 호스팅하는 컴퓨터 사이에서 발생하는 기타 로컬 네트워크 트래픽을 줄이거나 제거할 경우 데이터 전송 속도가 개선될 수 있습니다.
5. 불필요한 홉 제거 - AWS Snowball Edge 디바이스, 데이터 소스 및 이들 간에 터미널 연결을 실행하는 컴퓨터를 설정하여 단일 스위치에서 통신하는 유일한 시스템이 되도록 하는 것이 좋습니다. 이를 통해 데이터 전송 속도가 개선될 수 있습니다.

# 에 대한 보안 AWS Snowball Edge

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 모델](#)은 이 사항을 클라우드 내 보안 및 클라우드의 보안으로 설명합니다.

- 클라우드 보안 - AWS 는에서 AWS 서비스를 실행하는 인프라를 보호할 책임이 있습니다 AWS 클라우드. AWS 또한는 안전하게 사용할 수 있는 서비스를 제공합니다. 서드 파티 감사원은 정기적으로 [AWS 규정 준수 프로그램](#)의 일환으로 보안 효과를 테스트하고 검증합니다. 에 적용되는 규정 준수 프로그램에 대한 자세한 내용은 [AWS 규정 준수 프로그램 제공 범위 내 서비스를](#) AWS Snowball Edge참조하세요.
- 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 설명서를 사용할 때 공동 책임 모델을 적용하는 방법을 이해하는 데 도움이 됩니다 AWS Snowball Edge. 다음 주제에서는 보안 및 규정 준수 목표를 충족하도록 AWS Snowball Edge 를 구성하는 방법을 보여줍니다. 또한 AWS Snowball Edge 리소스를 모니터링하고 보호하는 데 도움이 되는 다른 AWS 서비스를 사용하는 방법을 알아봅니다.

## 주제

- [AWS Snowball Edge Edge의 데이터 보호](#)
- [의 자격 증명 및 액세스 관리 AWS Snowball Edge](#)
- [AWS Snowball Edge의 로깅 및 모니터링](#)
- [에 대한 규정 준수 검증 AWS Snowball Edge](#)
- [복원성](#)
- [의 인프라 보안 AWS Snowball Edge](#)

## AWS Snowball Edge Edge의 데이터 보호

AWS Snowball Edge 는 데이터 보호에 대한 규정 및 지침을 포함하는 AWS [공동 책임 모델](#)을 준수합니다. AWS 는 모든 AWS 서비스를 실행하는 글로벌 인프라를 보호할 책임이 있습니다. AWS 는 고객

콘텐츠 및 개인 데이터를 처리하기 위한 보안 구성 제어를 포함하여 이 인프라에서 호스팅되는 데이터에 대한 제어를 유지합니다. 데이터 컨트롤러 또는 데이터 프로세서 역할을 하는 AWS 고객과 APN 파트너는 입력하는 모든 개인 데이터에 대한 책임이 있습니다 AWS 클라우드.

데이터 보호를 위해 자격 AWS 계정 증명을 보호하고 AWS Identity and Access Management (IAM)를 사용하여 개별 사용자를 설정하여 각 사용자에게 직무를 수행하는 데 필요한 권한만 부여하는 것이 좋습니다. 또한 다음과 같은 방법으로 데이터를 보호하는 것이 좋습니다.

- 각 계정에 다중 인증(MFA)을 사용하세요.
- SSL/TLS를 사용하여 AWS 리소스와 통신합니다. TLS 1.2 이상을 권장합니다.
- 를 사용하여 API 및 사용자 활동 로깅을 설정합니다 AWS CloudTrail.
- 서비스 내의 AWS 모든 기본 보안 제어와 함께 AWS 암호화 솔루션을 사용합니다.
- Amazon S3에 저장된 개인 데이터를 검색하고 보호하는 데 도움이 되는 Amazon Macie와 같은 고급 관리형 보안 서비스를 사용합니다.
- 명령줄 인터페이스 또는 API를 AWS 통해 액세스할 때 FIPS 140-2 검증 암호화 모듈이 필요한 경우 FIPS 엔드포인트를 사용합니다. 사용 가능한 FIPS 엔드포인트에 대한 자세한 내용은 [Federal Information Processing Standard\(FIPS\) 140-2](#) 섹션을 참조하세요.

명칭 필드와 같은 자유 형식 필드에 고객 계정 번호와 같은 중요 식별 정보를 절대 입력하지 마세요. 여기에는 콘솔 AWS CLI, API 또는 AWS SDKs를 사용하여 AWS Snowball Edge 또는 기타 AWS 서비스를 사용하는 경우가 포함됩니다. AWS Snowball Edge 또는 기타 서비스에 입력하는 모든 데이터는 진단 로그에 포함하기 위해 선택될 수 있습니다. 외부 서버에 URL을 제공할 때 해당 서버에 대한 요청을 검증하기 위해 자격 증명 정보를 URL에 포함시키지 마세요.

데이터 보호에 대한 자세한 내용은 AWS 보안 블로그의 [AWS 공동 책임 모델 및 GDPR](#) 블로그 게시물을 참조하세요.

## 주제

- [클라우드에서 데이터 보호](#)
- [디바이스에서 데이터 보호](#)

## 클라우드에서 데이터 보호

AWS Snowball Edge 는 Amazon S3로 데이터를 가져오거나 내보낼 때, Snowball Edge 디바이스 주문 작업을 생성할 때, 디바이스가 업데이트될 때 데이터를 보호합니다. 다음 섹션에서는 Snowball Edge를



사용하고 클라우드 AWS 에서 온라인 상태이거나와 상호 작용할 때 데이터를 보호하는 방법을 설명합니다.

주제

- [에 대한 암호화 AWS Snowball Edge](#)
- [AWS Key Management Service in AWS Snowball Edge](#)

## 에 대한 암호화 AWS Snowball Edge

Snowball Edge를 사용하여 S3으로 데이터를 가져올 경우 디바이스에 전송되는 모든 데이터는 네트워크를 통해 SSL 암호화로 보호됩니다. 유휴 데이터를 보호하기 위해 AWS Snowball Edge에서는 SSE(서버 측 암호화)를 사용합니다.

의 서버 측 암호화 AWS Snowball Edge

AWS Snowball Edge 는 Amazon S3 관리형 암호화 키(SSE-S3)를 사용한 서버 측 암호화를 지원합니다. 서버 측 암호화는 저장 데이터를 보호하기 위한 것이고, SSE-S3에서는 Amazon S3의 저장 데이터를 보호하기 위해 강력한 멀티 팩터 암호화를 제공합니다. SSE-S3에 대한 자세한 내용은 Amazon Simple Storage Service 사용 설명서의 [Protecting Data Using Server-Side Encryption with Amazon S3-Managed Encryption Keys \(SSE-S3\)](#) 섹션을 참조하세요.

현재 AWS Snowball Edge 는 고객 제공 키(SSE-C)를 사용한 서버 측 암호화를 제공하지 않습니다. Snowball Edge의 Amazon S3 호환 스토리지는 로컬 컴퓨팅 및 스토리지 작업을 위한 SSE-C를 제공합니다. 그러나 해당 SSE 유형을 사용하여 가져온 데이터를 보호하고 싶을 수 있으며, 내보내려는 데이터에 해당 SSE 유형을 이미 사용하고 있을 수 있습니다. 이 경우에는 다음 사항에 유의해야 합니다.

- 가져오기 -

Amazon S3로 가져온 객체를 SSE-C를 사용하여 암호화하려면, SSE-KMS 또는 SSE-S3 암호화를 버킷 정책의 일부로 설정된 것으로 간주해야 합니다. 하지만 Amazon S3로 가져온 객체를 SSE-C를 사용하여 암호화해야 하는 경우에는 버킷 내의 객체를 복사하여 SSE-C로 암호화해야 합니다. 이를 위한 샘플 CLI 명령은 다음과 같습니다.

```
aws s3 cp s3://amzn-s3-demo-bucket/object.txt s3://amzn-s3-demo-bucket/object.txt --sse-c --sse-c-key 1234567891SAMPLEKEY
```

or

```
aws s3 cp s3://amzn-s3-demo-bucket s3://amzn-s3-demo-bucket --sse-c --sse-c-key
1234567891SAMPLEKEY --recursive
```

- 내보내기 - SSE-C를 사용하여 암호화된 객체를 내보내려면, 먼저 서버 측 암호화를 제공하지 않거나 버킷 정책에 SSE-KMS 또는 SSE-S3가 지정되어 있는 다른 버킷으로 해당 객체를 복사합니다.

## Snowball Edge에서 Amazon S3로 가져온 데이터에 대해 SSE-S3 활성화

Amazon S3 관리 콘솔에서 다음 절차를 사용하여 Amazon S3로 가져오는 데이터에 대해 SSE-S3를 활성화합니다. AWS Snow 패밀리 관리 콘솔 또는 Snowball 디바이스 자체에서는 구성이 필요하지 않습니다.

Amazon S3로 가져오는 데이터에 대해 SSE-S3 암호화를 활성화하려면 데이터를 가져오는 모든 버킷의 버킷 정책을 설정하면 됩니다. 업로드 요청에 `x-amz-server-side-encryption` 헤더가 포함되어 있지 않은 경우 객체 업로드(`s3:PutObject`) 권한을 거부하도록 정책을 업데이트합니다.

### Amazon S3로 가져오는 데이터에 대해 SSE-S3 활성화

1. 에 로그인 AWS Management Console 하고 <https://console.aws.amazon.com/s3/> Amazon S3 콘솔을 엽니다.
2. 버킷 목록에서 데이터를 가져오려는 버킷을 선택합니다.
3. Permissions를 선택합니다.
4. [Bucket Policy]를 선택합니다.
5. [Bucket policy editor]에 다음 정책을 입력합니다. 이 정책에서 *YourBucket*의 모든 인스턴스를 실제 버킷 이름으로 바꿉니다.

### JSON

```
{
  "Version": "2012-10-17",
  "Id": "PutObjPolicy",
  "Statement": [
    {
      "Sid": "DenyIncorrectEncryptionHeader",
      "Effect": "Deny",
      "Principal": "*",
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::YourBucket/*",
```

```

    "Condition": {
      "StringNotEquals": {
        "s3:x-amz-server-side-encryption": "AES256"
      }
    },
    {
      "Sid": "DenyUnEncryptedObjectUploads",
      "Effect": "Deny",
      "Principal": "*",
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::YourBucket/*",
      "Condition": {
        "Null": {
          "s3:x-amz-server-side-encryption": "true"
        }
      }
    }
  ]
}

```

## 6. 저장을 선택합니다.

Amazon S3 버킷 구성을 완료했습니다. 이 버킷으로 가져오는 데이터가 SSE-S3로 보호됩니다. 필요에 따라 다른 버킷에 대해 이 절차를 반복합니다.

## AWS Key Management Service in AWS Snowball Edge

AWS Key Management Service (AWS KMS)는 데이터를 암호화하는 데 사용되는 암호화 키를 쉽게 생성하고 제어할 수 있는 관리형 서비스입니다.는 하드웨어 보안 모듈(HSMs)을 AWS KMS 사용하여 키의 보안을 보호합니다. 특히에서 작업에 대해 선택한 AWS KMS 키의 Amazon 리소스 이름(ARN) AWS Snowball Edge 은 KMS 키와 연결됩니다. 이 KMS 키는 작업에 대한 잠금 해제 코드를 암호화하는 데 사용됩니다. 잠금 해제 코드는 매니페스트 파일에서 암호화의 맨 위 계층을 해독하는 데 사용됩니다. 매니페스트 파일에 저장된 암호화 키는 디바이스에서 데이터를 암호화 및 암호화 해제하는 데 사용됩니다.

에서 AWS Snowball Edge는 각 AWS Snowball Edge 디바이스의 데이터를 AWS KMS 보호하는 데 사용되는 암호화 키를 보호합니다. 작업을 생성할 때 소유한 기존 KMS 키도 선택합니다. AWS KMS 키 AWS KMS keys 에 대한 ARN을 지정하면를 사용하여 AWS Snowball Edge 디바이스의 고유 키를 암호화 AWS Snowball Edge 할 수 있습니다. AWS Snowball Edge 지원되는 Amazon S3 server-side-encryption 옵션에 대한 자세한 내용은 섹션을 참조하세요 [의 서버 측 암호화 AWS Snowball Edge](#).

## Snowball Edge Edge AWS KMS keys 용 관리형 고객 사용

계정에 AWS KMS keys 대해 생성된 Snowball Edge Edge의 관리형 고객을 사용하려면 다음 단계를 따르세요.

### 작업에 대해 AWS KMS keys 선택

1. 에서 작업 생성을 AWS Snow 패밀리 관리 콘솔선택합니다.
2. 작업 유형을 선택하고 [Next]를 선택합니다.
3. 배송 세부 정보를 제공한 다음 [Next]를 선택합니다.
4. 작업의 세부 정보를 작성한 다음 [Next]를 선택합니다.
5. 보안 옵션을 설정합니다. 암호화에서 KMS 키에 AWS 관리형 키 대해 또는 이전에 생성한 사용자 지정 키를 선택하거나 별도의 계정이 소유한 키를 입력해야 하는 경우 키 ARN 입력을 AWS KMS 선택합니다.

#### Note

AWS KMS key ARN은 고객 관리형 키의 글로벌 고유 식별자입니다.

6. 다음을 선택하여 선택을 완료합니다 AWS KMS key.
7. Snow 디바이스 IAM 사용자에게 KMS 키에 대한 액세스 권한을 부여합니다.
  - a. IAM 콘솔(<https://console.aws.amazon.com/iam/>)에서 암호화 키로 이동하여 디바이스의 데이터를 암호화하는 데 사용하기로 선택한 KMS 키를 엽니다.
  - b. 키 사용자에서 추가를 선택하고 Snow 디바이스 IAM 사용자를 검색한 다음 연결을 선택합니다.

### 사용자 지정 KMS 봉투 암호화 키 생성

에서 사용자 지정 AWS KMS 봉투 암호화 키를 사용할 수 있습니다 AWS Snowball Edge. 자체 키를 생성할 경우 작업이 생성된 곳과 동일한 리전에서 키를 생성해야 합니다.

작업에 대한 자체 AWS KMS 키를 생성하려면 AWS Key Management Service 개발자 안내서의 [키 생성을 참조하세요](#).

## 디바이스에서 데이터 보호

## 보안 AWS Snowball Edge

다음은를 사용할 때 고려하는 몇 가지 보안 요소와 디바이스가 처리를 AWS 위해 도착할 때 취하는 다른 보안 예방 조치에 대한 AWS Snowball Edge 몇 가지 상위 수준 정보입니다.

다음 보안 접근 방식을 따르는 것이 좋습니다.

- 디바이스가 처음 도착하면 손상되거나 명백히 변경된 부분이 있는지 검사합니다. 디바이스에서 의심이 가는 부분이 있으면 내부 네트워크에 연결하지 마세요. 그 대신에 [AWS Support](#)에 연락하면 새 디바이스를 받을 수 있습니다.
- 작업 보안 인증 정보가 누출되지 않도록 보호해야 합니다. 작업의 매니페스트 및 잠금 해제 코드에 대한 액세스 권한이 있는 모든 개인은 해당 작업에 대해 전송되는 디바이스의 콘텐츠에 액세스할 수 있습니다.
- 디바이스를 로딩 독에 두지 마세요. 로딩 독에 방치하면 여러 요소에 노출될 수 있습니다. 각 AWS Snowball Edge 디바이스는 견고하지만 날씨로 인해 하드웨어의 가장 강력한 성능이 손상될 수 있습니다. 디바이스가 도난, 분실 또는 파손된 경우 최대한 빨리 보고하세요. 문제를 빨리 보고할수록 다른 디바이스를 더 빨리 배송받아 작업을 완료할 수 있습니다.

### Note

AWS Snowball Edge 디바이스는의 속성입니다 AWS. 디바이스를 조작하는 것은 AWS 허용 가능한 사용 정책을 위반하는 것입니다. 자세한 내용은 <http://aws.amazon.com/aup/>를 참조하세요.

다음 보안 단계를 수행합니다.

- Amazon S3 어댑터를 통해 데이터를 전송할 때 객체 메타데이터가 유지되지 않습니다. filename 및 filesize 메타데이터만 동일하게 유지됩니다. 다른 모든 메타데이터는 다음 예시와 같이 설정됩니다. -rw-rw-r-- 1 root root [filesize] Dec 31 1969 [path/filename]
- NFS 인터페이스를 사용하여 데이터를 전송할 때 객체 메타데이터가 유지됩니다.
- 디바이스가 도착하면 변조의 징후가 있는지 AWS 검사하고 신뢰할 수 있는 플랫폼 모듈(TPM)에서 변경 사항이 감지되지 않았는지 확인합니다.는 변조 방지 엔클로저, 256비트 암호화, 데이터에 대한 보안 및 전체 관리 체인을 모두 제공하도록 설계된 업계 표준 TPM 등 데이터를 보호하도록 설계된 여러 계층의 보안을 AWS Snowball Edge 사용합니다.
- 데이터 전송 작업이 처리 및 확인되면 AWS 는 미디어 삭제를 위한 NIST(National Institute of Standards and Technology) 지침에 따라 Snowball 디바이스에서 소프트웨어를 삭제합니다.

## NFC 태그 검증

Snowball Edge 컴퓨팅 최적화 및 Snowball Edge 스토리지 최적화(데이터 전송용) 디바이스에는 NFC 태그가 내장되어 있습니다. Android에서 사용할 수 있는 AWS Snowball Edge 확인 앱을 사용하여 이러한 태그를 스캔할 수 있습니다. 이러한 NFC 태그를 스캔하고 검증함으로써 디바이스 사용 전에 디바이스가 변조되지 않았는지 확인할 수 있습니다.

NFC 태그 검증 과정에는 Snowball Edge 클라이언트를 사용해 디바이스별 QR 코드를 생성하여 스캔 중인 태그가 올바른 디바이스에 부합하는지 확인하는 작업이 포함됩니다.

다음 절차는 Snowball Edge 디바이스에서 NFC 태그를 검증하는 방법을 설명합니다. 시작하기 전에, 다음과 같은 시작하기 실습의 첫 다섯 단계를 수행했는지 확인해야 합니다.

1. Snowball Edge 작업을 생성합니다. 자세한 내용은 [Snowball Edge 디바이스 주문 작업 생성을 참조하세요](#).
2. 디바이스를 수령합니다. 자세한 내용은 [Snowball Edge 받기](#) 단원을 참조하십시오.
3. 로컬 네트워크에 연결합니다. 자세한 내용은 [로컬 네트워크에 Snowball Edge 연결](#) 단원을 참조하십시오.
4. 보안 인증 정보 및 도구를 가져옵니다. 자세한 내용은 [Snowball Edge에 액세스하기 위한 자격 증명 가져오기](#) 단원을 참조하십시오.
5. Snowball Edge 클라이언트를 다운로드 및 설치합니다. 자세한 내용은 [Snowball Edge 클라이언트 다운로드 및 설치](#) 단원을 참조하십시오.

## NFC 태그 검증

1. `snowballEdge get-app-qr-code` Snowball Edge 클라이언트 명령을 실행합니다. 이 명령을 클러스터의 노드에 대해 실행하는 경우, 일련 번호(--device-sn)를 입력하여 단일 노드에 대한 QR 코드를 가져옵니다. 클러스터의 각 노드마다 이 단계를 반복합니다. 이 명령의 사용에 대한 자세한 내용은 [Snowball Edge NFC 태그를 검증하기 위한 QR 코드 가져오기](#) 섹션을 참조하세요.

QR 코드는 원하는 위치에 .png 파일로 저장됩니다.

2. 저장한 .png 파일로 이동하여 앱으로 QR 코드를 스캔할 수 있도록 엽니다.
3. Android의 AWS Snowball Edge 확인 앱을 사용하여 이러한 태그를 스캔할 수 있습니다.

**Note**

AWS Snowball Edge 확인 앱을 다운로드할 수 없지만 앱이 이미 설치된 디바이스가 있는 경우 앱을 사용할 수 있습니다.

## 4. 앱을 시작하고 화면의 지침을 따릅니다.

이제 디바이스에 대한 NFC 태그를 성공적으로 스캔했고 검증했습니다.

스캔 중 문제가 발생하는 경우, 다음을 시도합니다.

- 디바이스에 Snowball Edge 컴퓨팅 최적화 옵션이 있는지 확인합니다.
- 다른 디바이스에 앱이 설치되어 있는 경우에는 해당 디바이스를 사용합니다.
- 디바이스가 다른 NFC 태그의 간섭을 벗어나도록 방의 격리된 공간으로 이동하여 다시 시도합니다.
- 문제가 지속되면 [AWS Support](#)로 문의하세요.

## 의 자격 증명 및 액세스 관리 AWS Snowball Edge

모든 AWS Snowball Edge 작업은 인증되어야 합니다. 계정에서 IAM 사용자를 생성하고 관리하여 인증합니다. IAM을 사용하여 AWS에서 사용자 및 권한을 생성하고 관리할 수 있습니다.

AWS Snowball Edge 사용자가 작업을 생성하려면 AWS Snowball Edge AWS Management Console에 액세스하려면 특정 IAM 관련 권한이 있어야 합니다. 가져오기 또는 내보내기 작업을 생성하는 IAM 사용자는 작업에 사용할 Amazon S3 버킷, 리소스, Amazon SNS 주제 및 엣지 컴퓨팅 작업에 사용할 Amazon EC2-compatible AMI와 같은 올바른 Amazon Simple Storage Service(SAmazon S3) AWS KMS 리소스에도 액세스할 수 있어야 합니다.

**Important**

디바이스에서 로컬로 IAM 사용에 대한 자세한 내용은 [Snowball Edge에서 로컬로 IAM 사용](#) 섹션을 참조하세요.

## 주제

- [Snowball Edge 콘솔에 대한 액세스 제어 및 작업 생성](#)

## Snowball Edge 콘솔에 대한 액세스 제어 및 작업 생성

모든 AWS 서비스와 마찬가지로 액세스하려면 요청을 인증하는 데 사용할 AWS 수 있는 자격 증명에 AWS Snowball Edge 필요합니다. 이러한 자격 증명에는 두 가지 방법으로 Amazon S3 버킷 또는 AWS Lambda function. AWS Snowball Edge differs와 같은 AWS 리소스에 액세스할 수 있는 권한이 있어야 합니다.

1. 의 작업에는 Amazon 리소스 이름(ARNs)이 AWS Snowball Edge 없습니다.
2. 온프레미스에서 디바이스에 대한 물리 및 네트워크 액세스 제어는 사용자의 책임입니다.

[AWS Identity and Access Management\(IAM\)](#)를 사용하는 방법과 AWS Snowball Edge 에서 리소스에 액세스할 수 있는 사용자를 제어하여 리소스를 보호하는 방법에 [에 대한 자격 증명 및 액세스 관리 AWS Snowball Edge](#) 대한 자세한 내용과 AWS 클라우드로컬 액세스 제어 권장 사항을 알아보려면 섹션을 참조하세요.

### 에 대한 자격 증명 및 액세스 관리 AWS Snowball Edge

AWS Identity and Access Management (IAM)는 관리자가 AWS 리소스에 대한 액세스를 안전하게 제어하는 데 도움이 되는 서비스입니다. IAM 관리자는 누가 AWS Snow Family 리소스를 사용할 수 있는 인증(로그인) 및 권한(권한 있음)을 받을 수 있는지 제어합니다. IAM은 추가 비용 없이 사용할 수 있는 AWS 서비스입니다.

#### 주제

- [대상](#)
- [ID를 통한 인증](#)
- [정책을 사용하여 액세스 관리](#)
- [AWS Snow Family 에서 IAM을 사용하는 방법](#)
- [에 대한 자격 증명 기반 정책 예제 AWS Snowball Edge](#)
- [AWS Snowball Edge 자격 증명 및 액세스 문제 해결](#)

#### 대상

AWS Identity and Access Management (IAM)를 사용하는 방법은 수행하는 작업에 따라 다릅니다 AWS Snow Family.



서비스 사용자 - AWS Snow Family 서비스를 사용하여 작업을 수행하는 경우 필요한 자격 증명과 권한을 관리자가 제공합니다. 더 많은 AWS Snow Family 기능을 사용하여 작업을 수행하게 되면 추가 권한이 필요할 수 있습니다. 액세스 권한 관리 방법을 이해하면 관리자에게 올바른 권한을 요청하는 데 도움이 됩니다. AWS Snow Family의 기능에 액세스할 수 없는 경우 [AWS Snowball Edge 자격 증명 및 액세스 문제 해결](#)을 참조하세요.

서비스 관리자 - 회사에서 AWS Snow Family 리소스를 책임지고 있는 경우에 대한 전체 액세스 권한을 가지고 있을 것입니다 AWS Snow Family. 서비스 관리자는 서비스 사용자가 액세스해야 하는 AWS Snow Family 기능과 리소스를 결정합니다. 그런 다음 IAM 관리자에게 요청을 제출하여 서비스 사용자의 권한을 변경해야 합니다. 이 페이지의 정보를 검토하여 IAM의 기본 개념을 이해하세요. 회사에서 IAM을 사용하는 방법에 대한 자세한 내용은 섹션을 AWS Snow Family참조하세요 [AWS Snow Family에서 IAM을 사용하는 방법](#).

IAM 관리자 - IAM 관리자라면 AWS Snow Family에 대한 액세스 권한 관리 정책 작성 방법을 자세히 알고 싶을 것입니다. IAM에서 사용할 수 있는 자격 증명 기반 정책 예제를 보려면 섹션을 참조하세요 [에 대한 자격 증명 기반 정책 예제 AWS Snowball Edge](#).

## ID를 통한 인증

인증은 AWS 자격 증명으로써 로그인하는 방법입니다. IAM 사용자 또는 AWS 계정 루트 사용자 IAM 역할을 수임하여 로 인증(로그인 AWS)되어야 합니다.

자격 증명 소스를 통해 제공된 자격 증명을 사용하여 페더레이션 자격 증명 AWS 으로 로그인할 수 있습니다. AWS IAM Identity Center (IAM Identity Center) 사용자, 회사의 Single Sign-On 인증 및 Google 또는 Facebook 자격 증명은 페더레이션 자격 증명의 예입니다. 페더레이션형 ID로 로그인할 때 관리자가 이전에 IAM 역할을 사용하여 ID 페더레이션을 설정했습니다. 페더레이션을 사용하여 AWS 에 액세스하면 간접적으로 역할을 수임하게 됩니다.

사용자 유형에 따라 AWS Management Console 또는 AWS 액세스 포털에 로그인할 수 있습니다. 로그인에 대한 자세한 내용은 AWS Sign-In 사용 설명서의 로그인하는 방법을 AWS참조하세요. [AWS 계정](#)

AWS 프로그래밍 방식으로 액세스하는 경우는 자격 증명을 사용하여 요청에 암호화 방식으로 서명할 수 있는 소프트웨어 개발 키트(SDK)와 명령줄 인터페이스(CLI)를 AWS 제공합니다. AWS 도구를 사용하지 않는 경우 요청에 직접 서명해야 합니다. 권장 방법을 사용하여 요청에 직접 서명하는 자세한 방법은 IAM 사용 설명서에서 [API 요청용 AWS Signature Version 4](#)를 참조하세요.

사용하는 인증 방법에 상관없이 추가 보안 정보를 제공해야 할 수도 있습니다. 예를 들어는 멀티 팩터 인증(MFA)을 사용하여 계정의 보안을 강화할 것을 AWS 권장합니다. 자세한 내용은 AWS IAM Identity Center 사용 설명서에서 [다중 인증](#) 및 IAM 사용 설명서에서 [IAM의 AWS 다중 인증](#)을 참조하세요.

## AWS 계정 루트 사용자

를 생성할 때 계정의 모든 AWS 서비스 및 리소스에 대한 완전한 액세스 권한이 있는 하나의 로그인 자격 증명으로 AWS 계정시작합니다. 이 자격 증명을 AWS 계정 루트 사용자라고 하며 계정을 생성하는데 사용한 이메일 주소와 암호로 로그인하여 액세스합니다. 일상적인 작업에 루트 사용자를 사용하지 않을 것을 강력히 권장합니다. 루트 사용자 자격 증명을 보호하고 루트 사용자만 수행할 수 있는 작업을 수행하는 데 사용합니다. 루트 사용자로 로그인해야 하는 전체 작업 목록은 IAM 사용 설명서의 [루트 사용자 보안 인증이 필요한 작업](#)을 참조하세요.

## 페더레이션 자격 증명

가장 좋은 방법은 관리자 액세스가 필요한 사용자를 포함한 인간 사용자에게 자격 증명 공급자와의 페더레이션을 사용하여 임시 자격 증명을 사용하여 AWS 서비스에 액세스하도록 요구하는 것입니다.

페더레이션 자격 증명은 엔터프라이즈 사용자 디렉터리, 웹 자격 증명 공급자, Directory Service, Identity Center 디렉터리 또는 자격 증명 소스를 통해 제공된 자격 증명을 사용하여 AWS 서비스에 액세스하는 모든 사용자의 사용자입니다. 페더레이션 자격 증명에 액세스할 때 역할을 AWS 계정수입하고 역할은 임시 자격 증명을 제공합니다.

중앙 집중식 액세스 관리를 위해 AWS IAM Identity Center을(를) 사용하는 것이 좋습니다. IAM Identity Center에서 사용자 및 그룹을 생성하거나 모든 및 애플리케이션에서 사용할 수 있도록 자체 ID 소스의 사용자 AWS 계정 및 그룹 집합에 연결하고 동기화할 수 있습니다. IAM Identity Center에 대한 자세한 내용은 AWS IAM Identity Center 사용 설명서에서 [IAM Identity Center란 무엇인가요?](#)를 참조하세요.

## IAM 사용자 및 그룹

[IAM 사용자](#)는 단일 사용자 또는 애플리케이션에 대한 특정 권한이 AWS 계정 있는 내의 자격 증명입니다. 가능하면 암호 및 액세스 키와 같은 장기 자격 증명이 있는 IAM 사용자를 생성하는 대신 임시 자격 증명을 사용하는 것이 좋습니다. 하지만 IAM 사용자의 장기 자격 증명이 필요한 특정 사용 사례가 있는 경우, 액세스 키를 교체하는 것이 좋습니다. 자세한 내용은 IAM 사용 설명서의 [장기 보안 인증이 필요한 사용 사례의 경우, 정기적으로 액세스 키 교체](#)를 참조하세요.

[IAM 그룹](#)은 IAM 사용자 컬렉션을 지정하는 자격 증명입니다. 사용자는 그룹으로 로그인할 수 없습니다. 그룹을 사용하여 여러 사용자의 권한을 한 번에 지정할 수 있습니다. 그룹을 사용하면 대규모 사용자 집합의 권한을 더 쉽게 관리할 수 있습니다. 예를 들어, IAMAdmins라는 그룹이 있고 이 그룹에 IAM 리소스를 관리할 권한을 부여할 수 있습니다.

사용자는 역할과 다릅니다. 사용자는 한 사람 또는 애플리케이션과 고유하게 연결되지만, 역할은 해당 역할이 필요한 사람이라면 누구나 수입할 수 있습니다. 사용자는 영구적인 장기 자격 증명을 가지고 있

지만, 역할은 임시 보안 인증만 제공합니다. 자세한 내용은 IAM 사용 설명서에서 [IAM 사용자 사용 사례](#)를 참조하세요.

## IAM 역할

[IAM 역할](#)은 특정 권한이 AWS 계정 있는 내의 자격 증명입니다. IAM 사용자와 유사하지만, 특정 개인과 연결되지 않습니다. 에서 IAM 역할을 일시적으로 수입하려면 사용자에서 IAM 역할(콘솔)로 전환할 AWS Management Console수 있습니다. [https://docs.aws.amazon.com/IAM/latest/UserGuide/id\\_roles\\_use\\_switch-role-console.html](https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_use_switch-role-console.html) 또는 AWS API 작업을 호출하거나 사용자 지정 URL을 AWS CLI 사용하여 역할을 수입할 수 있습니다. 역할 사용 방법에 대한 자세한 내용은 IAM 사용 설명서의 [역할 수입 방법](#)을 참조하세요.

임시 보안 인증이 있는 IAM 역할은 다음과 같은 상황에서 유용합니다.

- 페더레이션 사용자 액세스 - 페더레이션 ID에 권한을 부여하려면 역할을 생성하고 해당 역할의 권한을 정의합니다. 페더레이션 ID가 인증되면 역할이 연결되고 역할에 정의된 권한이 부여됩니다. 페더레이션 관련 역할에 대한 자세한 내용은 IAM 사용 설명서의 [Create a role for a third-party identity provider \(federation\)](#)를 참조하세요. IAM Identity Center를 사용하는 경우, 권한 집합을 구성합니다. 인증 후 ID가 액세스할 수 있는 항목을 제어하기 위해 IAM Identity Center는 권한 집합을 IAM의 역할과 연관짓습니다. 권한 집합에 대한 자세한 내용은 AWS IAM Identity Center 사용 설명서의 [권한 집합](#)을 참조하세요.
- 임시 IAM 사용자 권한 - IAM 사용자 또는 역할은 IAM 역할을 수입하여 특정 작업에 대한 다양한 권한을 임시로 받을 수 있습니다.
- 교차 계정 액세스 - IAM 역할을 사용하여 다른 계정의 사용자(신뢰할 수 있는 보안 주체)가 내 계정의 리소스에 액세스하도록 허용할 수 있습니다. 역할은 계정 간 액세스를 부여하는 기본적인 방법입니다. 그러나 일부 에서는 정책을 리소스에 직접 연결할 AWS 서비스수 있습니다(역할을 프록시로 사용하는 대신). 교차 계정 액세스에 대한 역할과 리소스 기반 정책의 차이점을 알아보려면 IAM 사용 설명서의 [IAM의 교차 계정 리소스 액세스](#)를 참조하세요.
- 교차 서비스 액세스 - 일부는 다른에서 기능을 AWS 서비스 사용합니다 AWS 서비스. 예를 들어, 서비스에서 호출하면 일반적으로 해당 서비스는 Amazon EC2에서 애플리케이션을 실행하거나 Amazon S3에 객체를 저장합니다. 서비스는 직접적으로 호출하는 위탁자의 권한을 사용하거나, 서비스 역할을 사용하거나, 또는 서비스 연결 역할을 사용하여 이 작업을 수행할 수 있습니다.
- 전달 액세스 세션(FAS) - IAM 사용자 또는 역할을 사용하여에서 작업을 수행하는 경우 AWS보안 주체로 간주됩니다. 일부 서비스를 사용하는 경우, 다른 서비스에서 다른 작업을 시작하는 작업을 수행할 수 있습니다. FAS를 호출하는 보안 주체의 권한을 다운스트림 서비스에 AWS 서비스 대한 요청과 AWS 서비스함께 사용합니다. FAS 요청은 서비스가 완료하기 위해 다른 AWS 서비스 또는 리소스와의 상호 작용이 필요한 요청을 수신할 때만 이루어집니다. 이 경우, 두 작업을 모두

수행할 수 있는 권한이 있어야 합니다. FAS 요청 시 정책 세부 정보는 [전달 액세스 세션](#)을 참조하세요.

- 서비스 역할 - 서비스 역할은 서비스가 사용자를 대신하여 작업을 수행하기 위해 맡는 [IAM 역할](#)입니다. IAM 관리자는 IAM 내에서 서비스 역할을 생성, 수정 및 삭제할 수 있습니다. 자세한 정보는 IAM 사용 설명서의 [Create a role to delegate permissions to an AWS 서비스](#)를 참조하세요.
- 서비스 연결 역할 - 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 링크 역할의 권한을 볼 수 있지만 편집은 할 수 없습니다.
- Amazon EC2에서 실행되는 애플리케이션 - IAM 역할을 사용하여 EC2 인스턴스에서 실행되고 AWS CLI 또는 AWS API 요청을 수행하는 애플리케이션의 임시 자격 증명을 관리할 수 있습니다. 이는 EC2 인스턴스 내에 액세스 키를 저장할 때 권장되는 방법입니다. EC2 인스턴스에 AWS 역할을 할당하고 모든 애플리케이션에서 사용할 수 있도록 하려면 인스턴스에 연결된 인스턴스 프로파일을 생성합니다. 인스턴스 프로필에는 역할이 포함되어 있으며 EC2 인스턴스에서 실행되는 프로그램이 임시 보안 인증을 얻을 수 있습니다. 자세한 정보는 IAM 사용 설명서의 [IAM 역할을 사용하여 Amazon EC2 인스턴스에서 실행되는 애플리케이션에 권한 부여](#)를 참조하세요.

## 정책을 사용하여 액세스 관리

정책을 AWS 생성하고 자격 증명 또는 리소스에 연결하여 AWS 에서 액세스를 제어합니다. 정책은 자격 증명 또는 리소스와 연결된 AWS 경우 권한을 정의하는의 객체입니다.는 보안 주체(사용자, 루트 사용자 또는 역할 세션)가 요청할 때 이러한 정책을 AWS 평가합니다. 정책에서 권한은 요청이 허용되거나 거부되는 지를 결정합니다. 대부분의 정책은에 JSON 문서 AWS 로 저장됩니다. JSON 정책 문서의 구조와 콘텐츠에 대한 자세한 정보는 IAM 사용 설명서의 [JSON 정책 개요](#)를 참조하세요.

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.

기본적으로, 사용자 및 역할에는 어떠한 권한도 없습니다. 사용자에게 사용자가 필요한 리소스에서 작업을 수행할 권한을 부여하려면 IAM 관리자가 IAM 정책을 생성하면 됩니다. 그런 다음 관리자가 IAM 정책을 역할에 추가하고, 사용자가 역할을 수임할 수 있습니다.

IAM 정책은 작업을 수행하기 위해 사용하는 방법과 상관없이 작업에 대한 권한을 정의합니다. 예를 들어, iam:GetRole 작업을 허용하는 정책이 있다고 가정합니다. 해당 정책이 있는 사용자는 AWS Management Console AWS CLI, 또는 API에서 역할 정보를 가져올 수 있습니다 AWS .

## ID 기반 정책

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 정의](#)를 참조하세요.

ID 기반 정책은 인라인 정책 또는 관리형 정책으로 한층 더 분류할 수 있습니다. 인라인 정책은 단일 사용자, 그룹 또는 역할에 직접 포함됩니다. 관리형 정책은 여러 사용자, 그룹 및 역할에 연결할 수 있는 독립 실행형 정책입니다 AWS 계정. 관리형 정책에는 AWS 관리형 정책 및 고객 관리형 정책이 포함됩니다. 관리형 정책 또는 인라인 정책을 선택하는 방법을 알아보려면 IAM 사용 설명서의 [관리형 정책 및 인라인 정책 중에서 선택](#)을 참조하세요.

## 리소스 기반 정책

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 위탁자가 해당 리소스와 어떤 조건에서 어떤 작업을 수행할 수 있는지를 정의합니다. 리소스 기반 정책에서 [위탁자를 지정](#)해야 합니다. 보안 주체에는 계정, 사용자, 역할, 페더레이션 사용자 또는가 포함될 수 있습니다 AWS 서비스.

리소스 기반 정책은 해당 서비스에 있는 인라인 정책입니다. 리소스 기반 정책에서는 IAM의 AWS 관리형 정책을 사용할 수 없습니다.

## 액세스 제어 목록(ACL)

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 S3 AWS WAF 및 Amazon VPC는 ACLs. ACL에 관한 자세한 내용은 Amazon Simple Storage Service 개발자 가이드의 [액세스 제어 목록\(ACL\) 개요](#)를 참조하세요.

## 기타 정책 타입

AWS 는 덜 일반적인 추가 정책 유형을 지원합니다. 이러한 정책 타입은 더 일반적인 정책 유형에 따라 사용자에게 부여되는 최대 권한을 설정할 수 있습니다.

- 권한 경계 – 권한 경계는 ID 기반 정책에 따라 IAM 엔터티(IAM 사용자 또는 역할)에 부여할 수 있는 최대 권한을 설정하는 고급 기능입니다. 개체에 대한 권한 경계를 설정할 수 있습니다. 그 결과로 얻

는 권한은 객체의 ID 기반 정책과 그 권한 경계의 교집합입니다. Principal 필드에서 사용자나 역할을 지정하는 리소스 기반 정책은 권한 경계를 통해 제한되지 않습니다. 이러한 정책 중 하나에 포함된 명시적 거부는 허용을 재정의합니다. 권한 경계에 대한 자세한 정보는 IAM 사용 설명서의 [IAM 엔티티에 대한 권한 경계](#)를 참조하세요.

- 서비스 제어 정책(SCPs) - SCPs는 조직 또는 조직 단위(OU)에 대한 최대 권한을 지정하는 JSON 정책입니다 AWS Organizations. AWS Organizations 는 비즈니스가 소유한 여러 AWS 계정 를 그룹화하고 중앙에서 관리하기 위한 서비스입니다. 조직에서 모든 기능을 활성화할 경우, 서비스 제어 정책(SCP)을 임의의 또는 모든 계정에 적용할 수 있습니다. SCP는 각각을 포함하여 멤버 계정의 엔티티에 대한 권한을 제한합니다 AWS 계정 루트 사용자. 조직 및 SCP에 대한 자세한 내용은 AWS Organizations 사용 설명서에서 [Service control policies](#)을 참조하세요.
- 리소스 제어 정책(RCP) - RCP는 소유한 각 리소스에 연결된 IAM 정책을 업데이트하지 않고 계정의 리소스에 대해 사용 가능한 최대 권한을 설정하는 데 사용할 수 있는 JSON 정책입니다. RCP는 멤버 계정의 리소스에 대한 권한을 제한하며 조직에 속하는지 여부에 AWS 계정 루트 사용자관계없이 포함 자격 증명에 대한 유효 권한에 영향을 미칠 수 있습니다. RCP를 AWS 서비스 지원하는 목록을 포함하여 조직 및 RCPs에 대한 자세한 내용은 AWS Organizations 사용 설명서의 [리소스 제어 정책\(RCPs\)](#)을 참조하세요.
- 세션 정책 - 세션 정책은 역할 또는 페더레이션 사용자에게 대해 임시 세션을 프로그래밍 방식으로 생성할 때 파라미터로 전달하는 고급 정책입니다. 결과적으로 얻는 세션의 권한은 사용자 또는 역할의 ID 기반 정책의 교차와 세션 정책입니다. 또한 권한을 리소스 기반 정책에서 가져올 수도 있습니다. 이러한 정책 중 하나에 포함된 명시적 거부는 허용을 재정의합니다. 자세한 정보는 IAM 사용 설명서의 [세션 정책](#)을 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 로직](#)을 참조하세요.

## AWS Snow Family 에서 IAM을 사용하는 방법

IAM을 사용하여에 대한 액세스를 관리하기 전에 사용할 수 있는 IAM 기능에 대해 AWS Snow Family 알아봅니다 AWS Snow Family.



## 에서 사용할 수 있는 IAM 기능 AWS Snow Family

IAM 기능	AWS Snow Family 지원
<a href="#">ID 기반 정책</a>	예
<a href="#">리소스 기반 정책</a>	예
<a href="#">정책 작업</a>	예
<a href="#">정책 리소스</a>	예
<a href="#">정책 조건 키(서비스별)</a>	예
<a href="#">ACLs</a>	아니요
<a href="#">ABAC(정책 내 태그)</a>	부분
<a href="#">임시 자격 증명</a>	예
<a href="#">전달 액세스 세션(FAS)</a>	예
<a href="#">서비스 역할</a>	예
<a href="#">서비스 연결 역할</a>	아니요

AWS Snow Family 및 기타 AWS 서비스가 대부분의 IAM 기능과 작동하는 방법을 개괄적으로 알아보려면 IAM 사용 설명서의 [AWS IAM으로 작업하는 서비스를](#) 참조하세요.

## 에 대한 자격 증명 기반 정책 AWS Snow Family

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 정의](#)를 참조하세요.

IAM ID 기반 정책을 사용하면 허용되거나 거부되는 작업과 리소스뿐 아니라 작업이 허용되거나 거부되는 조건을 지정할 수 있습니다. ID 기반 정책에서는 위탁자가 연결된 사용자 또는 역할에 적용되므로

위탁자를 지정할 수 없습니다. JSON 정책에서 사용하는 모든 요소에 대해 알아보려면 IAM 사용 설명서의 [IAM JSON 정책 요소 참조](#)를 참조하세요.

에 대한 자격 증명 기반 정책 예제 AWS Snow Family

자격 AWS Snow Family 증명 기반 정책의 예를 보려면 섹션을 참조하세요 [에 대한 자격 증명 기반 정책 예제 AWS Snowball Edge](#).

내의 리소스 기반 정책 AWS Snow Family

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 위탁자가 해당 리소스와 어떤 조건에서 어떤 작업을 수행할 수 있는지를 정의합니다. 리소스 기반 정책에서 [위탁자를 지정](#)해야 합니다. 보안 주체에는 계정, 사용자, 역할, 페더레이션 사용자 또는 포함될 수 있습니다 AWS 서비스.

교차 계정 액세스를 활성화하려는 경우, 전체 계정이나 다른 계정의 IAM 개체를 리소스 기반 정책의 위탁자로 지정할 수 있습니다. 리소스 기반 정책에 크로스 계정 보안 주체를 추가하는 것은 트러스트 관계 설정의 절반밖에 되지 않는다는 것을 유념하세요. 보안 주체와 리소스가 다른 경우 신뢰할 수 있는 계정의 IAM 관리자는 보안 주체 엔터티(사용자 또는 역할)에게 리소스에 액세스할 수 있는 권한도 부여해야 합니다. 엔터티에 ID 기반 정책을 연결하여 권한을 부여합니다. 하지만 리소스 기반 정책이 동일 계정의 위탁자에 액세스를 부여하는 경우, 추가 자격 증명 기반 정책이 필요하지 않습니다. 자세한 내용은 IAM 사용 설명서의 [교차 계정 리소스 액세스](#)를 참조하세요.

에 대한 정책 작업 AWS Snow Family

정책 작업 지원: 예

관리자는 AWS JSON 정책을 사용하여 누가 무엇에 액세스할 수 있는지 지정할 수 있습니다. 즉, 어떤 위탁자가 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. 정책 작업은 일반적으로 연결된 AWS API 작업과 이름이 동일합니다. 일치하는 API 작업이 없는 권한 전용 작업 같은 몇 가지 예외도 있습니다. 정책에서 여러 작업이 필요한 몇 가지 작업도 있습니다. 이러한 추가 작업을 일컬어 종속 작업이라고 합니다.

연결된 작업을 수행할 수 있는 권한을 부여하기 위한 정책에 작업을 포함하세요.



AWS Snow Family 작업 목록을 보려면 서비스 승인 참조의에서 [정의한 작업을 AWS Snow Family](#) 참조하세요.

의 정책 작업은 작업 앞에 다음 접두사를 AWS Snow Family 사용합니다.

```
snowball
```

단일 문에서 여러 작업을 지정하려면 다음과 같이 쉼표로 구분합니다.

```
"Action": [
  "snowball:action1",
  "snowball:action2"
]
```

자격 AWS Snow Family 증명 기반 정책의 예를 보려면 섹션을 참조하세요 [에 대한 자격 증명 기반 정책 예제 AWS Snowball Edge](#).

에 대한 정책 리소스 AWS Snow Family

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. 문에는 Resource 또는 NotResource 요소가 반드시 추가되어야 합니다. 모범 사례에 따라 [Amazon 리소스 이름\(ARN\)](#)을 사용하여 리소스를 지정합니다. 리소스 수준 권한이라고 하는 특정 리소스 유형을 지원하는 작업에 대해 이를 수행할 수 있습니다.

작업 나열과 같이 리소스 수준 권한을 지원하지 않는 작업의 경우, 와일드카드(\*)를 사용하여 해당 문이 모든 리소스에 적용됨을 나타냅니다.

```
"Resource": "*"

```

AWS Snow Family 리소스 유형 및 해당 ARNs 목록을 보려면 서비스 승인 참조의에서 [정의한 리소스를 AWS Snow Family](#) 참조하세요. 각 리소스의 ARN을 지정할 수 있는 작업을 알아보려면 [AWS Snow Family가 정의한 작업](#)을 참조하십시오.

자격 AWS Snow Family 증명 기반 정책의 예를 보려면 섹션을 참조하세요 [에 대한 자격 증명 기반 정책 예제 AWS Snowball Edge](#).

에 대한 정책 조건 키 AWS Snow Family

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 요소(또는 Condition 블록)를 사용하면 정책이 발효되는 조건을 지정할 수 있습니다. Condition 요소는 옵션입니다. 같거나 작음과 같은 [조건 연산자](#)를 사용하여 정책의 조건을 요청의 값과 일치시키는 조건식을 생성할 수 있습니다.

한 문에서 여러 Condition 요소를 지정하거나 단일 Condition 요소에서 여러 키를 지정하는 경우, AWS 는 논리적 AND 작업을 사용하여 평가합니다. 단일 조건 키에 여러 값을 지정하는 경우는 논리적 OR 작업을 사용하여 조건을 AWS 평가합니다. 문의 권한을 부여하기 전에 모든 조건을 충족해야 합니다.

조건을 지정할 때 자리 표시자 변수를 사용할 수도 있습니다. 예를 들어, IAM 사용자에게 IAM 사용자 이름으로 태그가 지정된 경우에만 리소스에 액세스할 수 있는 권한을 부여할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 [IAM 정책 요소: 변수 및 태그](#)를 참조하세요.

AWS 는 전역 조건 키와 서비스별 조건 키를 지원합니다. 모든 AWS 전역 조건 키를 보려면 IAM 사용 설명서의 [AWS 전역 조건 컨텍스트 키](#)를 참조하세요.

AWS Snow Family 조건 키 목록을 보려면 서비스 승인 참조의 [에 대한 조건 키를 AWS Snow Family](#) 참조하세요. 조건 키를 사용할 수 있는 작업과 리소스를 알아보려면 [에서 정의한 작업을 AWS Snow Family](#) 참조하세요.

자격 AWS Snow Family 증명 기반 정책의 예를 보려면 섹션을 참조하세요 [에 대한 자격 증명 기반 정책 예제 AWS Snowball Edge](#).

ACLs AWS Snow Family

ACL 지원: 아니요

액세스 제어 목록(ACL)은 어떤 위탁자(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.

## 를 사용한 ABAC AWS Snow Family

### ABAC 지원(정책의 태그): 부분적

속성 기반 액세스 제어(ABAC)는 속성에 근거하여 권한을 정의하는 권한 부여 전략입니다. 여기서 AWS이러한 속성을 태그라고 합니다. IAM 엔터티(사용자 또는 역할) 및 많은 AWS 리소스에 태그를 연결할 수 있습니다. ABAC의 첫 번째 단계로 개체 및 리소스에 태그를 지정합니다. 그런 다음 위탁자의 태그가 액세스하려는 리소스의 태그와 일치할 때 작업을 허용하도록 ABAC 정책을 설계합니다.

ABAC는 빠르게 성장하는 환경에서 유용하며 정책 관리가 번거로운 상황에 도움이 됩니다.

태그에 근거하여 액세스를 제어하려면 `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` 또는 `aws:TagKeys` 조건 키를 사용하여 정책의 [조건 요소](#)에 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 정의](#)를 참조하세요. ABAC 설정 단계가 포함된 자습서를 보려면 IAM 사용 설명서의 [속성 기반 액세스 제어\(ABAC\) 사용](#)을 참조하세요.

### 에서 임시 자격 증명 사용 AWS Snow Family

#### 임시 자격 증명 지원: 예

일부 AWS 서비스는 임시 자격 증명을 사용하여 로그인할 때 작동하지 않습니다. 임시 자격 증명으로 AWS 서비스 작업을 포함하는 추가 정보는 [AWS 서비스 IAM 사용 설명서의 IAM으로 작업하는](#) 섹션을 참조하세요.

사용자 이름과 암호를 제외한 방법을 AWS Management Console 사용하여 로그인하는 경우 임시 자격 증명을 사용합니다. 예를 들어 회사의 SSO(Single Sign-On) 링크를 AWS 사용하여 액세스하면 해당 프로세스가 임시 자격 증명을 자동으로 생성합니다. 또한 콘솔에 사용자로 로그인한 다음 역할을 전환할 때 임시 자격 증명을 자동으로 생성합니다. 역할 전환에 대한 자세한 내용은 IAM 사용 설명서의 [사용자에서 IAM 역할로 전환\(콘솔\)](#)을 참조하세요.

AWS CLI 또는 AWS API를 사용하여 임시 자격 증명을 수동으로 생성할 수 있습니다. 그런 다음 이러한 임시 자격 증명을 사용하여 장기 액세스 키를 사용하는 대신 임시 자격 증명을 동적으로 생성하는 `access AWS`. `AWS recommends`에 액세스할 수 있습니다. 자세한 정보는 [IAM의 임시 보안 자격 증명](#) 섹션을 참조하세요.

### 에 대한 전달 액세스 세션 AWS Snow Family

#### 전달 액세스 세션(FAS) 지원: 예

IAM 사용자 또는 역할을 사용하여에서 작업을 수행하는 경우 AWS보안 주체로 간주됩니다. 일부 서비스를 사용하는 경우, 다른 서비스에서 다른 작업을 시작하는 작업을 수행할 수 있습니다. FAS는 호출하는 보안 주체의 권한을 다운스트림 서비스에 AWS 서비스 대한 요청과 AWS 서비스함께 사용합니다. FAS 요청은 서비스가 완료하기 위해 다른 AWS 서비스 또는 리소스와의 상호 작용이 필요한 요청을 수신할 때만 이루어집니다. 이 경우, 두 작업을 모두 수행할 수 있는 권한이 있어야 합니다. FAS 요청 시 정책 세부 정보는 [전달 액세스 세션](#)을 참조하세요.

## AWS Snow Family의 서비스 역할

서비스 역할 지원: 예

서비스 역할은 서비스가 사용자를 대신하여 작업을 수행하는 것으로 가정하는 [IAM 역할](#)입니다. IAM 관리자는 IAM 내에서 서비스 역할을 생성, 수정 및 삭제할 수 있습니다. 자세한 정보는 IAM 사용 설명서의 [Create a role to delegate permissions to an AWS 서비스](#)를 참조하세요.

### Warning

서비스 역할에 대한 권한을 변경하면 AWS Snow Family 기능이 중단될 수 있습니다. 에서 관련 지침을 AWS Snow Family 제공하는 경우에만 서비스 역할을 편집합니다.

## 에 대한 서비스 연결 역할 AWS Snow Family

서비스 링크 역할 지원: 아니요

서비스 연결 역할은에 연결된 서비스 역할의 한 유형입니다 AWS 서비스. 서비스는 사용자를 대신하여 작업을 수행하기 위해 역할을 수입할 수 있습니다. 서비스 연결 역할은에 표시 AWS 계정 되며 서비스가 소유합니다. IAM 관리자는 서비스 링크 역할의 권한을 볼 수 있지만 편집은 할 수 없습니다.

서비스 연결 역할 생성 또는 관리에 대한 자세한 내용은 [IAM으로 작업하는AWS 서비스](#)를 참조하세요. 서비스 연결 역할 열에서 Yes이(가) 포함된 서비스를 테이블에서 찾습니다. 해당 서비스에 대한 서비스 연결 역할 설명서를 보려면 예 링크를 선택합니다.

## 에 대한 자격 증명 기반 정책 예제 AWS Snowball Edge

기본적으로 사용자 및 역할에는 AWS Snow Family 리소스를 생성하거나 수정할 수 있는 권한이 없습니다. 또한 AWS Management Console, AWS Command Line Interface (AWS CLI) 또는 AWS API를 사용하여 작업을 수행할 수 없습니다. 사용자에게 사용자가 필요한 리소스에서 작업을 수행할 권한을 부여하려면 IAM 관리자가 IAM 정책을 생성하면 됩니다. 그런 다음 관리자가 IAM 정책을 역할에 추가하고, 사용자가 역할을 맡을 수 있습니다.

이러한 예제 JSON 정책 문서를 사용하여 IAM 자격 증명 기반 정책을 생성하는 방법을 알아보려면 IAM 사용 설명서의 [IAM 정책 생성\(콘솔\)](#)을 참조하세요.

각 리소스 유형에 대한 ARNs 형식을 포함하여 AWS Snow Family에서 정의한 작업 및 리소스 유형에 대한 자세한 내용은 서비스 승인 참조의 [에 대한 작업, 리소스 및 조건 키를 AWS Snow Family](#) 참조하세요.

## 주제

- [정책 모범 사례](#)
- [AWS Snow Family 콘솔 사용](#)
- [사용자가 자신의 고유한 권한을 볼 수 있도록 허용](#)

## 정책 모범 사례

자격 증명 기반 정책에 따라 계정에서 사용자가 AWS Snow Family 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 따릅니다.

- 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 정보는 IAM 사용 설명서의 [AWS 관리형 정책](#) 또는 [AWS 직무에 대한 관리형 정책](#)을 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 권한](#)을 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 특정을 통해 사용되는 경우 조건을 사용하여 서비스 작업에 대한 액세스 권한을 부여할 수도 있습니다 CloudFormation. 자세한 정보는 IAM 사용 설명서의 [IAM JSON 정책 요소: 조건](#)을 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 검증](#)을 참조하세요.

- 다중 인증(MFA) 필요 -에서 IAM 사용자 또는 루트 사용자가 필요한 시나리오가 있는 경우 추가 보안을 위해 MFA를 AWS 계정킵니다. API 작업을 직접 호출할 때 MFA가 필요하다면 정책에 MFA 조건을 추가합니다. 자세한 내용은 IAM 사용 설명서의 [MFA를 통한 보안 API 액세스](#)를 참조하세요.

IAM의 모범 사례에 대한 자세한 내용은 IAM 사용 설명서의 [IAM의 보안 모범 사례](#)를 참조하세요.

## AWS Snow Family 콘솔 사용

AWS Snow Family 콘솔에 액세스하려면 최소 권한 집합이 있어야 합니다. 이러한 권한은의 AWS Snow Family 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 AWS Snow Family 콘솔을 계속 사용할 수 있도록 하려면 AWS Snow Family *ConsoleAccess* 또는 *ReadOnly* AWS 관리형 정책도 엔티티에 연결합니다. 자세한 내용은 IAM 사용 설명서의 [사용자에게 권한 추가](#)를 참조하세요.

## 사용자가 자신의 고유한 권한을 볼 수 있도록 허용

이 예제는 IAM 사용자가 자신의 사용자 ID에 연결된 인라인 및 관리형 정책을 볼 수 있도록 허용하는 정책을 생성하는 방법을 보여줍니다. 이 정책에는 콘솔에서 또는 AWS CLI 또는 AWS API를 사용하여 프로그래밍 방식으로이 작업을 완료할 수 있는 권한이 포함됩니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ViewOwnUserInfo",
      "Effect": "Allow",
      "Action": [
        "iam:GetUserPolicy",
        "iam:ListGroupsWithUser",
        "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": "*"
    }
  ]
}

```

## AWS Snowball Edge 자격 증명 및 액세스 문제 해결

다음 정보를 사용하여 IAM으로 작업할 때 발생할 수 있는 일반적인 문제를 진단 AWS Snow Family 하고 수정할 수 있습니다.

### 주제

- [에서 작업을 수행할 권한이 없음 AWS Snow Family](#)
- [iam:PassRole을 수행하도록 인증되지 않음](#)
- [내 외부의 사람이 내 AWS Snow Family 리소스에 액세스 AWS 계정 하도록 허용하고 싶습니다.](#)

### 에서 작업을 수행할 권한이 없음 AWS Snow Family

작업을 수행할 권한이 없다는 오류가 수신되면, 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 *snowball:GetWidget* 권한이 없을 때 발생합니다.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
snowball:GetWidget on resource: my-example-widget
```

이 경우, `snowball: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:PassRole`을 수행하도록 인증되지 않음

`iam:PassRole` 작업을 수행할 수 있는 권한이 없다는 오류가 수신되면 AWS Snow Family에 역할을 전달할 수 있도록 정책을 업데이트해야 합니다.

일부 AWS 서비스에서는 새 서비스 역할 또는 서비스 연결 역할을 생성하는 대신 기존 역할을 해당 서비스에 전달할 수 있습니다. 이렇게 하려면 사용자가 서비스에 역할을 전달할 수 있는 권한을 가지고 있어야 합니다.

다음 예 오류는 `marymajor`라는 IAM 사용자가 콘솔을 사용하여 AWS Snow Family에서 작업을 수행하려고 하는 경우에 발생합니다. 하지만 작업을 수행하려면 서비스 역할이 부여한 권한이 서비스에 있어야 합니다. `Mary`는 서비스에 역할을 전달할 수 있는 권한을 가지고 있지 않습니다.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
iam:PassRole
```

이 경우, `Mary`가 `iam:PassRole` 작업을 수행할 수 있도록 `Mary`의 정책을 업데이트해야 합니다.

도움이 필요한 경우 AWS 관리자에게 문의하세요. 관리자는 로그인 자격 증명을 제공한 사람입니다.

내 외부의 사람이 내 AWS Snow Family 리소스에 액세스 AWS 계정 하도록 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.

자세히 알아보려면 다음을 참조하세요.

- 에서 이러한 기능을 AWS Snow Family 지원하는지 여부를 알아보려면 섹션을 참조하세요 [AWS Snow Family에서 IAM을 사용하는 방법](#).
- 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](#).
- 리소스에 대한 액세스 권한을 타사에 제공하는 방법을 알아보려면 IAM 사용 설명서의 [타사 AWS 계정 소유에 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 액세스](#)를 참조하세요.

## 의 액세스 제어 AWS 클라우드

AWS에서 요청을 인증하는 데 유효한 보안 인증 정보가 있을 수 있습니다. 그러나 권한이 없으면 AWS 리소스를 생성하거나 액세스할 수 없습니다. 예를 들어 Snowball Edge 디바이스를 주문하려면 작업을 생성할 수 있는 권한이 있어야 합니다.

다음 단원에서는 AWS Snowball Edge에 대한 클라우드 기반 권한을 관리하는 방법에 대해 설명합니다. 먼저 개요를 읽어 보면 도움이 됩니다.

- [의 리소스에 대한 액세스 권한 관리 개요 AWS 클라우드](#)
- [에 대한 자격 증명 기반 정책\(IAM 정책\) 사용 AWS Snowball Edge](#)

## 의 리소스에 대한 액세스 권한 관리 개요 AWS 클라우드

모든 AWS 리소스에서는 소유하며 AWS 계정, 리소스를 생성하거나 액세스할 수 있는 권한은 권한 정책에 의해 관리됩니다. 계정 관리자는 IAM 자격 증명(즉, 사용자, 그룹 및 역할)에 권한 정책을 연결할 수 있으며 일부 서비스(예: AWS Lambda)는 리소스에 권한 정책 연결을 지원합니다.

### Note

계정 관리자 또는 관리자 사용자는 관리자 권한이 있는 사용자입니다. 자세한 설명은 IAM 사용자 가이드의 [IAM 모범 사례](#) 섹션을 참조하십시오.

## 주제

- [리소스 및 작업](#)
- [리소스 소유권 이해](#)
- [에서 리소스에 대한 액세스 관리 AWS 클라우드](#)
- [정책 요소 지정: 작업, 효과, 보안 주체](#)
- [정책에서 조건 지정](#)

## 리소스 및 작업

에서 기본 리소스는 작업입니다. AWS Snowball Edge에는 Snowball 및 디바이스와 같은 AWS Snowball Edge 디바이스도 있지만 기존 작업의 컨텍스트에서만 해당 디바이스를 사용할 수 있습니다. Amazon S3 버킷과 Lambda 함수는 각각 Amazon S3와 Lambda의 리소스입니다.

앞에서 언급했듯이 작업에는 연결된 Amazon 리소스 이름(ARN)이 없습니다. 그러나 Amazon S3 버킷과 같은 다른 서비스 리소스에는 다음 테이블에 나와 있는 것처럼 고유한 이름(ARN)이 연결되어 있습니다.

AWS Snowball Edge는 작업을 생성하고 관리하기 위한 일련의 작업을 제공합니다. 사용 가능한 작업 목록은 [AWS Snowball Edge API Reference](#) 섹션을 참조하세요.

### 리소스 소유권 이해

는 리소스를 생성한 사용자에게 관계없이 계정에서 생성된 리소스를 AWS 계정 소유합니다. 특히 리소스 소유자는 리소스 생성 요청을 인증하는 AWS 계정 [보안 주체 엔터티](#)(즉, 루트 계정, IAM 사용자 또는 IAM 역할)의 입입니다. 다음의 예제에서는 이러한 작동 방법을 설명합니다.

- 의 AWS 계정 루트 계정 자격 증명을 사용하여 S3 버킷을 생성하는 경우 AWS 계정은 리소스의 소유자입니다(에서 AWS Snowball Edge 리소스는 작업임).
- 에서 IAM 사용자를 생성하고 해당 사용자에게 Snowball Edge 디바이스를 주문하는 작업을 생성할 수 있는 권한을 AWS 계정에 부여하는 경우 사용자는 Snowball Edge 디바이스를 주문하는 작업을 생성할 수 있습니다. 하지만 사용자가 속한 AWS 계정이 작업 리소스를 소유합니다.
- 작업을 생성할 권한이 AWS 계정에 있는 IAM 역할을 생성하는 경우 역할을 수임할 수 있는 사람은 누구나 Snowball Edge 디바이스를 주문하는 작업을 생성할 수 있습니다. 역할 AWS 계정이 속한 작업 리소스를 소유합니다.

### 에서 리소스에 대한 액세스 관리 AWS 클라우드

권한 정책은 누가 무엇에 액세스할 수 있는지를 나타냅니다. 다음 섹션에서는 권한 정책을 만드는 데 사용 가능한 옵션에 대해 설명합니다.

#### Note

이 섹션에서는 컨텍스트에서 IAM을 사용하는 방법에 대해 설명합니다 AWS Snowball Edge. IAM 서비스에 대한 자세한 정보는 다루지 않습니다. IAM 설명서 전체 내용은 IAM 사용 설명서의 [IAM이란 무엇입니까?](#) 섹션을 참조하세요. IAM 정책 구문과 설명에 대한 자세한 내용은 IAM 사용 설명서의 [AWS IAM 정책 참조](#) 섹션을 참조하세요.

IAM 자격 증명에 연결된 정책을 자격 증명 기반 정책(IAM 정책)이라고 하고 리소스에 연결된 정책을 리소스 기반 정책이라고 합니다.는 자격 증명 기반 정책(IAM 정책)만 AWS Snowball Edge 지원합니다.

주제

- [리소스 기반 정책](#)

## 리소스 기반 정책

Amazon S3과 같은 다른 서비스도 리소스 기반 권한 정책을 지원합니다. 예를 들어 정책을 S3 버킷에 연결하여 해당 버킷에 대한 액세스 권한을 관리할 수 있습니다. AWS Snowball Edge 는 리소스 기반 정책을 지원하지 않습니다.

정책 요소 지정: 작업, 효과, 보안 주체

각 작업([리소스 및 작업](#) 섹션 참조)에 대해 서비스는 해당 작업을 생성 및 관리하기 위한 일련의 API 작업([AWS Snowball Edge API Reference](#) 섹션 참조)을 정의합니다. 이러한 API 작업에 대한 권한을 부여하기 위해서는 정책에서 지정할 수 있는 일련의 작업을 AWS Snowball Edge 정의합니다. 예를 들어, 작업에 대해 CreateJob, CancelJob 및 DescribeJob 작업을 정의합니다. API 작업을 수행하려면 둘 이상의 작업에 대한 권한이 필요할 수 있습니다.

다음은 가장 기본적인 정책 요소입니다.

- 리소스 – 정책에서 Amazon 리소스 이름(ARN)을 사용하여 정책을 적용할 리소스를 식별합니다. 자세한 내용은 [리소스 및 작업](#) 단원을 참조하십시오.

### Note

이는 Amazon S3, Amazon EC2 AWS KMS, AWS Lambda 및 기타 여러 서비스에서 지원됩니다.

Snowball은 IAM 정책 설명의 Resource 요소에 리소스 ARN을 지정하는 기능을 지원하지 않습니다. Snowball에 대한 액세스를 허용하려면 정책에서 "Resource": "\*"를 지정합니다.

- 조치 – 조치 키워드를 사용하여 허용 또는 거부할 리소스 작업을 식별합니다. 예를 들어 지정한 Effect에 따라 snowball:\*는 모든 작업을 수행할 수 있는 사용자 권한을 허용하거나 거부합니다.

### Note

이는 Amazon EC2, Amazon S3, IAM에서 지원됩니다.

- 결과 – 사용자가 특정 작업을 요청하는 경우의 결과를 지정합니다. 이는 허용 또는 거부 중에 하나가 될 수 있습니다. 명시적으로 리소스에 대한 액세스 권한을 부여(허용)하지 않는 경우, 액세스는 묵시적으로 거부됩니다. 다른 정책에서 액세스 권한을 부여하는 경우라도 사용자가 해당 리소스에 액세스할 수 없도록 하기 위해 리소스에 대한 권한을 명시적으로 거부할 수도 있습니다.

#### Note

이는 Amazon EC2, Amazon S3, IAM에서 지원됩니다.

- Principal – 자격 증명 기반 정책(IAM 정책)에서 정책이 연결되는 사용자는 암시적인 보안 주체입니다. 리소스 기반 정책의 경우 권한을 받을 사용자, 계정, 서비스 또는 기타 엔터티를 지정합니다(리소스 기반 정책에만 적용). AWS Snowball Edge 는 리소스 기반 정책을 지원하지 않습니다.

IAM 정책 구문과 설명에 대한 자세한 내용은 IAM 사용 설명서의 [AWS IAM 정책 참조](#)를 참조하세요.

모든 AWS Snowball Edge API 작업을 보여주는 표는 섹션을 참조하세요 [AWS Snowball Edge API 권한: 작업, 리소스 및 조건 참조](#).

#### 정책에서 조건 지정

권한을 부여할 때 IAM 정책 언어를 사용하여 정책이 적용되는 조건을 지정할 수 있습니다. 예를 들어, 특정 날짜 이후에만 정책을 적용할 수 있습니다. 정책 언어에서의 조건 지정에 관한 자세한 내용은 IAM 사용 설명서의 [조건](#)을 참조하십시오.

조건을 표시하려면 미리 정의된 조건 키를 사용합니다. AWS Snowball Edge에만 해당되는 특정한 조건 키는 없습니다. 그러나 필요에 따라 사용할 수 있는 AWS전체 조건 키가 있습니다. AWS전체 키의 전체 목록은 IAM 사용 설명서의 [조건에 사용 가능한 키를 참조하세요](#).

#### 에 대한 자격 증명 기반 정책(IAM 정책) 사용 AWS Snowball Edge

이 섹션에서는 계정 관리자가 IAM 자격 증명(즉 사용자, 그룹, 역할)에 권한 정책을 연결하는 방법을 설명하는 자격 증명 기반 정책의 예시를 제시합니다. 따라서 이러한 정책은의 AWS Snowball Edge 리소스에 대한 작업을 수행할 수 있는 권한을 부여합니다 AWS 클라우드.

#### Important

AWS Snowball Edge 리소스에 대한 액세스 관리를 위해 제공되는 기본 개념과 옵션 설명이 나온 소개 주제 부분을 먼저 읽는 것이 좋습니다. 자세한 내용은 [의 리소스에 대한 액세스 권한 관리 개요 AWS 클라우드](#) 단원을 참조하십시오.

이 주제의 섹션에서는 다음 내용을 학습합니다.

- [AWS Snowball Edge 콘솔을 사용하는 데 필요한 권한](#)
- [AWS에 대한 관리형\(미리 정의된\) 정책 AWS Snowball Edge](#)
- [고객 관리형 정책 예](#)

다음은 권한 정책의 예입니다.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetBucketLocation",
        "s3:GetObject",
        "s3:ListBucket"
      ],
      "Resource": "arn:aws:s3:::*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "snowball:*",
        "importexport:*"
      ],
      "Resource": "*"
    }
  ]
}
```

이 정책에는 두 명령문이 있습니다:

- 첫 번째 문은 `arn:aws:s3:::*`의 Amazon 리소스 이름(ARN)을 사용하는 모든 Amazon S3 버킷에서 세 가지 Amazon S3 작업(`s3:GetBucketLocation`, `s3:GetObject` 및 `s3:ListBucket`)에 대한 권한을 부여합니다. 사용자가 데이터를 내보낼 모든 Amazon S3 버킷을 선택할 수 있도록 ARN은 와일드카드 문자(\*)를 지정합니다.

- 두 번째 문은 모든 AWS Snowball Edge 작업에 대한 권한을 부여합니다. 이 작업은 리소스 수준 권한을 지원하지 않으므로 정책은 와일드카드 문자(\*)를 지정하고 Resource 값도 와일드카드 문자를 지정합니다.

ID 기반 정책에서는 권한을 가질 보안 주체를 지정하지 않으므로 이 정책은 Principal 요소를 지정하지 않습니다. 정책을 사용자에게 연결할 경우 사용자는 암시적인 보안 주체입니다. IAM 역할에 권한 정책을 연결하면 역할의 신뢰 정책에서 식별된 보안 주체가 권한을 얻습니다.

모든 AWS Snowball Edge 작업 관리 API 작업과 해당 작업이 적용되는 리소스를 보여주는 표는 섹션을 참조하세요 [AWS Snowball Edge API 권한: 작업, 리소스 및 조건 참조](#).

AWS Snowball Edge 콘솔을 사용하는 데 필요한 권한

권한 참조 표에는 AWS Snowball Edge 작업 관리 API 작업이 나열되고 각 작업에 필요한 권한이 표시됩니다. 작업 관리 API 작업에 대한 자세한 내용은 [AWS Snowball Edge API 권한: 작업, 리소스 및 조건 참조](#) 섹션을 참조하세요.

를 사용하려면 다음 권한 정책에 표시된 대로 추가 작업에 대한 권한을 부여 AWS Snow 패밀리 관리 콘솔해야 합니다.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetBucketLocation",
        "s3:GetBucketPolicy",
        "s3:ListBucket",
        "s3:ListBucketMultipartUploads",
        "s3:ListAllMyBuckets"
      ],
      "Resource": "arn:aws:s3:::*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:CreateBucket",
        "s3:PutObject",

```

```
        "s3:AbortMultipartUpload",
        "s3:ListMultipartUploadParts",
        "s3:PutObjectAcl"
    ],
    "Resource": "arn:aws:s3:::*"
},
{
    "Effect": "Allow",
    "Action": [
        "lambda:GetFunction",
        "lambda:GetFunctionConfiguration"
    ],
    "Resource": "arn:aws:lambda:*::function:*"
},
{
    "Effect": "Allow",
    "Action": [
        "lambda:ListFunctions"
    ],
    "Resource": "*"
},
{
    "Effect": "Allow",
    "Action": [
        "kms:CreateGrant",
        "kms:GenerateDataKey",
        "kms:Decrypt",
        "kms:Encrypt",
        "kms:RetireGrant",
        "kms:ListKeys",
        "kms:DescribeKey",
        "kms:ListAliases"
    ],
    "Resource": [
        "*"
    ]
},
{
    "Effect": "Allow",
    "Action": [
        "iam:AttachRolePolicy",
        "iam:CreatePolicy",
        "iam:CreateRole",
        "iam:ListRoles",
```

```
        "iam:ListRolePolicies",
        "iam:PutRolePolicy"
    ],
    "Resource": [
        "*"
    ]
},
{
    "Effect": "Allow",
    "Action": "iam:PassRole",
    "Resource": "arn:aws:iam::*:role/snowball*",
    "Condition": {
        "StringEquals": {
            "iam:PassedToService": "importexport.amazonaws.com"
        }
    }
},
{
    "Effect": "Allow",
    "Action": [
        "ec2:DescribeImages",
        "ec2:ModifyImageAttribute"
    ],
    "Resource": [
        "*"
    ]
},
{
    "Effect": "Allow",
    "Action": [
        "sns:CreateTopic",
        "sns:ListTopics",
        "sns:GetTopicAttributes",
        "sns:SetTopicAttributes",
        "sns:ListSubscriptionsByTopic",
        "sns:Subscribe"
    ],
    "Resource": [
        "*"
    ]
},
{
    "Effect": "Allow",
    "Action": [
```



```

        "greengrass:getServiceRoleForAccount"
    ],
    "Resource": [
        "*"
    ]
},
{
    "Effect": "Allow",
    "Action": [
        "snowball:*"
    ],
    "Resource": [
        "*"
    ]
}
]
}

```

## JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetBucketLocation",
        "s3:GetBucketPolicy",
        "s3:ListBucket",
        "s3:ListBucketMultipartUploads",
        "s3:ListAllMyBuckets"
      ],
      "Resource": "arn:aws:s3:::*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:CreateBucket",
        "s3:PutObject",
        "s3:AbortMultipartUpload",
        "s3:ListMultipartUploadParts",
        "s3:PutObjectAcl"
      ],
      "Resource": "arn:aws:s3:::*"
    }
  ]
}

```

```
    ],
    "Resource": "arn:aws:s3:::*"
  },
  {
    "Effect": "Allow",
    "Action": [
      "lambda:GetFunction",
      "lambda:GetFunctionConfiguration"
    ],
    "Resource": "arn:aws:lambda:::function:*"
  },
  {
    "Effect": "Allow",
    "Action": [
      "lambda:ListFunctions"
    ],
    "Resource": "arn:aws:lambda:::*"
  },
  {
    "Effect": "Allow",
    "Action": [
      "iam:AttachRolePolicy",
      "iam:CreatePolicy",
      "iam:CreateRole",
      "iam:ListRoles",
      "iam:ListRolePolicies",
      "iam:PutRolePolicy"
    ],
    "Resource": [
      "*"
    ]
  },
  {
    "Effect": "Allow",
    "Action": "iam:PassRole",
    "Resource": "arn:aws:iam::*:role/snowball*",
    "Condition": {
      "StringEquals": {
        "iam:PassedToService": "importexport.amazonaws.com"
      }
    }
  },
  {
    "Effect": "Allow",
```

```
    "Action": [
      "ec2:DescribeImages",
      "ec2:ModifyImageAttribute"
    ],
    "Resource": [
      "*"
    ]
  },
  {
    "Effect": "Allow",
    "Action": [
      "sns:CreateTopic",
      "sns:ListTopics",
      "sns:GetTopicAttributes",
      "sns:SetTopicAttributes",
      "sns:ListSubscriptionsByTopic",
      "sns:Subscribe"
    ],
    "Resource": [
      "*"
    ]
  },
  {
    "Effect": "Allow",
    "Action": [
      "greengrass:getServiceRoleForAccount"
    ],
    "Resource": [
      "*"
    ]
  },
  {
    "Effect": "Allow",
    "Action": [
      "snowball:*"
    ],
    "Resource": [
      "*"
    ]
  }
]
```

AWS Snowball Edge 콘솔에는 다음과 같은 이유로 이러한 추가 권한이 필요합니다.

- **ec2:** - 사용자가 Amazon EC2 호환 인스턴스를 설명하고 로컬 컴퓨팅 목적으로 인스턴스의 속성을 수정할 수 있게 허용합니다. 자세한 내용은 [Snowball Edge에서 Amazon EC2-compatible 컴퓨팅 인스턴스 사용](#) 단원을 참조하십시오.
- **kms:** - 사용자가 데이터를 암호화할 KMS 키를 생성하거나 선택할 수 있게 허용합니다. 자세한 내용은 [AWS Key Management Service in AWS Snowball Edge](#) 단원을 참조하십시오.
- **iam:** - 이를 통해 사용자는 작업 생성 및 처리와 관련된 AWS 리소스에 액세스하기 위해 수입 AWS Snowball Edge 할 IAM 역할 ARN을 생성하거나 선택할 수 있습니다.
- **sns:** - 사용자가 생성되는 작업에 대한 Amazon SNS 알림을 생성하거나 선택할 수 있게 허용합니다. 자세한 내용은 [Snowball Edge에 대한 알림](#) 단원을 참조하십시오.

## AWS에 대한 관리형(미리 정의된) 정책 AWS Snowball Edge

AWS 는에서 생성하고 관리하는 독립 실행형 IAM 정책을 제공하여 많은 일반적인 사용 사례를 처리합니다. AWS. 관리형 정책은 사용자가 필요한 권한을 조사할 필요가 없도록 일반 사용 사례에 필요한 권한을 부여합니다. 자세한 내용은 IAM 사용 설명서의 [AWS 관리형 정책](#)을 참조하세요.

다음 AWS관리형 정책들과 함께 사용할 수 있습니다 AWS Snowball Edge.

### Snowball Edge에 대한 IAM 역할 정책 생성

IAM 역할 정책은 Amazon S3 버킷에 대한 읽기 및 쓰기 권한을 사용하여 생성해야 합니다. IAM 역할에는 Snowball Edge와 신뢰 관계도 있어야 합니다. 신뢰 관계가 있다는 것은가 데이터를 가져오는지 내 보내 AWS 는지에 따라 Snowball 및 Amazon S3 버킷에 데이터를 쓸 수 있음을 의미합니다.

에서 Snowball Edge 디바이스를 주문하는 작업을 생성할 때 필요한 IAM 역할 AWS Snow 패밀리 관리 콘솔생성은 권한 섹션의 4단계에서 수행됩니다. 이 프로세스는 자동으로 수행됩니다. Snowball Edge 가 수입하도록 허용하는 IAM 역할은 전송된 데이터가 있는 Snowball이 도착할 때만 버킷에 데이터를 쓰는 데 사용됩니다 AWS. 다음 절차에서는 이 프로세스를 간략하게 설명합니다.

가져오기 작업을 위한 IAM 역할을 생성하려면

1. 에 로그인 AWS Management Console 하고 <https://console.aws.amazon.com/importexport/> AWS Snowball Edge 콘솔을 엽니다.
2. 작업을 생성을 선택합니다.
3. 1단계에서 가져오기 작업의 세부 정보를 Amazon S3에 작성한 다음 다음을 선택합니다.
4. 2단계에서 [Permission]에서 [Create/Select IAM Role]을 선택합니다.

IAM Management Console이 열리며, AWS 가 지정된 Amazon S3 버킷에 객체를 복사하는 데 사용하는 IAM 역할을 보여줍니다.

- 이 페이지에서 세부 정보를 검토한 후 허용을 선택합니다.

로 돌아갑니다. AWS Snow 패밀리 관리 콘솔에서 선택한 IAM 역할 ARN에는 방금 생성한 IAM 역할에 대한 Amazon 리소스 이름(ARN)이 포함됩니다.

- 다음을 선택하여 IAM 역할 생성을 완료합니다.

이전 절차에서는 사용자가 데이터를 가져오려고 하는 Amazon S3 버킷에 대한 쓰기 권한이 있는 IAM 역할을 생성합니다. 생성되는 IAM 역할은 가져오기 작업을 위한 것인지 내보내기 작업을 위한 것인지 여부에 따라 다음 중 한 가지 구조를 가집니다.

가져오기 작업을 위한 IAM 역할

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetBucketLocation",
        "s3:ListBucketMultipartUploads"
      ],
      "Resource": "arn:aws:s3:::*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetBucketPolicy",
        "s3:PutObject",
        "s3:AbortMultipartUpload",
        "s3:ListMultipartUploadParts",
        "s3:PutObjectAcl",
        "s3:ListBucket"
      ],
      "Resource": "arn:aws:s3:::*"
    }
  ]
}
```

```
]
}
```

AWS KMS관리형 키를 사용한 서버 측 암호화(SSE-KMS)를 사용하여 가져오기 작업과 연결된 Amazon S3 버킷을 암호화하는 경우 IAM 역할에 다음 명령문도 추가해야 합니다.

```
{
  "Effect": "Allow",
  "Action": [
    "kms:GenerateDataKey"
  ],
  "Resource": "arn:aws:kms:us-west-2:123456789012:key/abc123a1-abcd-1234-efgh-111111111111"
}
```

객체 크기가 클 경우 가져오기 프로세스에 사용되는 Amazon S3 클라이언트는 멀티파트 업로드를 사용합니다. SSE-KMS를 사용하여 멀티파트 업로드를 시작하면 업로드된 모든 파트가 지정된 AWS KMS 키를 사용하여 암호화됩니다. 파트는 암호화되므로 멀티파트 업로드를 완료하기 위해 어셈블하려면 먼저 해독해야 합니다. 따라서 SSE-KMS를 사용하여 Amazon S3에 멀티파트 업로드를 실행할 때 AWS KMS 키(kms:Decrypt)를 복호화할 수 있는 권한이 있어야 합니다.

다음은 kms:Decrypt 권한이 필요한 가져오기 작업에 필요한 IAM 역할의 예입니다.

```
{
  "Effect": "Allow",
  "Action": [
    "kms:GenerateDataKey", "kms:Decrypt"
  ],
  "Resource": "arn:aws:kms:us-west-2:123456789012:key/abc123a1-abcd-1234-efgh-111111111111"
}
```

다음은 내보내기 작업에 필요한 IAM 역할의 예입니다.

JSON

```
{
```

```

"Version": "2012-10-17",
"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "s3:GetBucketLocation",
      "s3:GetBucketPolicy",
      "s3:GetObject",
      "s3:ListBucket"
    ],
    "Resource": "arn:aws:s3:::*"
  }
]
}

```

AWS KMS관리형 키로 서버 측 암호화를 사용하여 내보내기 작업과 연결된 Amazon S3 버킷을 암호화하는 경우 IAM 역할에 다음 명령문도 추가해야 합니다.

```

{
  "Effect": "Allow",
  "Action": [
    "kms:Decrypt"
  ],
  "Resource": "arn:aws:kms:us-west-2:123456789012:key/abc123a1-abcd-1234-efgh-111111111111"
}

```

사용자 지정 IAM 정책을 생성하여 AWS Snowball Edge 작업 관리를 위한 API 작업에 대한 권한을 허용할 수 있습니다. 해당 권한이 필요한 IAM 사용자 또는 그룹에 이러한 사용자 지정 정책을 연결할 수 있습니다.

## 고객 관리형 정책에

이 섹션에서는 다양한 AWS Snowball Edge 작업 관리 작업에 대한 권한을 부여하는 예제 사용자 정책을 찾을 수 있습니다. 이러한 정책은 AWS SDK 또는 AWS CLI를 사용하는 경우에 유효합니다. 콘솔을 사용하는 경우 [AWS Snowball Edge 콘솔을 사용하는 데 필요한 권한](#)의 설명과 같이 콘솔에 특정한 추가 권한을 부여해야 합니다.

**Note**

모든 예는 us-west-2 리전을 사용하며 가상의 계정 ID를 포함합니다.

**예시**

- [예제 1: 사용자가 API를 사용하여 Snowball Edge 디바이스를 주문하도록 작업을 생성하도록 허용하는 역할 정책](#)
- [예 2: 가져오기 작업 생성을 위한 역할 정책](#)
- [예 3: 내보내기 작업 생성을 위한 역할 정책](#)
- [예시 4: 예상 역할 권한 및 신뢰 정책](#)
- [AWS Snowball Edge API 권한: 작업, 리소스 및 조건 참조](#)

예제 1: 사용자가 API를 사용하여 Snowball Edge 디바이스를 주문하도록 작업을 생성하도록 허용하는 역할 정책

다음 권한 정책은 작업 관리 API를 사용하여 작업 또는 클러스터 생성 권한을 허용하는 데 사용되는 정책의 필수 구성 요소입니다. 이 문은 Snowball IAM 역할을 위한 신뢰 관계 정책 설명으로 필요합니다.

**JSON**

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "importexport.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```



## 예 2: 가져오기 작업 생성을 위한 역할 정책

AWS IoT Greengrass 함수로 AWS Lambda 구동되는를 사용하는 Snowball Edge에 대한 가져오기 작업을 생성하려면 다음 역할 신뢰 정책을 사용합니다.

JSON

```
    {
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetBucketLocation",
        "s3:ListBucketMultipartUploads"
      ],
      "Resource": "arn:aws:s3:::*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetBucketPolicy",
        "s3:GetBucketLocation",
        "s3:ListBucketMultipartUploads",
        "s3:ListBucket",
        "s3:PutObject",
        "s3:AbortMultipartUpload",
        "s3:ListMultipartUploadParts",
        "s3:PutObjectAcl",
        "s3:GetObject"
      ],
      "Resource": "arn:aws:s3:::*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "snowball:*"
      ],
      "Resource": [
        "*"
      ]
    }
  ],
}
```

```
{
  "Effect": "Allow",
  "Action": [
    "iot:AttachPrincipalPolicy",
    "iot:AttachThingPrincipal",
    "iot:CreateKeysAndCertificate",
    "iot:CreatePolicy",
    "iot:CreateThing",
    "iot:DescribeEndpoint",
    "iot:GetPolicy"
  ],
  "Resource": [
    "*"
  ]
},
{
  "Effect": "Allow",
  "Action": [
    "lambda:GetFunction"
  ],
  "Resource": [
    "*"
  ]
},
{
  "Effect": "Allow",
  "Action": [
    "greengrass:CreateCoreDefinition",
    "greengrass:CreateDeployment",
    "greengrass:CreateDeviceDefinition",
    "greengrass:CreateFunctionDefinition",
    "greengrass:CreateGroup",
    "greengrass:CreateGroupVersion",
    "greengrass:CreateLoggerDefinition",
    "greengrass:CreateSubscriptionDefinition",
    "greengrass:GetDeploymentStatus",
    "greengrass:UpdateGroupCertificateConfiguration",
    "greengrass:CreateGroupCertificateAuthority",
    "greengrass:GetGroupCertificateAuthority",
    "greengrass:ListGroupCertificateAuthorities",
    "greengrass:ListDeployments",
    "greengrass:GetGroup",
    "greengrass:GetGroupVersion",
    "greengrass:GetCoreDefinitionVersion"
  ]
}
```

```

    ],
    "Resource": [
        "*"
    ]
  }
]
}

```

### 예 3: 내보내기 작업 생성을 위한 역할 정책

AWS IoT Greengrass 함수로 AWS Lambda 구동되는를 사용하는 Snowball Edge에 대한 내보내기 작업을 생성하려면 다음 역할 신뢰 정책을 사용합니다.

#### JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetBucketLocation",
        "s3:GetObject",
        "s3:ListBucket"
      ],
      "Resource": "arn:aws:s3:::*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "snowball:*"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [

```

```

        "iot:AttachPrincipalPolicy",
        "iot:AttachThingPrincipal",
        "iot:CreateKeysAndCertificate",
        "iot:CreatePolicy",
        "iot:CreateThing",
        "iot:DescribeEndpoint",
        "iot:GetPolicy"
    ],
    "Resource": [
        "*"
    ]
},
{
    "Effect": "Allow",
    "Action": [
        "lambda:GetFunction"
    ],
    "Resource": [
        "*"
    ]
},
{
    "Effect": "Allow",
    "Action": [
        "greengrass:CreateCoreDefinition",
        "greengrass:CreateDeployment",
        "greengrass:CreateDeviceDefinition",
        "greengrass:CreateFunctionDefinition",
        "greengrass:CreateGroup",
        "greengrass:CreateGroupVersion",
        "greengrass:CreateLoggerDefinition",
        "greengrass:CreateSubscriptionDefinition",
        "greengrass:GetDeploymentStatus",
        "greengrass:UpdateGroupCertificateConfiguration",
        "greengrass:CreateGroupCertificateAuthority",
        "greengrass:GetGroupCertificateAuthority",
        "greengrass:ListGroupCertificateAuthorities",
        "greengrass:ListDeployments",
        "greengrass:GetGroup",
        "greengrass:GetGroupVersion",
        "greengrass:GetCoreDefinitionVersion"
    ],
    "Resource": [
        "*"
    ]
}

```

```

    ]
  }
]
}

```

#### 예시 4: 예상 역할 권한 및 신뢰 정책

기존 서비스 역할을 사용하려면 다음과 같은 예상 역할 권한 정책이 필요합니다. 이는 한 번만 설정할 수 있습니다.

#### JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "sns:Publish",
      "Resource": ["[[snsArn]]"]
    },
    {
      "Effect": "Allow",
      "Action": [
        "cloudwatch:ListMetrics",
        "cloudwatch:GetMetricData",
        "cloudwatch:PutMetricData"
      ],
      "Resource": [
        "*"
      ],
      "Condition": {
        "StringEquals": {
          "cloudwatch:namespace": "AWS/SnowFamily"
        }
      }
    }
  ]
}

```

}

기존 서비스 역할을 사용하려면 다음과 같은 예상 역할 신뢰 정책이 필요합니다. 이는 한 번만 설정할 수 있습니다.

## JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "importexport.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

AWS Snowball Edge API 권한: 작업, 리소스 및 조건 참조

[의 액세스 제어 AWS 클라우드](#)을 설정하고 IAM 자격 증명에 연결할 수 있는 권한 정책(자격 증명 기반 정책)을 작성할 때 다음 목록을 참조로 사용할 수 있습니다. 예는 각 AWS Snowball Edge 작업 관리 API 작업과 작업을 수행할 수 있는 권한을 부여할 수 있는 해당 작업이 나와 있습니다. 또한 각 API 작업에 대해 권한을 부여할 수 있는 AWS 리소스도 포함됩니다. 정책의 Action 필드에서 작업을 지정하고, 정책의 Resource 필드에서 리소스 값을 지정합니다.

AWS Snowball Edge 정책에서 AWS 전체 조건 키를 사용하여 조건을 표시할 수 있습니다. AWS 전체 키의 전체 목록은 IAM 사용 설명서의 [사용 가능한 키를 참조하세요](#).

### Note

작업을 지정하려면 snowball: 접두사 다음에 API 작업 명칭을 사용합니다(예: snowball:CreateJob).

# AWS Snowball Edge의 로깅 및 모니터링

모니터링은 AWS Snowball Edge 및 AWS 솔루션의 안정성, 가용성 및 성능을 유지하는 데 중요한 부분입니다. 다중 지점 장애가 발생할 경우 보다 쉽게 디버깅할 수 있도록 모니터링 데이터를 수집해야 합니다. AWS Snowball Edge 리소스를 모니터링하고 잠재적 인시던트에 대응하기 위한 몇 가지 도구를 AWS 제공합니다.

## AWS CloudTrail 로그

CloudTrail은 AWS Snowball Edge 작업 관리 API에서 또는 AWS 콘솔을 사용할 때 사용자, 역할 또는 AWS 서비스가 수행한 작업에 대한 레코드를 제공합니다. CloudTrail에서 수집한 정보를 사용하여 AWS Snowball Edge 서비스에 수행된 API 요청, 요청이 수행된 IP 주소, 요청을 수행한 사람, 요청이 수행된 시간 및 추가 세부 정보를 확인할 수 있습니다. 자세한 내용은 [클라우드 트레일 사용하여 AWS Snowball Edge API 호출 로깅 AWS CloudTrail](#) 단원을 참조하십시오.

## 규정 준수 검증 AWS Snowball Edge

AWS 서비스가 특정 규정 준수 프로그램의 범위 내에 있는지 알아보려면 [AWS 서비스 규정 준수 프로그램 제공 범위](#) 섹션을 참조하고 관심 있는 규정 준수 프로그램을 선택합니다. 일반 정보는 [AWS 규정 준수 프로그램](#).

[클라우드 트레일 사용하여 타사 감사 보고서를 다운로드할 수 있습니다 AWS Artifact](#). 자세한 내용은 [에서 보고서 다운로드 AWS Artifact](#)에서.

사용 시 규정 준수 책임은 데이터의 민감도, 회사의 규정 준수 목표 및 관련 법률과 규정에 따라 AWS 서비스 결정됩니다. 규정 준수를 지원하기 위해 다음 리소스를 AWS 제공합니다.

- [보안 규정 준수 및 거버넌스](#) - 이러한 솔루션 구현 가이드에서는 아키텍처 고려 사항을 설명하고 보안 및 규정 준수 기능을 배포하는 단계를 제공합니다.
- [HIPAA 적격 서비스 참조](#) - HIPAA 적격 서비스가 나열되어 있습니다. 모두 HIPAA 자격이 AWS 서비스 있는 것은 아닙니다.
- [AWS 규정 준수 리소스](#) - 이 워크북 및 가이드 모음은 산업 및 위치에 적용될 수 있습니다.
- [AWS 고객 규정 준수 가이드](#) - 규정 준수의 관점에서 공동 책임 모델을 이해합니다. 이 가이드에는 여러 프레임워크(미국 국립표준기술연구소(NIST), 결제 카드 산업 보안 표준 위원회(PCI), 국제표준화기구(ISO) 포함)의 보안 제어에 대한 지침을 보호하고 AWS 서비스 매핑하는 모범 사례가 요약되어 있습니다.

- AWS Config 개발자 안내서의 [규칙을 사용하여 리소스 평가](#) -이 AWS Config 서비스는 리소스 구성 이 내부 관행, 업계 지침 및 규정을 얼마나 잘 준수하는지 평가합니다.
- [AWS Security Hub CSPM](#) - 내 보안 상태에 대한 포괄적인 보기를 AWS 서비스 제공합니다 AWS. Security Hub는 보안 컨트롤을 사용하여 AWS 리소스를 평가하고 보안 업계 표준 및 모범 사례에 대한 규정 준수를 확인합니다. 지원되는 서비스 및 제어 목록은 [Security Hub 제어 참조](#)를 참조하세요.
- [Amazon GuardDuty](#) - 의심스러운 악의적인 활동이 있는지 환경을 모니터링하여 사용자, AWS 계정 워크로드, 컨테이너 및 데이터에 대한 잠재적 위협을 AWS 서비스 탐지합니다. GuardDuty는 특정 규정 준수 프레임워크에서 요구하는 침입 탐지 요구 사항을 충족하여 PCI DSS와 같은 다양한 규정 준수 요구 사항을 따르는 데 도움을 줄 수 있습니다.
- [AWS Audit Manager](#) - 이를 AWS 서비스 통해 AWS 사용량을 지속적으로 감사하여 위협과 규정 및 업계 표준 준수를 관리하는 방법을 간소화할 수 있습니다.

## 복원성

AWS 글로벌 인프라는 AWS 리전 및 가용 영역을 중심으로 구축됩니다.는 물리적으로 분리되고 격리된 여러 가용 영역을 AWS 리전 제공하며,이 가용 영역은 지연 시간이 짧고 처리량이 높으며 중복성이 높은 네트워킹과 연결됩니다. 가용 영역을 사용하면 중단 없이 가용 영역 간에 자동으로 장애 조치가 이루어지는 애플리케이션 및 데이터베이스를 설계하고 운영할 수 있습니다. 가용 영역은 기존의 단일 또는 복수 데이터 센터 인프라보다 가용성, 내결함성, 확장성이 뛰어납니다.

AWS 리전 및 가용 영역에 대한 자세한 내용은 [AWS 글로벌 인프라](#)를 참조하세요.

## 의 인프라 보안 AWS Snowball Edge

관리형 서비스인 AWS Snow Family 는 AWS 글로벌 네트워크 보안으로 보호됩니다. AWS 보안 서비스 및가 인프라를 AWS 보호하는 방법에 대한 자세한 내용은 [AWS 클라우드 보안을](#) 참조하세요. 인프라 보안 모범 사례를 사용하여 AWS 환경을 설계하려면 보안 원칙 AWS Well-Architected Framework의 [인프라 보호](#)를 참조하세요.

AWS 에서 게시한 API 호출을 사용하여 네트워크를 AWS Snow Family 통해 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.



또한 요청은 액세스 키 ID 및 IAM 위탁자와 관련된 보안 암호 액세스 키를 사용하여 서명해야 합니다. 또는 [AWS Security Token Service](#)(AWS STS)를 사용하여 임시 자격 증명을 생성하여 요청에 서명할 수 있습니다.

## Snowball Edge 디바이스로 전송된 데이터 검증

다음은가 데이터 전송을 AWS Snowball Edge 검증하는 방법과 작업 중 및 작업 후에 데이터 무결성을 보장하기 위해 수행할 수 있는 수동 단계에 대한 정보입니다.

Amazon S3 인터페이스를 사용하여 로컬 데이터 소스에서 Snowball Edge로 파일을 복사하면 여러 체크섬이 생성됩니다. 이러한 체크섬은 전송되는 데이터의 유효성을 자동으로 검사하는 데 사용됩니다.

일반적으로 이러한 체크섬은 파일별로(또는 큰 파일의 부분에 대해) 생성됩니다. Snowball Edge의 경우 디바이스의 버킷에 대해 다음 AWS CLI 명령을 실행하면 이러한 체크섬이 표시됩니다. 체크섬은 전송 전반에서 데이터의 무결성을 검증하는 데 사용되며, 데이터가 올바르게 복사되었는지 확인하는 데 도움이 됩니다.

```
aws s3api list-objects --bucket bucket-name --endpoint http://ip:8080 --profile edge-profile
```

이러한 체크섬이 일치하지 않는 경우 연결된 데이터를 Amazon S3로 가져오지 않습니다.

## 로컬 파일 인벤토리 및 Snowball Edge 데이터 전송

Amazon S3 어댑터 또는 CLI를 사용할 때 Snowball Edge에 복사된 파일의 로컬 인벤토리를 생성합니다. 로컬 인벤토리의 콘텐츠를 사용하여 로컬 스토리지 또는 서버의 내용과 비교할 수 있습니다.

예:

```
aws s3 cp folder/ s3://bucket --recursive > inventory.txt
```

## Snowball Edge에서 데이터 검증 오류의 일반적인 원인

유효성 검사 오류가 발생하면 해당 데이터(파일 또는 큰 파일의 일부)가 대상에 작성되지 않습니다. 유효성 검사 오류의 일반적인 원인은 다음과 같습니다.

- 심볼 링크를 복사하려는 경우.
- 수정 중인 파일을 복사하려는 경우. 이 시도는 체크섬 검증에 실패하고 전송 실패로 표시됩니다.
- 크기가 5TB보다 큰 파일을 복사하려는 경우.
- 크기가 2GiB보다 큰 부품 크기를 복사하려는 경우.

- 데이터 스토리지 용량이 이미 다 찬 Snowball Edge 디바이스로 파일을 복사하려는 경우
- Amazon S3에 대한 [객체 키 명명 지침](#)을 따르지 않는 Snowball Edge 디바이스로 파일을 복사하려는 경우

이러한 유효성 검사 오류 중 하나가 발생하면 해당 내역이 기록됩니다. 절차에 따라 유효성 검사에 실패한 파일과 그 이유를 수동으로 찾을 수 있습니다. 자세한 내용은 [Amazon S3로 가져온 후 Snowball Edge 디바이스에서 수동으로 데이터 검증](#) 단원을 참조하세요.

## Amazon S3로 가져온 후 Snowball Edge 디바이스에서 수동으로 데이터 검증

가져오기 작업이 완료되면 다음에 설명된 여러 옵션으로 Amazon S3의 데이터 유효성을 수동으로 검사할 수 있습니다.

### 작업 완료 보고서 및 연결된 로그 확인

Amazon S3에서 데이터를 가져오거나 내보낼 때마다 다운로드 가능한 PDF 작업 보고서가 생성됩니다. 가져오기 작업의 경우 가져오기 프로세스가 끝날 때 이 보고서를 사용할 수 있습니다. 자세한 내용은 [데이터 전송 작업 완료 보고서 및 로그 가져오기](#) 섹션을 참조하세요.

### S3 인벤토리

여러 작업에서 Amazon S3로 대량의 데이터를 전송한 경우, 각 작업 완료 보고서를 검토하게 되면 시간을 효율적으로 사용하지 못할 수 있습니다. 대신 하나 이상의 Amazon S3 버킷에 있는 모든 객체의 인벤토리를 가져올 수 있습니다. Amazon S3 인벤토리는 일일 또는 주간 기준으로 객체 및 해당 메타 데이터를 보여 주는 CSV(쉼표로 구분된 값) 파일을 제공합니다. 이 파일에는 Amazon S3 버킷 또는 공유 접두사의 객체(즉 이름이 공통 문자열로 시작하는 객체)가 포함됩니다.

데이터를 가져온 Amazon S3 버킷의 인벤토리가 있으면 이를 원본 데이터 위치에 전송한 파일과 손쉽게 비교할 수 있습니다. 이러한 방법으로 전송되지 않은 파일을 신속하게 파악할 수 있습니다.

Amazon S3 동기화 명령을 사용합니다.

워크스테이션이 인터넷에 연결할 수 있는 경우 AWS CLI 명령을 실행하여 전송된 모든 파일을 최종 검증할 수 있습니다. `aws s3 sync`. 이 명령은 디렉터리 및 S3 접두사를 동기화합니다. 이 명령은 소스 디렉터리의 새 파일 및 업데이트된 파일을 대상에 복사합니다. 자세한 내용은 AWS CLI 명령 레퍼런스의 [동기화](#) 섹션을 참조하세요.

**⚠ Important**

로컬 스토리지를 이 명령의 대상으로 지정한 경우, 동기화하는 파일의 백업이 있는지 확인하세요. 이러한 파일은 지정된 Amazon S3 소스의 콘텐츠로 덮어쓰기됩니다.





```

    "Service": "service.amazonaws.com"
  },
  "Action": [
    "kms:Decrypt",
    "kms:GenerateDataKey"
  ],
  "Resource": "*",
  "Condition": {
    "ArnLike": {
      "aws:SourceArn": "arn:aws:service:region:customer-account-id:resource-type/customer-resource-id"
    }
  },
  "StringEquals": {
    "kms:EncryptionContext:aws:sns:topicArn": "arn:aws:sns:your_region:customer-account-id:your_sns_topic_name"
  }
}
}
}

```

8. 변경 사항 저장을 선택하여 변경 내용을 적용하고 정책 편집기를 종료합니다.

## AWS Snow용 Amazon SNS 알림 예제

Amazon SNS 알림은 작업 상태가 변경될 때 다음과 같은 이메일 메시지를 생성합니다. 이러한 메시지는 Email-JSON SNS 주제 프로토콜의 예입니다.

작업 상태	SNS 알림 JSON
작업 생성됨	<pre> {   "Type" : "Notification",   "MessageId" : "dc1e94d9-56c5-5e96-808d-cc7f68faa162",   "TopicArn" : "arn:aws:sns:us-east-2:111122223333:ExampleTopic1",   "Message" : "Your job Job-name (JID8bca334a-6c2f-4cd0-97e2-3f5a4dc9bd6d) has been created. More info - https://console.aws.amazon.com/importexport", </pre>

작업 상태	SNS 알림 JSON
	<pre> "Timestamp" : "2023-02-23T00:27: 58.831Z",   "SignatureVersion" : "1",   "Signature" : "FMG5t1ZhJNHLHUXvZ gtZz1k24FzVa7oX0T4P03neeXw8 ZEXZx6z35j2F0TuNYShn2h0bKNC/ zLTnMyIxEzmi2X1sh0BWsJHkrW2xkR58ABZ F+4uWHEE73yDVR4SyYAIkP9jstZzDRm +bcVs8+T0yaLiEGLrIIIL4esi11lhIkg ErCuy5btPcWXBdio2fpCRD5x9oR 6gmE/rd5071X1c1uvnv4r1Lkk4pqP2/ iUfxFZva1xLSRvgyfm6D9hNk1VyPfy+7 Ta1MD01zmJu0rExtnSIbZew3foxgx8GT +1bZkLd0ZdtdRJ1IyPRP44eyq78sU0Eo/ LsDr0Iak4ZDpg8dXg==",   "SigningCertURL" : "https:// sns.us-east-1.amazonaws.com/ SimpleNotificationService-010a507c1 833636cd94bdb98bd93083a.pem",   "UnsubscribeURL" : "https:// sns.us-east-2.amazonaws.com/? Action=Unsubscribe&amp;SubscriptionArn =arn:aws:sns:us-east-2:1111 22223333:ExampleTopic1:e103 9402-24e7-40a3-a0d4-797da162b297" } </pre>



작업 상태	SNS 알림 JSON
어플라이언스 준비 중	<pre> {   "Type" : "Notification",   "MessageId" : "dc1e94d9-56c5-5e9 6-808d-cc7f68faa162",   "TopicArn" : "arn:aws:sns:us-ea st-2:111122223333:ExampleTopic1",   "Message" : "Your job Job-name (JID8bca334a-6c2f-4cd0-97e2 -3f5a4dc9bd6d) is being prepared. More info - https://console.aw s.amazon.com/importexport",   "Timestamp" : "2023-02-23T00:27: 58.831Z",   "SignatureVersion" : "1",   "Signature" : "FMG5t1ZhJNHLHUXvZ gtZz1k24FzVa7oX0T4P03neeXw8 ZEXZx6z35j2F0TuNYShn2h0bKNC/ zLTnMyIxEzmi2X1sh0BWsJHkrW2xkr58ABZ F+4uWHEE73yDVR4SyYAIkP9jstZzDRm +bcVs8+T0yaLiEGLrIIIL4esi111hIkG ErCuy5btPcWXBdio2fpCRD5x9oR 6gmE/rd5071X1c1uvnv4r1Lkk4pqP2/ iUfxFZva1xLSRvgyfm6D9hNk1VyPfy+7 Ta1MD01zmJu0rExtnSIbZew3foxgx8GT +1bZkLd0ZdtRj1IyPRP44eyq78sU0Eo/ LsDr0Iak4ZDpg8dXg==",   "SigningCertURL" : "https:// sns.us-east-1.amazonaws.com/ SimpleNotificationService-010a507c1 833636cd94bdb98bd93083a.pem",   "UnsubscribeURL" : "https:// sns.us-east-2.amazonaws.com/? Action=Unsubscribe&amp;SubscriptionArn =arn:aws:sns:us-east-2:1111 22223333:ExampleTopic1:e103 9402-24e7-40a3-a0d4-797da162b297" } </pre>

작업 상태	SNS 알림 JSON
내보내는 중	<pre> {   "Type" : "Notification",   "MessageId" : "dc1e94d9-56c5-5e9 6-808d-cc7f68faa162",   "TopicArn" : "arn:aws:sns:us-ea st-2:111122223333:ExampleTopic1",   "Message" : "Your job Job-name (JID8bca334a-6c2f-4cd0-97e2 -3f5a4dc9bd6d) is being Exported. More info - https://console.aw s.amazon.com/importexport",   "Timestamp" : "2023-02-23T00:27: 58.831Z",   "SignatureVersion" : "1",   "Signature" : "FMG5t1ZhJNHLHUXvZ gtZz1k24FzVa7oX0T4P03neeXw8 ZEXZx6z35j2F0TuNYShn2h0bKNC/ zLTnMyIxEzmi2X1sh0BWsJHkrW2xkr58ABZ F+4uWHEE73yDVR4SyYAIkP9jstZzDRm +bcVs8+T0yaLiEGLrIIIL4esi111hIkG ErCuy5btPcWXBdio2fpCRD5x9oR 6gmE/rd5071X1c1uvnv4r1Lkk4pqP2/ iUfxFZva1xLSRvgyfm6D9hNk1VyPfy+7 Ta1MD01zmJu0rExtnSIbZew3foxgx8GT +1bZkLd0ZdtRj1IyPRP44eyq78sU0Eo/ LsDr0Iak4ZDpg8dXg==",   "SigningCertURL" : "https:// sns.us-east-1.amazonaws.com/ SimpleNotificationService-010a507c1 833636cd94bdb98bd93083a.pem",   "UnsubscribeURL" : "https:// sns.us-east-2.amazonaws.com/? Action=Unsubscribe&amp;SubscriptionArn =arn:aws:sns:us-east-2:1111 22223333:ExampleTopic1:e103 9402-24e7-40a3-a0d4-797da162b297" } </pre>

작업 상태	SNS 알림 JSON
배송 중	<pre> {   "Type" : "Notification",   "MessageId" : "dc1e94d9-56c5-5e9 6-808d-cc7f68faa162",   "TopicArn" : "arn:aws:sns:us-ea st-2:111122223333:ExampleTopic1",   "Message" : "Your job Job-name (JID8bca334a-6c2f-4cd0-97e2 -3f5a4dc9bd6d) is in transit to you. More info - https://console.aw s.amazon.com/importexport",   "Timestamp" : "2023-02-23T00:27: 58.831Z",   "SignatureVersion" : "1",   "Signature" : "FMG5t1ZhJNHLHUXvZ gtZz1k24FzVa7oX0T4P03neeXw8 ZEXZx6z35j2F0TuNYShn2h0bKNC/ zLTnMyIxEzmi2X1sh0BWsJHkrW2xkr58ABZ F+4uWHEE73yDVR4SyYAIkP9jstZzDRm +bcVs8+T0yaLiEGLrIIIL4esi111hIkG ErCuy5btPcWXBdio2fpCRD5x9oR 6gmE/rd5071X1c1uvnv4r1Lkk4pqP2/ iUfxFZva1xLSRvgyfm6D9hNk1VyPfy+7 Ta1MD01zmJu0rExtnSIbZew3foxgx8GT +1bZkLd0ZdtRj1IyPRP44eyq78sU0Eo/ LsDr0Iak4ZDpg8dXg==",   "SigningCertURL" : "https:// sns.us-east-1.amazonaws.com/ SimpleNotificationService-010a507c1 833636cd94bdb98bd93083a.pem",   "UnsubscribeURL" : "https:// sns.us-east-2.amazonaws.com/? Action=Unsubscribe&amp;SubscriptionArn =arn:aws:sns:us-east-2:1111 22223333:ExampleTopic1:e103 9402-24e7-40a3-a0d4-797da162b297" } </pre>

작업 상태	SNS 알림 JSON
사용자에게 배달됨	<pre> {   "Type" : "Notification",   "MessageId" : "dc1e94d9-56c5-5e9 6-808d-cc7f68faa162",   "TopicArn" : "arn:aws:sns:us-ea st-2:111122223333:ExampleTopic1",   "Message" : "Your job Job-name (JID8bca334a-6c2f-4cd0-97e2 -3f5a4dc9bd6d) was delivered to you. More info - https://console.aw s.amazon.com/importexport",   "Timestamp" : "2023-02-23T00:27: 58.831Z",   "SignatureVersion" : "1",   "Signature" : "FMG5t1ZhJNHLHUXvZ gtZz1k24FzVa7oX0T4P03neeXw8 ZEXZx6z35j2F0TuNYShn2h0bKNC/ zLTnMyIxEzmi2X1sh0BWsJHkrW2xkR58ABZ F+4uWHEE73yDVR4SyYAIkP9jstZzDRm +bcVs8+T0yaLiEGLrIIIL4esi111hIkG ErCuy5btPcWXBdio2fpCRD5x9oR 6gmE/rd5071X1c1uvnv4r1Lkk4pqP2/ iUfxFZva1xLSRvgyfm6D9hNk1VyPfy+7 Ta1MD01zmJu0rExtnSIbZew3foxgx8GT +1bZkLd0ZdtRj1IyPRP44eyq78sU0Eo/ LsDr0Iak4ZDpg8dXg==",   "SigningCertURL" : "https:// sns.us-east-1.amazonaws.com/ SimpleNotificationService-010a507c1 833636cd94bdb98bd93083a.pem",   "UnsubscribeURL" : "https:// sns.us-east-2.amazonaws.com/? Action=Unsubscribe&amp;SubscriptionArn =arn:aws:sns:us-east-2:1111 22223333:ExampleTopic1:e103 9402-24e7-40a3-a0d4-797da162b297" } </pre>

작업 상태	SNS 알림 JSON
로 전송 중 AWS	<pre> {   "Type" : "Notification",   "MessageId" : "dc1e94d9-56c5-5e9 6-808d-cc7f68faa162",   "TopicArn" : "arn:aws:sns:us-ea st-2:111122223333:ExampleTopic1",   "Message" : "Your job Job-name (JID8bca334a-6c2f-4cd0-97e2 -3f5a4dc9bd6d) is in transit to AWS. More info - https://console.aw s.amazon.com/importexport",   "Timestamp" : "2023-02-23T00:27: 58.831Z",   "SignatureVersion" : "1",   "Signature" : "FMG5t1ZhJNHLHUXvZ gtZz1k24FzVa7oX0T4P03neeXw8 ZEXZx6z35j2F0TuNYShn2h0bKNC/ zLTnMyIxEzmi2X1sh0BWsJHkrW2xkR58ABZ F+4uWHEE73yDVR4SyYAIkP9jstZzDRm +bcVs8+T0yaLiEGLrIIIL4esi111hIkG ErCuy5btPcWXBdio2fpCRD5x9oR 6gmE/rd5071X1c1uvnv4r1Lkk4pqP2/ iUfxFZva1xLSRvgyfm6D9hNk1VyPfy+7 Ta1MD01zmJu0rExtnSIbZew3foxgx8GT +1bZkLd0ZdtRj1IyPRP44eyq78sU0Eo/ LsDr0Iak4ZDpg8dXg==",   "SigningCertURL" : "https:// sns.us-east-1.amazonaws.com/ SimpleNotificationService-010a507c1 833636cd94bdb98bd93083a.pem",   "UnsubscribeURL" : "https:// sns.us-east-2.amazonaws.com/? Action=Unsubscribe&amp;SubscriptionArn =arn:aws:sns:us-east-2:1111 22223333:ExampleTopic1:e103 9402-24e7-40a3-a0d4-797da162b297" } </pre>

## 작업 상태

## SNS 알림 JSON

분류 시설에 있음

```

{
  "Type" : "Notification",
  "MessageId" : "dc1e94d9-56c5-5e9
6-808d-cc7f68faa162",
  "TopicArn" : "arn:aws:sns:us-ea
st-2:111122223333:ExampleTopic1",
  "Message" : "Your job Job-name
(JID8bca334a-6c2f-4cd0-97e2
-3f5a4dc9bd6d) is at AWS sorting
facility. More info - https://
console.aws.amazon.com/impor
texport",
  "Timestamp" : "2023-02-23T00:27:
58.831Z",
  "SignatureVersion" : "1",
  "Signature" : "FMG5t1ZhJNHLHUXvZ
gtZz1k24FzVa7oX0T4P03neeXw8
ZEXZx6z35j2F0TuNYShn2h0bKNC/
zLTnMyIxEzmi2X1shOBWsJHkrW2xkR58ABZ
F+4uWHEE73yDVR4SyYAIkP9jstZzDRm
+bcVs8+T0yaLiEGLrIIIL4esi11lhIkG
ErCuy5btPcWXBdio2fpCRD5x9oR
6gmE/rd507lX1c1uvnv4r1Lkk4pqP2/
iUfxFZva1xLSRvgyfm6D9hNk1VyPfy+7
Ta1MD0lzmJu0rExtnSIbZew3foxgx8GT
+1bZkLd0ZdtdRJIyPRP44eyq78sU0Eo/
LsDr0Iak4ZDpg8dXg==",
  "SigningCertURL" : "https://
sns.us-east-1.amazonaws.com/
SimpleNotificationService-010a507c1
833636cd94bdb98bd93083a.pem",
  "UnsubscribeURL" : "https://
sns.us-east-2.amazonaws.com/?
Action=Unsubscribe&SubscriptionArn
=arn:aws:sns:us-east-2:1111
22223333:ExampleTopic1:e103
9402-24e7-40a3-a0d4-797da162b297"
}

```

작업 상태	SNS 알림 JSON
의 AWS	<pre> {   "Type" : "Notification",   "MessageId" : "dc1e94d9-56c5-5e9 6-808d-cc7f68faa162",   "TopicArn" : "arn:aws:sns:us-ea st-2:111122223333:ExampleTopic1",   "Message" : "Your job Job-name (JID8bca334a-6c2f-4cd0-97e2 -3f5a4dc9bd6d) is at AWS. More info - https://console.aws.amazon.com/ importexport",   "Timestamp" : "2023-02-23T00:27: 58.831Z",   "SignatureVersion" : "1",   "Signature" : "FMG5t1ZhJNHLHUXvZ gtZz1k24FzVa7oX0T4P03neeXw8 ZEXZx6z35j2F0TuNYShn2h0bKNC/ zLTnMyIxEzmi2X1sh0BWsJHkrW2xkR58ABZ F+4uWHEE73yDVR4SyYAIkP9jstZzDRm +bcVs8+T0yaLiEGLrIIIL4esi111hIkG ErCuy5btPcWXBdio2fpCRD5x9oR 6gmE/rd5071X1c1uvnv4r1Lkk4pqP2/ iUfxFZva1xLSRvgyfm6D9hNk1VyPfy+7 Ta1MD01zmJu0rExtnSIbZew3foxgx8GT +1bZkLd0ZdtRj1IyPRP44eyq78sU0Eo/ LsDr0Iak4ZDpg8dXg==",   "SigningCertURL" : "https:// sns.us-east-1.amazonaws.com/ SimpleNotificationService-010a507c1 833636cd94bdb98bd93083a.pem",   "UnsubscribeURL" : "https:// sns.us-east-2.amazonaws.com/? Action=Unsubscribe&amp;SubscriptionArn =arn:aws:sns:us-east-2:1111 22223333:ExampleTopic1:e103 9402-24e7-40a3-a0d4-797da162b297" } </pre>

작업 상태	SNS 알림 JSON
가져오기	<pre> {   "Type" : "Notification",   "MessageId" : "dc1e94d9-56c5-5e9 6-808d-cc7f68faa162",   "TopicArn" : "arn:aws:sns:us-ea st-2:111122223333:ExampleTopic1",   "Message" : "Your job Job-name (JID8bca334a-6c2f-4cd0-97e2 -3f5a4dc9bd6d) is being imported. More info - https://console.aw s.amazon.com/importexport",   "Timestamp" : "2023-02-23T00:27: 58.831Z",   "SignatureVersion" : "1",   "Signature" : "FMG5t1ZhJNHLHUXvZ gtZz1k24FzVa7oX0T4P03neeXw8 ZEXZx6z35j2F0TuNYShn2h0bKNC/ zLTnMyIxEzmi2X1sh0BWsJHkrW2xkr58ABZ F+4uWHEE73yDVR4SyYAIkP9jstZzDRm +bcVs8+T0yaLiEGLrIIIL4esi111hIkG ErCuy5btPcWXBdio2fpCRD5x9oR 6gmE/rd5071X1c1uvnv4r1Lkk4pqP2/ iUfxFZva1xLSRvgyfm6D9hNk1VyPfy+7 Ta1MD01zmJu0rExtnSIbZew3foxgx8GT +1bZkLd0ZdtRj1IyPRP44eyq78sU0Eo/ LsDr0Iak4ZDpg8dXg==",   "SigningCertURL" : "https:// sns.us-east-1.amazonaws.com/ SimpleNotificationService-010a507c1 833636cd94bdb98bd93083a.pem",   "UnsubscribeURL" : "https:// sns.us-east-2.amazonaws.com/? Action=Unsubscribe&amp;SubscriptionArn =arn:aws:sns:us-east-2:1111 22223333:ExampleTopic1:e103 9402-24e7-40a3-a0d4-797da162b297" } </pre>



작업 상태	SNS 알림 JSON
완료됨	<pre> {   "Type" : "Notification",   "MessageId" : "dc1e94d9-56c5-5e9 6-808d-cc7f68faa162",   "TopicArn" : "arn:aws:sns:us-ea st-2:111122223333:ExampleTopic1",   "Message" : "Your job Job-name (JID8bca334a-6c2f-4cd0-97e2 -3f5a4dc9bd6d) complete.\nThanks for using AWS Snowball Edge.\nCan you take a quick survey on your experienc e? Survey here: http://bit.ly/1pLQ JMY. More info - https://console.aw s.amazon.com/importexport",   "Timestamp" : "2023-02-23T00:27: 58.831Z",   "SignatureVersion" : "1",   "Signature" : "FMG5t1ZhJNHLHUXvZ gtZz1k24FzVa7oX0T4P03neeXw8 ZEXZx6z35j2F0TuNYShn2h0bKNC/ zLTnMyIxEzmi2X1shOBWsJHkrW2xkR58ABZ F+4uWHEE73yDVR4SyYAIkP9jstZzDRm +bcVs8+T0yaLiEGLrIIIL4esi11lhIkG ErCuy5btPcWXBdio2fpCRD5x9oR 6gmE/rd5071X1c1uvnv4r1Lkk4pqP2/ iUfxFZva1xLSRvgyfm6D9hNk1VyPfy+7 Ta1MD01zmJu0rExtnSIbZew3foxgx8GT +1bZkLd0ZdtdRJIyPRP44eyq78sU0Eo/ LsDr0Iak4ZDpg8dXg==",   "SigningCertURL" : "https:// sns.us-east-1.amazonaws.com/ SimpleNotificationService-010a507c1 833636cd94bdb98bd93083a.pem",   "UnsubscribeURL" : "https:// sns.us-east-2.amazonaws.com/? Action=Unsubscribe&amp;SubscriptionArn =arn:aws:sns:us-east-2:1111 22223333:ExampleTopic1:e103 9402-24e7-40a3-a0d4-797da162b297" } </pre>

작업 상태	SNS 알림 JSON

작업 상태	SNS 알림 JSON
취소됨	<pre>{   "Type" : "Notification",   "MessageId" : "dc1e94d9-56c5-5e9 6-808d-cc7f68faa162",   "TopicArn" : "arn:aws:sns:us-ea st-2:111122223333:ExampleTopic1",   "Message" : "Your job Job-name (JID8bca334a-6c2f-4cd0-97e2 -3f5a4dc9bd6d) was canceled. More info - https://console.aws.amazon. com/importexport",   "Timestamp" : "2023-02-23T00:27: 58.831Z",   "SignatureVersion" : "1",   "Signature" : "FMG5t1ZhJNHLHUXvZ gtZz1k24FzVa7oX0T4P03neeXw8 ZEXZx6z35j2F0TuNYShn2h0bKNC/ zLTnMyIxEzmi2X1sh0BWsJHkrW2xkr58ABZ F+4uWHEE73yDVR4SyYAIkP9jstZzDRm +bcVs8+T0yaLiEGLrIIIL4esi111hIkG ErCuy5btPcWXBdio2fpCRD5x9oR 6gmE/rd5071X1c1uvnv4r1Lkk4pqP2/ iUfxFZva1xLSRvgyfm6D9hNk1VyPfy+7 Ta1MD01zmJu0rExtnSIbZew3foxgx8GT +1bZkLd0ZdtRj1IyPRP44eyq78sU0Eo/ LsDr0Iak4ZDpg8dXg==",   "SigningCertURL" : "https:// sns.us-east-1.amazonaws.com/ SimpleNotificationService-010a507c1 833636cd94bdb98bd93083a.pem",   "UnsubscribeURL" : "https:// sns.us-east-2.amazonaws.com/? Action=Unsubscribe&amp;SubscriptionArn =arn:aws:sns:us-east-2:1111 22223333:ExampleTopic1:e103 9402-24e7-40a3-a0d4-797da162b297" }</pre>

# 를 사용하여 AWS Snowball Edge API 호출 로깅 AWS CloudTrail

AWS Snowball 또는 Snowball Edge 서비스는 사용자 AWS CloudTrail, 역할 또는 서비스가 수행한 작업에 대한 레코드를 제공하는 서비스와 통합됩니다. CloudTrail은 AWS Snowball Edge 서비스에 대한 모든 API 호출을 캡처합니다. 캡처되는 호출에는 AWS Snowball Edge 패밀리 콘솔의 호출과 AWS Snowball Edge 패밀리 작업 관리 API에 대한 코드 호출이 포함됩니다. 추적을 생성하면 AWS Snowball Edge Family API 직접 호출에 대한 이벤트를 포함한 CloudTrail 이벤트를 지속적으로 Amazon S3 버킷에 배포할 수 있습니다. 트레일을 구성하지 않은 경우에도 CloudTrail 콘솔의 이벤트 기록에서 최신 이벤트를 볼 수 있습니다. CloudTrail에서 수집한 정보를 사용하여 AWS Snowball Edge Family API로 수행된 요청, 수행된 요청의 IP 주소, 요청을 수행한 사람, 요청이 수행된 시간 및 추가 세부 정보를 확인할 수 있습니다.

CloudTrail에 대한 자세한 설명은 [AWS CloudTrail 사용자 가이드](#)를 참조하십시오.

## AWS Snowball Edge CloudTrail의 정보

CloudTrail은 계정을 생성할 AWS 계정 때에서 활성화됩니다. 에서 활동이 발생하면 AWS Snowball Edge 해당 활동이 이벤트 기록의 다른 AWS 서비스 이벤트와 함께 CloudTrail 이벤트에 기록됩니다. 에서 최근 이벤트를 보고 검색하고 다운로드할 수 있습니다 AWS 계정. 자세한 내용은 AWS CloudTrail 사용 설명서에서 [CloudTrail 이벤트 기록을 사용하여 이벤트 보기를](#) 참조하세요.

에 대한 이벤트를 AWS 계정 포함하여 이벤트를 지속적으로 기록하려면 추적을 AWS Snowball Edge 생성합니다. CloudTrail은 추적을 사용하여 Amazon S3 버킷으로 로그 파일을 전송할 수 있습니다. 콘솔에서 추적을 생성하면 기본적으로 모든 AWS 지역에 추적이 적용됩니다. 추적은 AWS 파티션 AWS 리전 의 모든에서 이벤트를 로깅하고 지정한 Amazon S3 버킷으로 로그 파일을 전송합니다. 또한 CloudTrail 로그에서 수집된 이벤트 데이터를 추가 분석 및 작업하도록 다른 AWS 서비스를 구성할 수 있습니다. 자세한 내용은 AWS CloudTrail 사용 설명서에서 다음 주제를 참조하세요.

- [트레일 생성 개요](#)
- [CloudTrail 지원 서비스 및 통합](#)
- [CloudTrail에 대한 Amazon SNS 알림 구성](#)
- [여러 리전으로부터 CloudTrail 로그 파일 받기 및 여러 계정으로부터 CloudTrail 로그 파일 받기](#)

모든 작업 관리 작업은 [AWS Snowball Edge API Reference](#)에 설명되어 있고 다음과 같은 예외를 고려하여 CloudTrail에서 로깅됩니다.

- [CreateAddress](#) 작업은 고객의 민감한 정보를 보호하기 위해 기록되지 않습니다.
- 모든 읽기 전용 API 호출은 (Get, Describe의 접두사로 시작하는 API 작업용, 또는 List) 응답 요소를 기록하지 않습니다.

모든 이벤트 또는 로그 항목에는 요청을 생성했던 사용자에 대한 정보가 포함됩니다. 자격 증명을 이용하면 다음을 쉽게 판단할 수 있습니다.

- 요청이 루트 또는 AWS Identity and Access Management (IAM 사용자) 자격 증명으로 이루어졌는지 여부입니다.
- 역할 또는 페더레이션 사용자에게 대한 임시 보안 인증을 사용하여 요청이 생성되었는지 여부.
- 요청이 다른 AWS 서비스에서 이루어졌는지 여부입니다.

자세한 내용은 AWS CloudTrail 사용 설명서의 [CloudTrail userIdentity 요소](#)를 참조하세요.

## 에 대한 로그 파일 항목 이해 AWS Snowball Edge

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 예시는 [DescribeJob](#) 작업을 보여주는 CloudTrail 로그 항목입니다.

```

{"Records": [
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "Root",
    "principalId": "111122223333",
    "arn": "arn:aws:iam::111122223333:root",
    "accountId": "111122223333",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "sessionContext": {"attributes": {
      "mfaAuthenticated": "false",
      "creationDate": "2019-01-22T21:58:38Z"
    }}
  }
},

```

```
    "invokedBy": "signin.amazonaws.com"
  },
  "eventTime": "2019-01-22T22:02:21Z",
  "eventSource": "snowball.amazonaws.com",
  "eventName": "DescribeJob",
  "awsRegion": "eu-west-1",
  "sourceIPAddress": "192.0.2.0",
  "userAgent": "signin.amazonaws.com",
  "requestParameters": {"jobId": "JIDa1b2c3d4-0123-abcd-1234-0123456789ab"},
  "responseElements": null,
  "requestID": "12345678-abcd-1234-abcd-ab0123456789",
  "eventID": "33c7ff7c-3efa-4d81-801e-7489fe6fff62",
  "eventType": "AwsApiCall",
  "recipientAccountId": "444455556666"
}
[]}
```

# AWS Snowball Edge 할당량

아래에서 AWS Snowball Edge 디바이스 사용 제한에 대한 정보를 찾을 수 있습니다.

## Important

Snowball Edge를 사용하여 Amazon Simple Storage Service(S3)에 데이터를 전송하는 경우, 개별 Amazon S3 객체 크기는 최소 0바이트에서 최대 5테라바이트(TB)까지 가능하다는 점에 유의하세요.

## 의 리전 가용성 AWS Snowball Edge

다음 표에서는를 사용할 수 있는 리전 AWS Snowball Edge 을 강조 표시합니다.

리전	Snowball Edge 가용성
미국 동부(오하이오)	✓
미국 동부(버지니아 북부)	✓
미국 서부(캘리포니아 북부)	✓
미국 서부(오리건)	✓
AWS GovCloud(미국 동부)	✓
AWS GovCloud(미국 서부)	✓
캐나다(중부)	✓
아시아 태평양(자카르타)	✓
아시아 태평양(뭄바이)	✓
아시아 태평양(오사카)	✓
아시아 태평양(서울)	✓
아시아 태평양(싱가포르)	✓

리전	Snowball Edge 가용성
아시아 태평양(시드니)	✓
아시아 태평양(도쿄)	✓
유럽(프랑크푸르트)	✓
유럽(아일랜드)	✓
유럽(런던)	✓
유럽(밀라노)	✓
유럽(파리)	✓
유럽(스톡홀름)	✓
중동(UAE)	✓
남아메리카(상파울루)	✓

지원되는 AWS 리전 및 엔드포인트에 대한 자세한 내용은 [AWS의 Snowball Edge 엔드포인트 및 할당량을 참조하세요](#). AWS 일반 참조

## AWS Snowball Edge 작업에 대한 제한 사항

AWS Snowball Edge 디바이스 작업 생성에는 다음과 같은 제한 사항이 있습니다.

- 보안을 위해 AWS Snowball Edge 디바이스를 사용하는 작업은 준비 후 360일 이내에 완료해야 합니다. 하나 이상의 디바이스를 360일보다 오래 유지해야 하는 경우에는 [Snowball Edge 디바이스에서 SSL 인증서 업데이트](#) 섹션을 참조하세요. 그렇지 않으면 360일이 지났을 때 디바이스가 잠겨 더 이상 액세스할 수 없게 되므로 반환해야 합니다. 가져오기 작업 중에 AWS Snowball Edge 디바이스가 잠기는 경우에도 디바이스의 기존 데이터를 Amazon S3로 전송할 수 있습니다.
- AWS Snowball Edge 는 Amazon S3-managed 암호화와 AWS Key Management Service 관리형 키를 사용한 서버 측 암호화(SSE-KMS)를 지원합니다. Snowball Edge의 Amazon S3 호환 스토리지는 로컬 컴퓨팅 및 스토리지 작업에 SSE-C를 지원합니다. 자세한 내용은 Amazon Simple Storage Service 사용 설명서의 [서버 측 암호화를 사용하여 데이터 보호](#)를 참조하세요.



- AWS Snowball Edge 디바이스를 사용하여 데이터를 가져오고 단일 Snowball Edge 디바이스에 맞는 것보다 더 많은 데이터를 전송해야 하는 경우 추가 작업을 생성합니다. 각 내보내기 작업은 여러 Snowball Edge 디바이스를 사용할 수 있습니다.
- 한 번에 보유할 수 있는 Snowball Edge 디바이스 수의 기본 서비스 제한은 계정당 1개입니다. AWS 리전. 서비스 제한을 높이거나 클러스터 작업을 생성하려면 [AWS Support](#)에 문의하세요.
- 디바이스로 전송된 객체의 메타데이터는 유지되지 않습니다. filename 및 filesize 메타데이터만 동일하게 유지됩니다. 다른 모든 메타데이터는 다음 예시와 같이 설정됩니다.

```
-rw-rw-r-- 1 root root [filesize] Dec 31 1969 [path/filename]
```

## 의 속도 제한 AWS Snowball Edge

속도 제한기는 서버 클러스터 환경에서 요청 속도를 제어하는 데 사용됩니다.

### Amazon Snow S3 어댑터 연결 한도

Amazon S3의 Snowball Edge에 대한 최대 연결 한도는 1,000개입니다. 1000개를 초과하는 모든 연결은 배제됩니다.

## Snowball Edge 디바이스를 사용한 온프레미스 데이터 전송에 대한 제한 사항

온프레미스에서 AWS Snowball Edge 디바이스로 데이터를 전송하거나 디바이스에서 데이터를 전송하기 위한 다음과 같은 제한 사항이 있습니다.

- 작성하는 동안 파일은 정적 상태여야 합니다. 전송 중에 수정되는 파일은 Amazon S3로 가져오기되지 않습니다.
- 점보 프레임, 즉 페이로드가 1,500바이트를 초과하는 이더넷 프레임은 지원되지 않습니다.
- 내보낼 데이터를 선택할 때 이름에 후행 슬래시(/ 또는 \)가 있는 객체는 전송되지 않음을 유의하세요. 후행 슬래시가 있는 객체를 내보내기 전에 슬래시를 제거하여 이름을 업데이트하세요.
- 멀티파트 데이터를 전송할 때 최대 크기는 2GiB입니다.

## Snowball Edge 디바이스에서 컴퓨팅 인스턴스 사용 시 할당량

다음은 AWS Snowball Edge 디바이스의 컴퓨팅 리소스에 대한 스토리지 할당량 및 공유 리소스 제한 사항입니다.

## Snowball Edge의 컴퓨팅 리소스 스토리지 할당량

컴퓨팅 리소스에 사용할 수 있는 스토리지는 Snowball Edge 디바이스의 전용 Amazon S3 스토리지와 분리된 별도 리소스입니다. 스토리지 할당량은 다음과 같습니다.

Snowball Edge 컴퓨팅 최적화 디바이스는 인스턴스당 최대 20AMIs와 10개의 볼륨을 실행할 수 있습니다.

인스턴스 유형	vCPU 코어	메모리 (GiB)	지원되는 디바이스 옵션
sbe1.small	1	1	스토리지 최적화
sbe1.medium	1	2	스토리지 최적화
sbe1.large	2	4	스토리지 최적화
sbe1.xlarge	4	8	스토리지 최적화
sbe1.2xlarge	8	16	스토리지 최적화

인스턴스 유형	vCPU 코어	메모리 (GiB)	지원 되는 디바 이스 옵션
sbe1.4xlarge	16	32	스토리지 최적화
sbe1.6xlarge	24	32	스토리지 최적화
sbe-c.small	1	2	컴퓨팅 최적화
sbe-c.medium	1	4	컴퓨팅 최적화
sbe-c.large	2	8	컴퓨팅 최적화
sbe-c.xlarge	4	16	컴퓨팅 최적화
sbe-c.2xlarge	8	32	컴퓨팅 최적화

인스턴스 유형	vCPU 코어	메모리 (GiB)	지원되는 디바이스 옵션
sbe-c.4xlarge	16	64	컴퓨팅 최적화
sbe-c.8xlarge	32	128	컴퓨팅 최적화
sbe-c.12xlarge	48	192	컴퓨팅 최적화
sbe-c.16xlarge	64	256	컴퓨팅 최적화
sbe-c.24xlarge	96	384	컴퓨팅 최적화

## Snowball Edge의 공유 컴퓨팅 리소스 제한 사항

Snowball Edge 디바이스의 모든 서비스는 디바이스에서 유한한 리소스를 사용합니다. 사용 가능한 컴퓨팅 리소스가 최대화된 Snowball Edge 디바이스는 새 컴퓨팅 리소스를 시작할 수 없습니다. 예를 들어 스토리지 최적화 디바이스에서 sbe1.4xlarge 컴퓨팅 인스턴스를 실행하면서 동시에 NFS 인터페이스를 시작하려는 경우 NFS 인터페이스 서비스는 시작되지 않습니다. 다음은 각 서비스의 요구 사항과 다양한 서비스 옵션에서 사용 가능한 리소스의 개요입니다.

- 컴퓨팅 서비스가 ACTIVE 상태가 아닌 경우,
  - 스토리지 최적화 옵션에서 컴퓨팅 인스턴스에 대해 24개 vCPU 및 32GiB 메모리가 사용됩니다.
  - 컴퓨팅 최적화 옵션에는 컴퓨팅 인스턴스용 vCPUs개와 메모리 208GiB가 있습니다.

- AWS IoT Greengrass 및 AWS Lambda 가 AWS IoT Greengrass 인 경우 ACTIVE:
  - 스토리지 최적화 옵션에서 이러한 서비스는 4개 vCPU 코어 및 8GiB 메모리를 사용합니다.
  - 컴퓨팅 최적화 옵션에서 이러한 서비스는 vCPU 코어 1개 및 1GiB 메모리를 사용합니다.
  - NFS 인터페이스가 ACTIVE 상태인 경우 Snowball Edge 디바이스에 대해 vCPU 코어 8개 및 16GiB 메모리가 사용됩니다.
  - Snowball Edge의 Amazon S3 호환 스토리지는 AMD EPYC Gen2 및 NVME로 최적화된 Snowball Edge 컴퓨팅에서 ACTIVE이지만 Snowball Edge의 Amazon S3 호환 스토리지 최소 구성이 3TB 인 단일 노드의 경우 vCPU 코어 8개와 메모리 16GB를 사용합니다. Snowball Edge에서 3TB 이상의 Amazon S3 호환 스토리지가 있는 단일 노드의 경우 vCPU 코어 20개와 메모리 40GB를 사용합니다. 클러스터는 20개의 vCPU 코어와 40GB의 메모리를 사용합니다.

Snowball Edge 클라이언트에서 `snowballEdge describe-service` 명령을 사용하여 서비스가 Snowball Edge에서 ACTIVE 상태인지 확인할 수 있습니다. 자세한 내용은 [Snowball Edge에서 실행되는 서비스의 위반 상태](#) 단원을 참조하십시오.

## Snowball Edge Edge 디바이스 배송에 대한 제한 사항

AWS Snowball Edge 디바이스 배송에는 다음과 같은 제한 사항이 있습니다.

- AWS 는 Snowball Edge Edge 디바이스를 우체국 상자로 배송하지 않습니다.
- AWS 는 EU(아일랜드)에서 EU(프랑크푸르트) 또는 아시아 태평양(시드니)으로 등 미국 외 리전 간에 Snowball Edge Edge 디바이스를 배송하지 않습니다.
- Snowball Edge Edge 디바이스를 작업 생성 시 지정된 국가 외부 주소로 이동하는 것은 허용되지 않으며 AWS 서비스 약관을 위반하는 것입니다.


배송에 대한 자세한 내용은 [Snowball Edge의 배송 고려 사항](#) 섹션을 참조하세요.

## 반환된 Snowball Edge Edge 가져오기 처리에 대한 제한 사항

데이터를 로 가져오려면 디바이스 AWS가 다음 요구 사항을 충족해야 합니다.

- AWS Snowball Edge 디바이스가 손상되지 않아야 합니다. 전면, 후면 및 상단에서 세 개의 문을 열거나 선택적 공기 필터를 추가 및 교체하는 경우를 제외하고 어떤 이유로든 AWS Snowball Edge 디바이스를 열지 마십시오.

- 디바이스가 물리적으로 손상되어서는 안 됩니다. 래치에서 딸깍 소리가 날 때까지 Snowball Edge Edge 디바이스의 세 문을 닫으면 손상을 방지할 수 있습니다.
- Snowball Edge Edge 디바이스의 E Ink 디스플레이가 표시되어야 합니다. 또한 데이터를 AWS Snowball Edge 디바이스로 전송한 후 자동으로 생성된 반환 레이블도 표시해야 합니다.

 Note

이러한 요구 사항을 충족하지 않는 반환된 모든 Snowball Edge Edge 디바이스는 작업을 수행하지 않고 삭제됩니다.

## 문제 해결 AWS Snowball Edge

문제를 해결할 때는 다음과 같은 일반 지침을 염두에 두세요.

- Amazon S3 객체의 크기는 최대 5TB입니다.
- AWS Snowball Edge 디바이스로 전송된 객체의 최대 키 길이는 933바이트입니다. 1바이트보다 큰 문자를 포함하는 키 이름의 최대 키 길이 또한 여전히 933바이트입니다. 키 길이를 확인할 경우 파일 또는 객체 이름과 경로 또는 접두사를 포함해야 합니다. 따라서 파일 이름이 짧더라도 경로가 여러 개 중첩되어 길어지면 933바이트를 초과할 수 있습니다. 키 길이를 확인할 때 버킷 이름은 경로에 포함되지 않습니다. 다음 예를 참조하세요.

객체 이름	버킷 이름	경로와 버킷 이름	키 길이
sunflower-1.jpg	pictures	sunflower-1.jpg	15자
receipts.csv	MyTaxInfo	/Users/Eric/Documents/2016/January/	47자
bhv.1	\$7\$zWwwXKQj\$gLA0oZCj\$r8p	/.VfV/FqGC3QN\$7Bxys3KHYePfuIOMNjY83dVxugPY1xVg/evpcQEJLT/rSwZc\$M1Vvf/\$hwefVISRqwepB\$/BiiD/PPF\$twRAjrD/fIMp/0NY	135자

- 보안을 위해 AWS Snowball Edge 디바이스를 사용하는 작업은 준비 후 360일 이내에 완료해야 합니다. 하나 이상의 디바이스를 360일보다 오래 유지해야 하는 경우에는 [Snowball Edge 디바이스에서 SSL 인증서 업데이트](#) 섹션을 참조하세요. 그렇지 않으면 360일이 지났을 때 디바이스가 잠겨 더 이상 액세스할 수 없게 되므로 반환해야 합니다. 가져오기 작업 중에 AWS Snowball Edge 디바이스가 잠기는 경우에도 디바이스의 기존 데이터를 Amazon S3로 전송할 수 있습니다.
- AWS Snowball Edge 디바이스를 사용하여 예기치 않은 오류가 발생하는 경우 이에 대해 듣고 싶습니다. 관련 로그를 복사하여 메시지에서 발생한 문제에 대한 간략한 설명과 함께 포함합니다 AWS Support. 로그에 대한 자세한 내용은 [Snowball Edge 클라이언트 구성 및 사용](#) 섹션을 참조하세요.

## 주제

- [Snowball Edge를 식별하는 방법](#)
- [Snowball Edge의 부팅 문제 해결](#)
- [Snowball Edge의 연결 문제 해결](#)
- [Snowball Edge의 unlock-device 명령 문제 해결](#)
- [Snowball Edge의 자격 증명 문제 해결](#)
- [Snowball Edge의 데이터 전송 문제 해결](#)
- [Snowball Edge AWS CLI 문제 해결](#)
- [Snowball Edge의 컴퓨팅 인스턴스 문제 해결](#)

## Snowball Edge를 식별하는 방법

describe-device 명령을 사용하여 디바이스 유형을 찾은 다음 아래 표에 있는 DeviceType의 반환된 값을 찾아 구성을 확인합니다.

```
snowballEdge describe-device
```

### Example **describe-device** 출력

```
{
  "DeviceId" : "JID-206843500001-35-92-20-211-23-06-02-18-24",
  "UnlockStatus" : {
```



```

    "State" : "UNLOCKED"
    ...
    "DeviceType" : "V3_5C"
  }

```

## DeviceType 및 Snowball Edge 디바이스 구성

DeviceType 값	디바이스 구성
V3_5C	Snowball Edge 컴퓨팅 최적화(AMD EPYC Gen2 및 NVME 포함)
V3_5S	Snowball Edge 스토리지 최적화 210TB

Snowball Edge 디바이스 구성에 대한 자세한 내용은 [AWS Snowball Edge 디바이스 하드웨어 정보](#) 섹션을 참조하세요.

## Snowball Edge의 부팅 문제 해결

다음 정보는 Snowball Edge 부팅 시 발생할 수 있는 특정 문제를 해결하는 데 도움이 될 수 있습니다.

- 디바이스가 부팅되는 데 10분이 걸릴 수 있습니다. 이 시간 동안에는 디바이스를 옮기거나 사용하지 마세요.
- 전원을 공급하는 케이블의 양쪽 끝이 단단히 연결되어 있는지 확인하세요.
- 전원을 공급하는 케이블을 정상 작동이 확인된 다른 케이블로 교체하세요.
- 전원을 공급하는 케이블을 정상 작동이 확인된 다른 전원에 연결합니다.

## 부팅 중 Snowball Edge 디바이스의 LCD 디스플레이 관련 문제 해결

때때로 Snowball Edge 디바이스의 전원을 켜 후 LCD 디스플레이에 문제가 발생할 수 있습니다.

- Snowball Edge 디바이스에 전원을 연결하고 LCD 화면 위의 전원 버튼을 누르면 LCD 화면은 검은색이며 이미지가 표시되지 않습니다.
- LCD 화면이 Snowball Edge를 설정하고 있습니다. 몇 분 정도 걸릴 수 있습니다. 메시지를 지나서 더 이상 진전되지 않고 네트워크 구성 화면이 나타나지 않습니다.

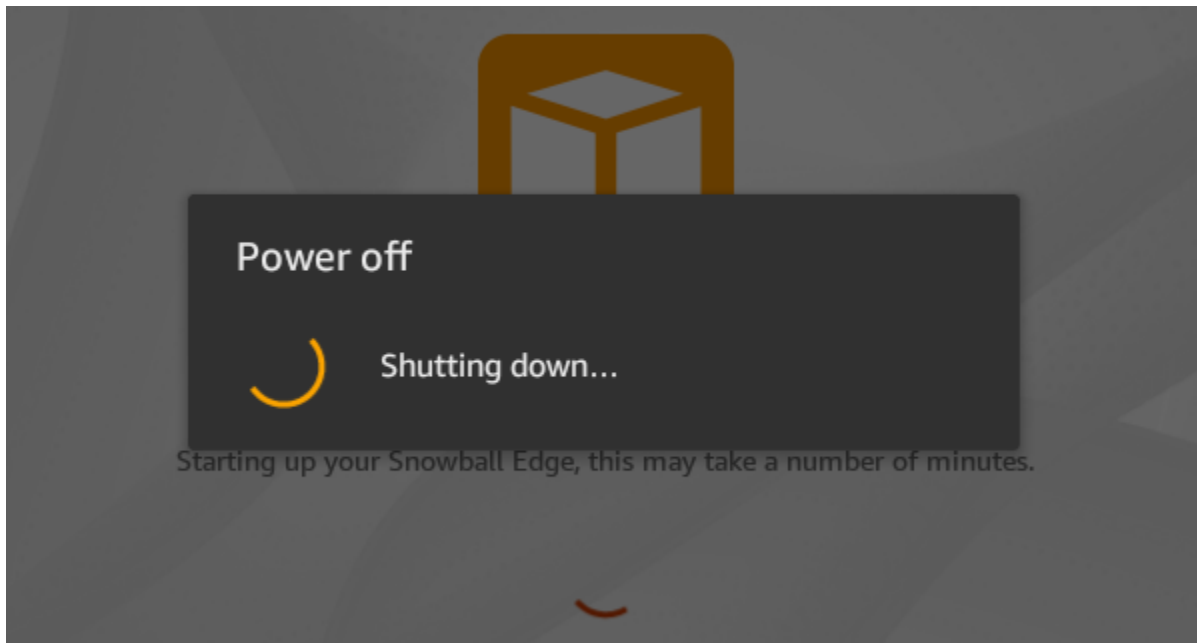


전원 버튼을 누른 후 LCD 화면이 검은색일 때 수행할 작업

1. Snowball Edge 디바이스가 전원에 연결되어 있고 전원이 공급되고 있는지 확인합니다.
2. 1~2시간 동안 디바이스를 전원에 연결된 상태로 둡니다. 디바이스 전면과 후면의 도어가 열려 있는지 확인합니다.
3. 디바이스로 돌아가면 LCD 화면을 사용할 준비가 됩니다.

Snowball Edge가 네트워크 구성 화면으로 진행되지 않을 때 수행할 작업

1. 화면에 10분 동안 Snowball Edge 설정을 그대로 두세요. 몇 분 정도 걸릴 수 있습니다. 메시지를 표시하세요.
2. 화면에서 디스플레이 재시작 버튼을 선택합니다. 종료 중... 메시지가 표시되고 Snowball Edge를 설정하는 데 몇 분 정도 걸릴 수 있습니다. 메시지가 표시되고 디바이스가 정상적으로 시작됩니다.



디스플레이 재시작 버튼을 사용한 후 LCD 화면에 Snowball Edge 설정을 지나치지 않으면 몇 분 정도 걸릴 수 있습니다. 메시지가 표시되면 다음 절차를 사용하세요.

#### 취할 조치

1. LCD 화면 위의 전원 버튼을 눌러 디바이스의 전원을 끕니다.
2. 디바이스에서 모든 케이블을 분리합니다.
3. 디바이스의 전원을 끄고 20분 동안 연결 해제 상태로 둡니다.
4. 전원 및 네트워크 케이블을 연결합니다.
5. LCD 화면 위의 전원 버튼을 눌러 디바이스의 전원을 켵니다.

문제가 지속되면 AWS Support 에 문의하여 디바이스를 반환하고 새 Snowball Edge 디바이스를 받습니다.

## 부팅 중 E Ink 디스플레이 관련 문제 해결

때때로 Snowball Edge 디바이스의 전원을 켜 후 디바이스 상단의 E Ink 디스플레이에 다음 메시지가 표시될 수 있습니다.

The appliance has timed out

이 메시지는 디바이스에 문제가 있음을 나타내지 않습니다. 정상적으로 사용하고 끄면 반쯤 AWS배송 정보가 예상대로 표시됩니다.

## Snowball Edge의 연결 문제 해결

다음 정보는 Snowball Edge와 연결할 때 발생할 수 있는 특정 문제를 해결하는 데 도움이 될 수 있습니다.

- 초당 100메가바이트의 속도로 작동하는 라우터와 스위치는 Snowball Edge에서는 작동하지 않습니다. 초당 1GB 이상의 속도로 작동하는 스위치를 사용할 것을 권장합니다.
- 디바이스에 특이한 연결 오류가 발생하는 경우 Snowball Edge의 전원을 끄고 모든 케이블을 뽑은 후 10분 동안 그대로 둡니다. 10분이 지나면 디바이스를 다시 시작한 후 다시 연결을 시도합니다.
- 안티바이러스 소프트웨어 또는 방화벽이 Snowball Edge 디바이스의 네트워크 연결을 차단하고 있는지 확인합니다.
- NFS 인터페이스와 Amazon S3 인터페이스의 IP 주소는 다르다는 점에 유의하세요.

더 높은 수준에서 연결 문제를 해결하려면 다음 단계를 따르세요.

- Snowball Edge와 통신할 수 없다면 디바이스의 IP 주소를 ping합니다. ping이 no connect를 반환하면 디바이스의 IP 주소를 확인하고 로컬 네트워크 구성을 확인합니다.
- IP 주소가 정확한데 디바이스 후면의 등이 깜박이는 경우 telnet을 사용하여 포트 22, 9091, 8080에서 디바이스를 테스트합니다. 포트 22를 테스트하면 Snowball Edge가 제대로 작동하고 있는지 확인할 수 있습니다. 포트 9091을 테스트하면를 사용하여 디바이스에 명령을 전송할 AWS CLI 수 있는지 여부가 결정됩니다. 포트 8080 테스트는 디바이스가 S3 어댑터만으로 Amazon S3 버킷에 쓸 수 있는지 확인합니다. 포트 22에서 연결할 수 있지만 포트 8080에서는 연결할 수 없다면 먼저 Snowball Edge의 전원을 끈 다음, 모든 케이블을 뽑습니다. 10분 동안 디바이스를 그대로 두었다가 다시 연결한 후 다시 시작합니다.

## Snowball Edge의 **unlock-device** 명령 문제 해결

unlock-device 명령이 connection refused를 반환할 경우 명령 구문을 잘못 입력했거나 컴퓨터 또는 네트워크 구성으로 인해 명령이 Snow 디바이스에 도달하지 못할 수 있습니다. 상황을 해결하려면 다음 조치를 취하세요.

1. 명령이 올바르게 입력되었는지 확인합니다.
  - a. 디바이스의 LCD 화면을 사용하여 명령에 사용된 IP 주소가 올바른지 확인합니다.

- b. 파일 이름을 포함하여 명령에 사용된 매니페스트 파일의 경로가 올바른지 확인합니다.
  - c. [AWS Snowball Edge Management Console](#)을 사용하여 명령에 사용된 잠금 해제 코드가 올바른지 확인합니다.
2. 사용 중인 컴퓨터가 Snow 디바이스와 동일한 네트워크 및 서브넷에 있는지 확인합니다.
  3. 사용 중인 컴퓨터와 네트워크가 Snow 디바이스에 대한 액세스를 허용하도록 구성되어 있는지 확인합니다. 운영 체제의 ping 명령을 사용하여 컴퓨터가 네트워크를 통해 Snow 디바이스에 도달할 수 있는지 확인합니다. 바이러스 백신 소프트웨어의 구성, 방화벽 구성, 가상 프라이빗 네트워크(VPN) 또는 컴퓨터 및 네트워크의 기타 구성을 확인합니다.

## Snowball Edge의 매니페스트 파일 문제 해결

각 작업에는 연결된 특정 매니페스트 파일이 있습니다. 작업을 여러 개 생성하는 경우 각 매니페스트가 어떤 작업에 해당하는지 추적합니다.

매니페스트 파일을 분실했거나 매니페스트 파일이 손상되었다면 특정 작업에 대해 매니페스트 파일을 다시 다운로드할 수 있습니다. 콘솔 AWS CLI 또는 AWS APIs.

Snowball Edge에서 업데이트를 실행하는 경우 새 매니페스트 파일을 다운로드하여 작업에 사용해야 합니다. 매니페스트 파일 다운로드에 대한 자세한 내용은 섹션을 참조하세요 [Snowball Edge에 액세스하기 위한 자격 증명 가져오기](#).

## Snowball Edge의 자격 증명 문제 해결

다음 주제를 통해 Snowball Edge 디바이스에서 발생하는 자격 증명 문제를 해결할 수 있습니다.

### Snowball Edge의 자격 증명을 찾을 AWS CLI 수 없음

를 사용하여 Amazon S3 인터페이스를 통해 AWS Snowball Edge 디바이스와 통신하는 경우 자격 증명을 찾을 AWS CLI 수 없음이라는 오류 메시지가 표시될 수 있습니다. "aws configure"를 실행하여 자격 증명을 구성할 수 있습니다.

#### 취할 조치

가 명령을 실행하는 데 AWS CLI 사용하는 AWS 자격 증명을 구성합니다. 자세한 내용은 AWS Command Line Interface 사용 설명서의 [AWS CLI 구성](#)을 참조하세요.

## Snowball Edge 오류 메시지 문제 해결: 보안 액세스 키 및 서명 확인

Amazon S3 인터페이스를 사용하여 Snowball Edge로 데이터를 전송할 때 다음과 같은 오류 메시지가 발생할 수 있습니다.

An error occurred (SignatureDoesNotMatch) when calling the CreateMultipartUpload operation: The request signature we calculated does not match the signature you provided.

Check your AWS secret access key and signing method. For more details go to:

<http://docs.aws.amazon.com/AmazonS3/latest/dev/>

[RESTAuthentication.html#ConstructingTheAuthenticationHeader](http://docs.aws.amazon.com/AmazonS3/latest/dev/RESTAuthentication.html#ConstructingTheAuthenticationHeader)

### 취할 조치

Snowball Edge 클라이언트에서 자격 증명을 받으세요. 자세한 내용은 [Snowball Edge에 대한 자격 증명 가져오기](#) 단원을 참조하십시오.

## Snowball Edge의 데이터 전송 문제 해결

Snowball Edge로 데이터를 전송하거나 Snowball Edge로부터 데이터를 받을 때 성능 문제가 발생할 경우 [Snowball Edge와 최상의 데이터 전송 성능을 위한 권장 사항](#)에서 전송 성능을 향상시키기 위한 권장 사항 및 지침을 참조하세요. 다음은 Snowball Edge와의 데이터 송수신 시 발생할 수 있는 문제를 해결하는 데 도움이 되는 정보입니다.

- Snowball Edge의 루트 디렉터리로 데이터를 전송할 수 없습니다. 데이터를 디바이스로 전송하는 데 문제가 있는 경우, 하위 디렉터리로 데이터 전송을 시도하고 있는지 확인하세요. 최상위 하위 디렉터리에 귀하가 작업에 포함한 Amazon S3 버킷의 이름이 있습니다. 데이터를 이 하위 디렉터리에 배치합니다.
- Linux를 사용할 경우 UTF-8 문자 파일을 AWS Snowball Edge 디바이스에 업로드할 수 없다면 Linux 서버에서 UTF-8 문자 인코딩을 인식하지 못하기 때문일 수 있습니다. 이 문제는 `locales` 패키지를 Linux 서버에 설치하고 `en_US.UTF-8`과 같은 UTF-8 로캘 중 하나로 구성하면 해결할 수 있습니다. `locales` 패키지는 환경 변수 `LC_ALL`을 내보내서 구성할 수 있습니다(예: `export LC_ALL=en_US.UTF-8`).
- Amazon S3 인터페이스를와 함께 사용하는 경우 `my photo.jpg` 또는와 같이 이름에 공백이 있는 파일 또는 폴더로 작업할 AWS CLI 수 있습니다 `My Documents`. 하지만 이러한 경우 공백을 제대로 처리해야 합니다. 자세한 내용은 AWS Command Line Interface 사용 설명서의 [AWS CLI에 대한 파라미터 값 지정](#)을 참조하세요.

## Snowball Edge의 가져오기 작업 문제 해결

Amazon S3에 파일 가져오기가 실패하는 경우가 있습니다. 다음과 같은 문제가 발생하는 경우 문제를 해결하기 위해 지정된 작업을 시도하세요. 파일 가져오기가 실패하면 가져오기를 다시 시도합니다. 다시 가져오려면 Snowball Edge에 대한 새로운 작업이 필요할 수 있습니다.

객체 이름의 잘못된 문자로 인해 Amazon S3에 파일 가져오기 실패

이 문제는 파일 또는 폴더 이름에 Amazon S3에서 지원하지 않는 문자가 있는 경우 발생합니다. Amazon S3에는 객체 이름에 포함할 수 있는 문자에 대한 규칙이 있습니다. 자세한 내용을 알아보려면 Amazon S3 사용 설명서의 [객체 키 이름 생성](#)을 참조하세요.

취할 조치

이 문제가 발생하는 경우 작업 완성 보고서에서 가져오기에 실패한 파일 및 폴더의 목록이 표시됩니다.

경우에 따라 이 목록은 엄청나게 크거나 목록의 파일이 너무 커서 인터넷으로 전송할 수 없습니다. 이러한 경우 새 Snowball 가져오기 작업을 생성하고, Amazon S3 규칙에 맞게 파일 및 폴더 이름을 변경하고, 파일을 다시 전송해야 합니다.

파일이 작고 많은 수가 없는 경우 AWS CLI 또는를 통해 Amazon S3에 파일을 복사할 수 있습니다 AWS Management Console. 자세한 내용은 Amazon Simple Storage Service Console 사용 설명서의 [S3 버킷에 파일 및 폴더를 업로드하려면 어떻게 해야 하나요?](#) 섹션을 참조하세요.

## Snowball Edge의 내보내기 작업 문제 해결

워크스테이션에 파일 내보내기가 실패하는 경우가 있습니다. 다음과 같은 문제가 발생하는 경우 문제를 해결하기 위해 지정된 작업을 시도하세요. 파일 내보내기가 실패하면 내보내기를 다시 시도합니다. 다시 내보내려면 Snowball Edge에 대한 새로운 작업이 필요할 수 있습니다.

Microsoft Windows Server로 파일 내보내기 실패

파일 또는 관련 폴더가 Windows가 지원하지 않는 형식으로 이름이 지정된 경우 Microsoft Windows Server로 파일 내보내기가 실패할 수 있습니다. 예를 들어 파일 또는 폴더 이름에 콜론(:)이 있는 경우 Windows가 파일 또는 폴더 이름에 해당 문자를 허용하지 않기 때문에 내보내기가 실패합니다.

취할 조치

- 오류의 원인인 이름의 목록을 만듭니다. 내보내기에 실패한 파일 및 폴더의 이름은 로그에서 확인할 수 있습니다. 자세한 내용은 [Snowball Edge에서 로그 보기 및 다운로드](#) 섹션을 참조하세요.

2. Amazon S3에서 문제의 원인인 객체의 이름을 변경하여 지원되지 않는 문자를 제거 또는 교체합니다.
3. 이름 목록이 엄청나게 크거나 목록의 파일이 너무 커서 인터넷으로 전송할 수 없는 경우 해당 객체에 대해 특히 새 내보내기 작업을 생성합니다.

파일이 작고 파일이 많지 않은 경우 AWS CLI 또는를 통해 Amazon S3에서 이름이 변경된 객체를 복사합니다 AWS Management Console. 자세한 내용은 Amazon Simple Storage Service Console 사용 설명서의 [S3 버킷에서 객체를 다운로드하려면 어떻게 해야 하나요?](#)를 참조하세요.

## Snowball Edge의 NFS 인터페이스 문제 해결

Snowball Edge는 NFS 인터페이스의 상태가 임을 나타낼 수 있습니다DEACTIVATED. 이는 NFS 인터페이스를 먼저 중지하지 않고 Snowball Edge의 전원이 꺼진 경우 발생할 수 있습니다.

### 취할 조치

문제를 해결하려면 다음 단계를 사용하여 NFS 서비스를 중지하고 다시 시작합니다.

1. describe-service 명령을 사용하여 서비스 상태를 확인합니다.

```
snowballEdge describe-service --service-id nfs
```

명령은 다음을 반환합니다.

```
{
  "ServiceId" : "nfs",
  "Status" : {
    "State" : "DEACTIVATED"
  }
}
```

2. stop-service 명령을 사용하여 NFS 서비스를 중지합니다.

```
snowballEdge stop-service --service-id nfs
```



3. `start-service` 명령을 사용하여 NFS 서비스를 중지합니다. 자세한 내용은 [NFS 인터페이스 관리](#)를 참조하세요.

```
snowballEdge start-service --virtual-network-interface-arns vni-arn --service-id
nfs --service-configuration AllowedHosts=0.0.0.0/0
```

4. `describe-service` 명령을 사용하여 서비스가 실행 중인지 확인합니다.

```
snowballEdge describe-service --service-id nfs
```

State 이름 값이 ACTIVE이면 NFS 인터페이스 서비스가 활성 상태입니다.

```
{
  "ServiceId" : "nfs",
  "Status" : {
    "State" : "ACTIVE"
  },
  "Endpoints" : [ {
    "Protocol" : "nfs",
    "Port" : 2049,
    "Host" : "192.0.2.0"
  } ],
  "ServiceConfiguration" : {
    "AllowedHosts" : [ "10.24.34.0/23", "198.51.100.0/24" ]
  }
}
```

## S3 인터페이스를 사용하여 데이터를 전송할 때 액세스 거부 오류 문제 해결

S3 인터페이스를 사용하여 Snowball Edge 디바이스와 데이터를 주고받을 때 액세스 거부 오류가 발생할 수 있습니다. 이 오류는 IAM 사용자 또는 버킷 정책의 결과일 수 있습니다.

### 취할 조치

1. 다음 구문 문제에 대해 사용 중인 S3 버킷의 정책을 확인합니다.

- a. KMS 헤더가 전달된 경우에만 정책이 데이터 업로드를 허용하는 경우 정책이 사용자 ID 대신 보안 주체 ARN을 지정하는지 확인합니다. 아래 예제는 올바른 구문을 보여줍니다.

```
{
  "Sid": "Statement3",
  "Effect": "Deny",
  "Principal": "*",
  "Action": "s3:PutObject",
  "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
  "Condition": {
    "StringNotLike": {
      "aws:PrincipalArn": "arn:aws:iam::111122223333:role/JohnDoe"
    },
    "StringNotEquals": {
      "s3:x-amz-server-side-encryption": [
        "aws:kms",
        "AES256"
      ]
    }
  }
},
{
  "Sid": "Statement4",
  "Effect": "Deny",
  "Principal": "*",
  "Action": "s3:PutObject",
  "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
  "Condition": {
    "StringNotLike": {
      "aws:PrincipalArn": "arn:aws:iam::111122223333:role/JohnDoe"
    },
    "Null": {
      "s3:x-amz-server-side-encryption": "true"
    }
  }
}
}
```

- b. 버킷 정책에서 올바른 헤더가 전달된 경우에만 버킷에 대한 업로드를 허용하는 경우 Snowball Edge 디바이스의 업로드는 기본적으로 헤더를 전달하지 않습니다. 데이터를 업로

드하는 데 사용된 IAM 사용자에게 대한 예외를 허용하도록 정책을 수정합니다. 다음은 이에 대한 올바른 구문의 예입니다.

```
{
  "Sid": "Statement3",
  "Effect": "Deny",
  "Principal": "",
  "Action": "s3:PutObject",
  "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/",
  "Condition": {
    "StringNotEquals": {
      "s3:x-amz-server-side-encryption": "AES256"
    },
    "StringNotLike": {
      "aws:PrincipalArn": "arn:aws:iam::111122223333:role/JohnDoe"
    }
  }
},
{
  "Sid": "Statement4",
  "Effect": "Deny",
  "Principal": "",
  "Action": "s3:PutObject",
  "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/",
  "Condition": {
    "Null": {
      "s3:x-amz-server-side-encryption": "true"
    },
    "StringNotLike": {
      "aws:PrincipalArn": "arn:aws:iam::111122223333:role/JohnDoe"
    }
  }
}
```

- 보안 주체 요소의 올바른 구문에 대해 사용 중인 KMS 키의 정책을 확인합니다. 아래 예제에서 올바른 구문을 확인하세요.

```
{
  "Sid": "Statement2",
  "Effect": "Allow",
```

```

    "Principal": {
      "AWS": [
        "arn:aws:iam::111122223333:role/service-role/JohnDoe"
      ]
    },
    "Action": [
      "kms:Encrypt",
      "kms:Decrypt",
      "kms:ReEncrypt*",
      "kms:GenerateDataKey*",
      "kms:DescribeKey"
    ],
    "Resource": "*"
  }

```

## S3 인터페이스를 사용하여 데이터를 전송할 때 403 금지 오류 문제 해결

S3 인터페이스를 사용하여 Snowball Edge 디바이스와 데이터를 주고받을 때 403 금지 오류가 발생할 수 있습니다. 이 오류는 IAM 사용자 또는 버킷 정책의 결과일 수 있습니다. 다음 구문 문제에 대해 사용 중인 S3 버킷의 정책을 확인합니다.

### 취할 조치

1. 정책은 PrincipalArn을 제공하지 않습니다. 다음 정책을 예로 사용하여 aws:PrincipalArn 헤더를 사용하고 없이 IAM 역할 ARN을 제공합니다.\*.

### JSON

```

{
  "Version": "2012-10-17",
  "Statement": [{
    "Sid": "DenyIncorrectEncryptionHeader",
    "Effect": "Deny",
    "Principal": "*",
    "Action": "s3:PutObject",
    "Resource": "arn:aws:s3:::BucketName/*",
    "Condition": {
      "StringNotLike": {

```

```

        "aws:PrincipalArn": "arn:aws:iam::1234567890:role/
RoleName"
    },
    "StringNotEquals": {
        "s3:x-amz-server-side-encryption": [
            "aws:kms",
            "AES256"
        ]
    }
},
{
    "Sid": "DenyUnEncryptedObjectUploads",
    "Effect": "Deny",
    "Principal": "*",
    "Action": "s3:PutObject",
    "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*",
    "Condition": {
        "StringNotLike": {
            "aws:PrincipalArn": "arn:aws:iam::1234567890:role/
RoleName"
        },
        "Null": {
            "s3:x-amz-server-side-encryption": "true"
        }
    }
},
{
    "Sid": "DenyInsecureTransport",
    "Effect": "Deny",
    "Principal": "*",
    "Action": "s3:*",
    "Resource": [
        "arn:aws:s3:::BucketName/*",
        "arn:aws:s3:::BucketName"
    ],
    "Condition": {
        "Bool": {
            "aws:SecureTransport": "false"
        }
    }
},
{
    "Sid": "AllowSnowballPutObjectAccess",

```

```

        "Effect": "Allow",
        "Principal": {
            "AWS": "arn:aws:iam::1234567890:role/RoleName"
        },
        "Action": "s3:PutObject",
        "Resource": "arn:aws:s3:::BucketName/*"
    }
]
}s

```

2. KMS 정책이 잘못된 IAM 역할 형식을 사용하는 경우 403 오류가 발생할 수 있습니다. 데이터를 업로드하는 데 사용된 IAM 사용자에게 예외를 허용하도록 정책을 수정합니다. 다음은 이에 대한 올바른 구문의 예입니다.

```

{{
    "Sid": "Allow use of the key",
    "Effect": "Allow",
    "Principal": {
        "AWS": [
            "arn:aws:iam::1234567890:role/service-role/RoleName"
        ]
    },
    "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:GenerateDataKey*"
    ],
    "Resource": "*"
}

```

3. IAM 역할은 암호화 헤더 조건을 우회해야 할 수 있습니다. 기본적으로 Snowball Edge 디바이스에 저장된 모든 객체는 SSE-S3 암호화로 암호화됩니다. 아래 정책을 사용하여 IAM 역할이 암호화 헤더 없이 객체를 업로드하는 예외를 제공합니다.

## JSON

```

{
    "Version": "2012-10-17",

```

```

    "Id": "PutObjPolicy",
    "Statement": [{
      "Sid": "DenyIncorrectEncryptionHeader",
      "Effect": "Deny",
      "Principal": "*",
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::BucketName/",
      "Condition": {
        "StringNotEquals": {
          "s3:x-amz-server-side-encryption": "AES256"
        },
        "StringNotLike": {
          "aws:PrincipalArn": "arn:aws:iam::1234567890:role/
RoleName"
        }
      }
    },
    {
      "Sid": "DenyUnEncryptedObjectUploads",
      "Effect": "Deny",
      "Principal": "*",
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::BucketName/*",
      "Condition": {
        "Null": {
          "s3:x-amz-server-side-encryption": "true"
        },
        "StringNotLike": {
          "aws:PrincipalArn": "arn:aws:iam::1234567890:role/
RoleName"
        }
      }
    }
  ]
}

```

- 오류 메시지는 IP 조건과 함께 NotPrincipal을 사용하여 PutObject에 대한 액세스가 거부되었음을 나타냅니다. 아래와 같이 Snowball Edge IAM 역할에 대한 예외를 추가합니다.

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Statement1",
      "Effect": "Deny",
      "NotPrincipal": {
        "AWS": [
          "IAMRole"
        ]
      },
      "Action": [
        "s3:PutObject",
        "s3:GetObject"
      ],
      "Resource": [
        "arn:aws:s3:::BucketName/*",
        "arn:aws:s3:::BucketName"
      ],
      "Condition": {
        "NotIpAddress": {
          "aws:SourceIp": [
            "IPAddress"
          ]
        },
        "StringNotEquals": {
          "aws:PrincipalArn": "arn:aws:iam::1234567890:role/
RoleName"
        }
      }
    }
  ]
}

```

## Snowball Edge AWS CLI 문제 해결

다음 주제는 AWS Snowball Edge 디바이스 및 AWS CLI 작업을 할 때 발생하는 문제를 해결하는 데 도움이 됩니다.



## AWS CLI 오류 메시지 문제 해결: Snowball Edge에서 "Profile Cannot Be Null"

로 작업 AWS CLI할 때 프로파일이 null일 수 없다는 오류 메시지가 표시될 수 있습니다. AWS CLI 가 설치되지 않았거나 AWS CLI 프로파일이 구성되지 않은 경우이 오류가 발생할 수 있습니다.

### 취할 조치

워크스테이션 AWS CLI 에서를 다운로드하고 구성했는지 확인합니다. 자세한 내용은 AWS Command Line Interface 사용 설명서의 [AWS CLI의 최신 버전 설치 또는 업데이트](#)를 참조하세요.

## Snowball Edge를 AWS CLI 사용하여 로 데이터를 전송할 때 null 포인터 오류 문제 해결

AWS CLI 를 사용하여 데이터를 전송할 때 null 포인터 오류가 발생할 수 있습니다. 이 오류는 다음과 같은 조건에서 발생할 수 있습니다.

- 지정된 파일 이름의 철자가 잘못된 경우. 예: flower.png 대신에 flower.npg 또는 flower.png로 되어 있는 경우.
- 지정된 경로가 잘못된 경우. 예: C:\Documents\flower.png 대신에 C:\Documents \flower.png로 되어 있는 경우.
- 파일이 손상된 경우

### 취할 조치

파일 이름과 경로가 정확한지 확인하고 다시 시도합니다. 이 문제가 계속 발생하면 파일이 손상된 것은 아닌지 확인하여 전송을 포기하거나 파일 복구를 시도합니다.

## Snowball Edge의 컴퓨팅 인스턴스 문제 해결

다음에서 Snowball Edge 디바이스의 컴퓨팅 인스턴스에 대한 문제 해결 팁을 확인할 수 있습니다.

### 주제

- [가상 네트워크 인터페이스의 IP 주소가 0.0.0.0인 경우](#)
- [대규모 컴퓨팅 인스턴스를 시작할 때 Snowball Edge 디바이스가 응답 중지](#)
- [Snowball Edge의 인스턴스에 루트 볼륨이 하나 있음](#)
- [보호되지 않는 프라이빗 키 파일 오류](#)

## 가상 네트워크 인터페이스의 IP 주소가 0.0.0.0인 경우

이 문제는 VNIC(가상 네트워크 인터페이스)에 연결된 물리적 NIC(네트워크 인터페이스)의 IP 주소도 0.0.0.0인 경우 발생할 수 있습니다. NIC에 IP 주소가 구성되지 않은 경우(예: 디바이스 전원을 방금 켜 경우)에도 이 문제가 생길 수 있습니다. 또한 잘못된 인터페이스를 사용 중인 경우에도 이 문제가 발생할 수 있습니다. 예를 들어 SFP+ 인터페이스의 IP 주소를 얻으려 하나 네트워크에 연결된 인터페이스가 RJ45인 경우입니다.

### 취할 조치

이러한 상황이 발생하면 다음 작업을 수행할 수 있습니다.

- IP 주소가 있는 NIC와 연결된 새 VNI를 생성합니다. 자세한 내용은 [Snowball Edge의 컴퓨팅 인스턴스에 대한 네트워크 구성](#) 단원을 참조하십시오.
- 기존 VNI를 업데이트합니다. 자세한 내용은 [Snowball Edge에서 가상 네트워크 인터페이스 업데이트](#) 단원을 참조하십시오.

## 대규모 컴퓨팅 인스턴스를 시작할 때 Snowball Edge 디바이스가 응답 중지

Snowball Edge가 인스턴스 시작을 중지한 것으로 보일 수 있습니다. 이는 일반적인 사례가 아닙니다. 그러나 최대 규모의 컴퓨팅 인스턴스를 시작하는 데 한 시간 이상 걸릴 수 있습니다.

인스턴스의 상태를 확인하려면 Snowball Edge의 HTTP 또는 HTTPS Amazon EC2-compatible 엔드포인트에 대해 AWS CLI 명령 `aws ec2 describe-instances` 실행을 사용합니다.

## Snowball Edge의 인스턴스에 루트 볼륨이 하나 있음

인스턴스에는 기본적으로 한 개의 루트 볼륨이 있습니다. 모든 sbe 인스턴스는 단일 루트 볼륨을 사용하지만, Snowball Edge 디바이스에서는 애플리케이션의 필요에 따라 블록 스토리지를 추가하거나 제거할 수 있습니다. 자세한 내용은 [Snowball Edge에서 Amazon EC2-compatible 인스턴스와 함께 블록 스토리지 사용](#) 단원을 참조하십시오.

## 보호되지 않는 프라이빗 키 파일 오류

이 오류는 컴퓨팅 인스턴스의 .h 파일에 읽기/쓰기 권한이 없는 경우 발생할 수 있습니다.

### 취할 조치

다음 절차를 통해 파일에 대한 권한을 변경하여 이 문제를 해결할 수 있습니다.

1. 터미널을 열고 .pem 파일을 저장한 위치로 이동합니다.
2. 다음 명령을 입력합니다.

```
chmod 400 filename.pem
```

## 문서 이력

- API 버전: 1.0
- 최신 설명서 업데이트: 2024년 3월 14일

다음 테이블에서는 2018년 6월 이후 AWS Snowball Edge 개발자 안내서에서 변경된 중요 사항에 대해 설명합니다. 설명서 업데이트에 대한 알림을 받으려면 RSS 피드를 구독하시면 됩니다.

변경 사항	설명	날짜
<a href="#">Snowball Edge 디바이스의 Tape Gateway가 더 이상 사용되지 않음</a>	Snowball Edge 디바이스에서 Tape Gateway 기능을 더 이상 사용할 수 없습니다.	2024년 3월 14일
<a href="#">파일 인터페이스가 더 이상 사용되지 않음</a>	파일 인터페이스는 더 이상 데이터 전송에 사용할 수 없습니다.	2024년 3월 1일
<a href="#">Snowball Edge의 Amazon S3 호환 스토리지 Snowball Edge 스토리지 최적화 210TB 디바이스에서 사용 가능</a>	Snowball Edge의 Amazon S3 호환 스토리지는 Snowball Edge 스토리지 최적화 210TB 디바이스의 S3 스토리지에 사용할 수 있습니다. 자세한 내용은 <a href="#">Snowball Edge에서 Amazon S3 호환 스토리지 사용을 참조하세요</a> .	2024년 2월 26일
<a href="#">디바이스 주문 시 사용자 지정 AMI 포함</a>	이제 Snowball Edge 작업을 주문하는 동안 사용자 지정 Amazon Machine Image를 미리 로드할 수 있습니다. 자세한 내용은 <a href="#">에서 AMI 추가를 AWS Marketplace 참조하세요</a> .	2023년 11월 15일
<a href="#">Snowball Edge의 Amazon S3 호환 스토리지 정식 출시</a>	Snowball Edge의 Amazon S3 호환 스토리지는 Snowball Edge 컴퓨팅 최적화 디바이스	2023년 4월 20일

에서 지원됩니다. 자세한 내용은 [Snowball Edge의 Amazon S3 호환 스토리지](#)를 참조하세요.

### [새로 AWS 리전 지원됨](#)

AWS Snowball Edge 이제 중동(UAE) 리전에서 지원됩니다. 이 리전의 엔드포인트에 대한 자세한 내용은 [Snowball Edge 엔드포인트 및 할당량을 참조하세요](#) AWS 일반 참조. 배송에 대한 자세한 내용은 [Shipping Considerations for Snowball Edge](#) 섹션을 참조하세요.

2023년 3월 6일

### [새로 AWS 리전 지원됨](#)

AWS Snowball Edge 이제 아시아 태평양(자카르타) 리전에서 지원됩니다. 이 리전의 엔드포인트에 대한 자세한 내용은 [Snowball Edge 엔드포인트 및 할당량을 참조하세요](#) AWS 일반 참조. 배송에 대한 자세한 내용은 [Shipping Considerations for Snowball Edge](#) 섹션을 참조하세요.

2022년 9월 7일

### [Snowball Edge를 위한 라지 데이터 마이그레이션](#)

Snowball Edge는 이제 라지 데이터 마이그레이션 계획의 자동화를 지원합니다. 자세한 내용은 [Large Data Migration](#)(수동 단계) 및 필요한 경우에는 [Create a Large Data Migration Plan](#) 섹션을 참조하세요.

2022년 4월 27일

<a href="#"><u>소개 AWS Snowball Edge Device Management</u></a>	Snowball Edge 디바이스 관리를 사용하면 Snowball Edge 디바이스 및 로컬 AWS 서비스를 원격으로 관리할 수 있습니다. 모든 Snowball Edge 디바이스는 Snowball Edge 디바이스 관리를 지원하며, Snowball Edge를 사용할 수 AWS 리전 있는 대부분의에서 새 디바이스에 사전 설치됩니다. 자세한 내용은 <a href="#"><u>사용하여 디바이스 관리를 참조 AWS Snowball Edge Device Management</u></a> 하세요.	2022년 4월 27일
<a href="#"><u>Snowball Edge NFS 구성</u></a>	스토리지 최적화 디바이스용 <a href="#"><u>Snowball Edge NFS 구성</u></a> 을 추가했습니다.	2022년 4월 21일
<a href="#"><u>로드 밸런서 속도 제한</u></a>	Snowball Edge는 이제 서버 클러스터 환경에서 요청을 분산하기 위한 <a href="#"><u>속도 제한</u></a> 을 지원합니다.	2022년 4월 19일
<a href="#"><u>Tape Gateway를 통한 Snowball Edge 지원</u></a>	이제 Tape Gateway 서비스를 호스팅하도록 특별히 구성된 Snowball Edge 디바이스를 주문할 수 있습니다. 이러한 기술 조합을 사용하면 안전한 오프라인 테이프 데이터 마이그레이션이 용이해집니다.	2021년 11월 30일
<a href="#"><u>NTP(Network Time Protocol) 서버 구성 지원</u></a>	이제 Snowcone Edge 디바이스가 외부 NTP(Network Time Protocol) 서버 구성을 지원합니다.	2021년 11월 16일

## [NFS 오프라인 데이터 전송 지원](#)

이제 Snowball Edge 디바이스가 NFS를 사용한 오프라인 데이터 전송을 지원합니다. 자세한 내용은 [Using NFS for Offline Data Transfer](#) 섹션을 참조하세요.

2021년 8월 4일

## [새로 AWS 리전 지원됨](#)

이제 아프리카(케이프타운)에서 Snowball Edge 디바이스를 사용할 수 있습니다 AWS 리전. 자세한 내용은의 [Snowball Edge Edge 엔드포인트 및 할당량을 참조하세요](#)AWS 일반 참조. 배송에 대한 자세한 내용은 [Shipping Considerations for Snowball Edge](#) 섹션을 참조하세요.

2020년 11월 23일

## [자신의 이미지를 디바이스로 가져오기 위한 지원](#)

이제 이미지 스냅샷을 Snowball Edge 디바이스로 가져와서 Amazon EC2 호환 Amazon Machine Image(AMI)로 등록할 수 있습니다. 자세한 내용은 [Importing an Image into Your Device as an Amazon EC2 AMI](#) 섹션을 참조하세요.

2020년 11월 9일

[새로 AWS 리전 지원됨](#)

이제 유럽(밀라노)에서 Snowball Edge 디바이스를 사용할 수 있습니다 AWS 리전. 자세한 내용은의 [Snowball Edge Edge 엔드포인트 및 할당량을 참조하세요](#) AWS 일반 참조. 배송에 대한 자세한 내용은 [Shipping Considerations for Snowball Edge](#) 섹션을 참조하세요.

2020년 9월 30일

[콘텐츠 재구성](#)

AWS Snow 패밀리 관리 콘솔 워크플로에 맞는 시작하기 섹션을 생성하고 명확성을 위해 다른 섹션을 업데이트했습니다. 자세한 내용은 [Getting Started with an AWS Snowball Edge](#) 섹션을 참조하세요.

2020년 9월 17일

[소개 AWS OpsHub](#)

이제 Snowball Edge는 디바이스 및 로컬 AWS 서비스를 관리하는 데 사용할 수 AWS OpsHub있는 사용자 친화적인 도구를 제공합니다. 자세한 내용은 [를 사용하여 Snowball 디바이스 관리를 참조 AWS OpsHub](#) 하세요.

2020년 4월 16일

[AWS Identity and Access Management 이제 AWS Snowball Edge 디바이스에서 로컬로 \(IAM\)를 사용할 수 있습니다.](#)

이제 AWS Identity and Access Management (IAM)을 사용하여 AWS Snowball Edge 디바이스에서 실행되는 리소스에 대한 AWS 액세스를 안전하게 제어할 수 있습니다. 자세한 내용은 [Using IAM Locally](#) 섹션을 참조하세요.

2020년 4월 16일



[새로운 Snowball Edge 스토리지 최적화\(데이터 전송용\) 디바이스 옵션 도입](#)

Snowball은 현재 컴퓨팅 최적화 및 GPU 디바이스를 기반으로 새로운 스토리지 최적화 디바이스를 추가합니다. 자세한 내용은 [Snowball Edge Device Options](#) 섹션을 참조하세요.

2020년 3월 23일

[NFC 태그 검증 지원](#)

Snowball Edge 컴퓨팅 최적화 디바이스(GPU 기반 또는 미 기반)에는 NFC 태그가 내장되어 있습니다. Android에서 사용할 수 있는 AWS Snowball Edge 확인 앱을 사용하여 이러한 태그를 스캔할 수 있습니다. 자세한 내용은 [Validating NFC Tags](#) 섹션을 참조하세요.

2018년 12월 13일

[컴퓨팅 인스턴스에 대한 보안 그룹 사용 가능](#)

Snowball Edge 디바이스의 보안 그룹은 AWS 클라우드의 보안 그룹과 유사하지만 몇 가지 세밀한 차이점이 있습니다. 자세한 내용은 [Security Groups in Snowball Edge Devices](#) 섹션을 참조하세요.

2018년 11월 26일

[온프레미스 업데이트 도입](#)

이제 Snowball Edge 디바이스를 로컬 환경에서 실행하도록 하는 소프트웨어를 업데이트할 수 있습니다. 온프레미스 업데이트의 경우 인터넷 연결이 필요합니다. 자세한 내용은 [Updating an Snowball Edge](#) 섹션을 참조하세요.

2018년 11월 26일

<a href="#">Snowball Edge에 대한 새 디바이스 옵션 도입</a>	Snowball Edge 디바이스에는 3가지 옵션, 즉 스토리지 최적화, 컴퓨팅 최적화, GPU 기반이 있습니다. 자세한 내용은 <a href="#">Snowball Edge Device Options</a> 섹션을 참조하세요.	2018년 11월 15일
<a href="#">새로 AWS 리전 지원됨</a>	이제 아시아 태평양(뭄바이)에서 Snowball Edge 디바이스를 사용할 수 있습니다. 이 리전에서에서는 AWS Lambda 제공하는 컴퓨팅 인스턴스 및 AWS IoT Greengrass 가 지원되지 않습니다.	2018년 9월 24일
<a href="#">Snowball Edge 디바이스에서 Amazon EC2 호환 컴퓨팅 인스턴스를 지원하는 기능 도입</a>	AWS Snowball Edge 는 이제 Snowball Edge 디바이스에서 실행되는 <a href="#">Amazon EC2 컴퓨팅 인스턴스</a> 를 사용하는 로컬 작업을 지원합니다.	2018년 7월 17일
<a href="#">문제 해결 콘텐츠 개선</a>	문제 해결에 관한 장이 업데이트 및 재구성되었습니다.	2018년 7월 11일

기계 번역으로 제공되는 번역입니다. 제공된 번역과 원본 영어의 내용이 상충하는 경우에는 영어 버전이 우선합니다.